University of Modena and Reggio Emilia

XXXII cycle of the International Doctorate School in
Information and Communication Technologies

Doctor of Philosophy dissertation in
Computer Engineering and Science

Learning to describe salient objects in images with vision and language

Saliency prediction, image captioning, and cross-modal retrieval

Marcella Cornia

Supervisor: Prof. Rita Cucchiara
PhD Course Coordinator: Prof. Sonia Bergamaschi

Modena, 2020

Review committee composed of:

Elisa Ricci, University of Trento and Fondazione Bruno Kessler
Hamed R. Tavakoli, Nokia Technologies

Contents

1 Introduction **1**

2 Literature survey **7**

2.1 Saliency prediction 7

2.1.1 Image saliency 7

2.1.2 Video saliency 8

2.2 Automatic captioning 9

2.2.1 Image captioning 9

2.2.2 Video captioning 11

2.3 Cross-modal retrieval 13

3 Saliency prediction **15**

3.1 Multi-Level Network (ML-Net) 18

3.1.1 Model architecture 18

3.1.2 Loss function 21

3.2 Saliency Attentive Model (SAM) 21

3.2.1 Model architecture 22

3.2.2 Loss function 27

3.3 Evaluating saliency models 28

3.3.1 Datasets 28

3.3.2 Evaluation metrics 29

3.4 Experimental results 31

3.4.1 Training and implementation details 31

3.4.2 Evaluation of ML-Net model 33

3.4.3 Evaluation of SAM model 34

3.4.4 Comparison with state of the art 38

3.4.5 Qualitative results 42

4 Image captioning 47

4.1 Boosting captioning with saliency 49

4.1.1 What is hit by saliency? 50

4.1.2 Saliency and context aware attention 53

4.1.3 Experimental settings 56

4.1.4 Comparison with baselines 58

4.1.5 Comparison with state of the art 62

4.1.6 Analysis of generated captions 63

4.1.7 Analysis of attentive states 64

4.1.8 Qualitative results 65

4.2 Boosting captioning with fully-attentive models 68

4.2.1 Meshed-Memory Transformer 69

4.2.2 Training 74

4.2.3 Experimental settings 75

4.2.4 Ablation study 77

4.2.5 Comparison with state of the art 80

4.2.6 Qualitative results and visualization 82

4.2.7 Describing novel objects 86

5 Controllable captioning 89

5.1 Controlling caption generation via detection selection 90

5.1.1 Preliminaries 92

5.1.2 Generating controllable captions 93

5.1.3 Objective 96

5.1.4 Controllability through a set of detections 97

5.1.5 Implementation details 98

5.1.6 The COCO Entities dataset 100

5.1.7 Experimental settings 102

5.1.8 Baselines 103

5.1.9 Quantitative results 104

5.1.10 Qualitative results 105

5.2 Controlling caption generation via people identity 110

5.2.1 The M-VAD Names dataset 111

5.2.2 Replacing the “someone” 117

5.2.3 Experimental settings 121

5.2.4 Experimental results 121

5.2.5 Testing on different movies 124
5.2.6 Qualitative results 125

6 Cross-modal retrieval 129
6.1 Toward cycle-consistent approaches 130
6.1.1 A cycle-consistent text and image retrieval network . . . 131
6.1.2 Experimental settings 136
6.1.3 Comparison with baselines 138
6.1.4 Comparison with state of the art 141
6.1.5 Cross-dataset experiments 142
6.1.6 Text reconstruction results 143
6.1.7 Complexity and run-time analysis 144
6.1.8 Qualitative results 145
6.2 Retrieving images and textual sentences in the artistic domain . 147
6.2.1 Semi-supervised cross-modal retrieval 147
6.2.2 The BibleVSA dataset 152
6.2.3 Experimental settings 155
6.2.4 Analysis of artistic visual-semantic data 156
6.2.5 Cross-modal retrieval results 157
6.2.6 Evaluation of semi-supervised embeddings 158

7 Conclusions 163

A Other research activities 169
A.1 Image-to-image translation 170
A.2 Face retrieval 172
A.3 Vision-and-language navigation 175

B List of publications 177

Chapter 1

Introduction

Replicating the human ability to connect vision and language has recently been gaining a lot of attention in computer vision, artificial intelligence, and natural language processing, resulting in new models and architectures capable of automatically describing images with textual descriptions. This task, called image captioning, requires not only to recognize salient objects in an image and understand their interactions, but also to verbalize them using natural language, which makes itself very challenging. In this thesis, we present state-of-the-art solutions for the aforementioned problems covering all aspects involved in the generation of natural sentences, starting from the prediction of which parts of the image are salient for human eyes and then moving to the development of image captioning models.

When humans describe a scene, they look at an object before naming it in a sentence, as selective mechanisms attract their gaze on salient and relevant parts of the scene. Motivated by the importance of automatically estimating the human focus of attention on images, the first part of the dissertation introduces two different saliency prediction models based on deep neural networks. In the first model, presented in Sec. 3.1, we use a combination of image features extracted at different levels of a convolutional neural network to estimate the saliency of an image. In the second model, introduced in Sec. 3.2, we instead employ a recurrent architecture together with neural attentive mechanisms that focus on the most salient regions of the input image to iteratively refine the predicted saliency map. We experimentally validate both solutions on different saliency benchmarks demonstrating their effectiveness with respect to several other methods designed for this task (Sec. 3.3 and 3.4).

Despite saliency prediction identifies the most relevant regions of an image, it has rarely been incorporated in a captioning architecture, even though this type of supervision could improve image captioning performance. Following this intuition, in Sec. 4.1, we show how incorporating saliency prediction to effectively enhance the quality of image descriptions and introduce a captioning model that extends the classical machine attention paradigm in order to take into account salient regions as well as the context of the image. Inspired by the recent advent of fully-attentive models, in Sec. 4.2, we instead investigate the use of the Transformer model in image captioning and we propose a novel captioning architecture in which the recurrent relation is abandoned in favor of the use of self-attention. Experimental results show that our solution is able to achieve a new state of the art for standard image captioning reaching the first place of the public leaderboard of the most important captioning benchmark.

In addition to the use of fully-attentive models, the captioning task has achieved strong improvements also thanks to modern training strategies based on reinforcement learning and attentive mechanisms over image regions. Nevertheless, standard captioning approaches provide no way of controlling which regions are described and what importance is given to each region. This lack of controllability creates a distance between humans and machine intelligence, as humans can manage the variety of ways in which an image can be described and select the most appropriate one depending on the task and the context at hand. Most importantly, this also limits the applicability of captioning algorithms to complex scenarios in which some control over the generation process is needed. To explicitly address these shortcomings, in Sec. 5.1, we present an image captioning model that can generate diverse natural language captions depending on a control signal that can be given either as a sequence or as a set of image regions which need to be described. We experimentally validate our proposal and we demonstrate that our method achieves state-of-the-art performances on controllable image captioning, in terms of both caption quality and diversity of generated textual descriptions. On a related line, we also explore another possible application scenario of controllable captioning, *i.e.* that of naming characters in movies with their proper names (Sec. 5.2). This task not only requires to detect, track, and recognize people within a set of characters, but also to have a form of conditioning of the language model which is in charge of generating the textual description enriched by character names. To this aim, we introduce a novel dataset that links character names mentioned in the movie captions to their visual appearances, and we develop different multimodal approaches which can solve the naming problem from already generated captions.

In the last part of the thesis, we present different solutions for cross-modal retrieval, another task related to vision and language that consists of finding images corresponding to a given textual query and, vice versa, identifying textual elements which describe a given query image. In particular, we first cast the problem of cross-modal retrieval as that of learning a translation between the textual and visual domain with a reconstruction objective that keeps the overall process cycle-consistent (Sec. 6.1). Then, in Sec. 6.2, we show the application of retrieval techniques in a challenging scenario, *i.e.* that of digital humanities and cultural heritage, obtaining promising results using semi-supervised approaches.

Activities carried out during the Ph.D.

Beside the research activities described in this thesis, and those briefly summarized in Appendix A, I also took part in other teaching and service activities, which are reported below together with a list of attended conferences and schools. The complete list of my publications is instead reported in Appendix B.

Teaching activities

- Teaching assistant for the Computer Architecture undergraduate course, at University of Modena and Reggio Emilia (2019).
- Laboratory lecturer for the Computer Vision graduate course, at University of Modena and Reggio Emilia (2017, 2018).

Grants and awards

- Certificate of merit for national and international research from University of Modena and Reggio Emilia, 2019.
- Travel award for Women in Computer Vision, CVPR Workshops (Long Beach, United States), 2019.
- Travel award for Women in Computer Vision, ECCV Workshops (Munich, Germany), 2018.
- Runner-up, best paper award at International Summer School on Artificial Intelligence (Udine, Italy), 2018.

- Travel award for Women in Computer Vision, CVPR Workshops (Salt Lake City, United States), 2018.
- First place at LSUN Saliency Prediction Challenge, CVPR Workshops (Honolulu, United States), 2017.
- Travel award for Women in Computer Vision, CVPR Workshops (Las Vegas, United States), 2016.

Participation to national projects

- “AI for Cultural Heritage” project, co-funded by Fondazione Cassa di Risparmio di Modena.
- “CultMEDIA” project of the National Technological Cluster on Technologies for the Cultural Heritage, co-funded by the Italian Ministry of Education, University and Research.
- “Vision for Augmented Experience” project, co-funded by Fondazione Cassa di Risparmio di Modena.

Journals reviewing

- IEEE Transactions on Image Processing
- IEEE Transactions on Multimedia
- ACM Transactions on Multimedia Computing Communications and Applications
- Computer Vision and Image Understanding
- IEEE Robotics and Automation Letters
- IEEE Signal Processing Letters

Other academic services

- Organizer of “Women in ICPR” Workshop at International Conference on Pattern Recognition, 2020.
- Reviewer for European Conference on Computer Vision (2020).

- Reviewer for IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020).
- Program committee member of AAAI Conference on Artificial Intelligence (2020).
- Invited Tutorial “Vision, Language, and Action: from Captioning to Embodied AI” at International Conference on Image Analysis and Processing, 2019.
- Reviewer for IEEE/CVF International Conference on Computer Vision (2019).
- Program committee member of ACM Multimedia (2019).

Conferences and schools attended

- International Conference on Image Analysis and Processing (Trento, Italy), 2019.
- IEEE/CVF Conference on Computer Vision and Pattern Recognition (Long Beach, United States), 2019.
- European Conference on Computer Vision (Munich, Germany), 2018.
- International Summer School on Artificial Intelligence (Udine, Italy), 2018.
- IEEE/CVF Conference on Computer Vision and Pattern Recognition (Salt Lake City, United States), 2018.
- International Workshop on Computer Vision (Modena, Italy), 2018.
- International Conference on Computer Vision (Venice, Italy), 2017.
- International Conference on Image Analysis and Processing (Catania, Italy), 2017.
- International Computer Vision Summer School (Sicily, Italy), 2017.
- IEEE/CVF Conference on Computer Vision and Pattern Recognition, (Las Vegas, United States), 2016.

Chapter 2

Literature survey

In this chapter, we provide an overview of the most important research works related to the tasks tackled in this thesis. In details, we first review the literature related to saliency prediction (Sec. 2.1), then we move to image and video captioning (Sec. 2.2), and we finally focus on cross-modal retrieval (Sec. 2.3).

2.1 Saliency prediction

In this section, we provide a brief overview of image and video saliency models proposed before and after the advent of deep learning. For a comprehensive analysis of saliency literature, we refer the reader to different surveys published on this topic which focus on traditional saliency prediction methods [16] and on saliency models based on deep neural networks [15].

2.1.1 Image saliency

Pioneering works on saliency prediction were based on the Feature Integration Theory proposed by Treisman *et al.* [188] in the eighties. Itti *et al.* [81] defined the first computational model to predict saliency on images: this work, inspired by Koch and Ullman [96], computed a set of individual topographical maps representing low-level cues such as color, intensity and orientation and combined them into a global saliency map. After this seminal work, a large variety of methods explored the same idea of combining complementary low-level features [19, 61, 42] and often included additional center-surround cues [137, 235]. Other methods enriched predictions exploiting semantic classifiers for detecting higher level concepts such as faces, people, cars, and horizons [24, 89, 238, 51, 14].

Only in the last few years, thanks to the large spread of deep learning techniques, the saliency prediction task has achieved a considerable improvement. One of the first proposals has been the *Ensemble of Deep Networks* (*e*DN) model by Vig *et al.* [199]. This model consists of three convolutional layers followed by a linear classifier that blends feature maps coming from the previous layers. After this work, Kümmerer *et al.* [103, 105] proposed two deep saliency prediction networks: the first, called *DeepGaze I*, was based on the AlexNet model [99], while the second, *DeepGaze II*, was built upon the VGG-19 network [168]. Liu *et al.* [119] presented a multi-resolution CNN (*Mr-CNN*) fine-tuned over image patches centered on fixation and non-fixation locations.

It is well known that deep learning approaches strongly depend on the availability of sufficiently large datasets. The publication of a large-scale eye-fixation dataset, SALICON [84], indeed contributed to a big progress of deep saliency prediction models. Huang *et al.* [75] introduced an architecture consisting of a deep neural network applied at two different image scales. They compared different standard CNN architectures such as AlexNet [99], VGG-16 [168] and GoogleNet [179], in particular showing the effectiveness of the VGG network.

After this work, several deep saliency models based on the VGG network have been published [100, 141, 82, 101, 182, 140, 37]. Recently, Pan *et al.* [140] introduced *SalGAN*, a deep network for saliency prediction trained with adversarial examples. As all other Generative Adversarial Networks, it is composed of two modules, a generator and a discriminator, which combine efforts to produce saliency maps. Liu *et al.* [118], instead, employed the ResNet [64] architecture to extract feature maps from the input image and proposed a model that simultaneously incorporates global and scene contexts to infer image saliency thanks to a deep spatial contextual LSTM which scans the image both horizontally and vertically.

2.1.2 Video saliency

When considering video inputs, saliency estimation is quite different with respect to still images. Indeed, motion is a key factor that strongly attracts human attention. Accordingly, some video saliency models paired bottom-up feature extraction with a further motion estimation step, that can be performed either by means of optical flow [239] or feature tracking [234]. Somehow differently, some models have been proposed to force the coherence of bottom-up features across time. In this setting, previous works addressed feature extraction both in a supervised [130] and unsupervised [209] fashion, whereas temporal smoothness of output maps can be achieved through optical flow motion cues [239] or explicitly conditioning the current map on information from previous frames [162].

As discussed for the image saliency setting, the representation capability of deep learning architectures, along with large labeled datasets, can yield better results. However, only a few video saliency models based on deep neural networks have been proposed [11, 176, 54, 208]. Among them, Bazzani *et al.* [11] introduced a video saliency model based on a recurrent architecture that iteratively updates its hidden state over time and emits the saliency map at each step by means of a Gaussian Mixture Model.

2.2 Automatic captioning

In this section, we review the literature related to image and video captioning. In particular, we give an overview of standard image captioning (Sec. 2.2.1) starting from recurrent-based captioning models and then moving toward fully-attentive approaches. Moreover, we provide a brief description of two other related lines of works: saliency-boosted captioning, and grounded and diverse image description. Finally, we review video captioning approaches mainly focusing on the task of linking visual tracks in the context of movies and TV series to the proper character names (Sec. 2.2.2).

2.2.1 Image captioning

A broad collection of methods have been proposed in the field of image captioning in the last few years. Earlier captioning approaches were based on the generation of simple templates, filled by the output of an object detector or attribute predictor [170, 224]. With the advent of deep neural networks, most captioning techniques have employed RNNs as language models and used the output of one or more layers of a CNN to encode visual information and condition language generation [202, 155, 38, 87]. On the training side, while initial methods were based on a time-wise cross-entropy training, a notable achievement has been made with the introduction of Reinforcement Learning, which enabled the use of non-differentiable caption metrics as optimization objectives [155, 152, 120]. On the image encoding side, instead, single-layer attention mechanisms have been adopted to incorporate spatial knowledge, initially from a grid of CNN features [216, 123, 228], and then using image regions extracted with an object detector [4, 146, 124]. To further improve the encoding of objects and their relationships, Yao *et al.* [226] have proposed to use a graph convolution neural network in the image encoding phase to integrate semantic and spatial relationships between objects. On the same line, Yang *et al.* [223] used a multimodal graph convolution network to modulate scene graphs into visual representations.

Despite their wide adoption, RNN-based models suffer from their limited representation power and sequential nature. After the emergence of convolutional language models, which have been explored for captioning as well [6], new fully-attentive paradigms [192, 35, 174] have been proposed and achieved state-of-the-art results in machine translation and language understanding tasks. Likewise, some recent approaches have investigated the application of the Transformer model [192] to the image captioning task.

In a nutshell, the Transformer comprises an encoder made of a stack of self-attention and feed-forward layers, and a decoder which uses self-attention on words and cross-attention over the output of the last encoder layer. Herdade *et al.* [67] used the Transformer architecture for image captioning and incorporated geometric relations between detected input objects. In particular, they computed an additional geometric weight between object pairs which is used to scale attention weights. Liu *et al.* [111] used the Transformer in a model that exploits visual information and additional semantic knowledge given by an external tagger. On a related line, Huang *et al.* [74] introduced an extension of the attention operator in which the final attended information is weighted by a gate guided by the context. In their approach, a Transformer-like encoder was paired with an LSTM decoder.

Saliency and captioning

Only a few other previous works have investigated the contribution of human eye fixations to generate image descriptions [173, 151, 183]. The first work that has explored this idea was that proposed in [173] which presented an extension of a neural attentive captioning architecture. In particular, the proposed model incorporates human fixation points (obtained with eye-tracking devices) instead of computed saliency maps to generate image captions. This kind of strategy mainly suffers of the need of having both eye fixation and caption annotations. Currently, only the SALICON dataset [84], being a subset of the Microsoft COCO dataset [116], is available with both human descriptions and saliency maps.

Ramanishka *et al.* [151], instead, introduced an encoder-decoder captioning model in which spatiotemporal heatmaps are produced for predicted captions and arbitrary query sentences without explicit attention layers. They refer to these heatmaps as saliency maps, even though they are internal representations of the network, not related with human attention.

Inspired by the relation between saliency, object importance, and language previously studied for scene understanding [233, 232], the approach presented in [183] explores if image descriptions, by humans or models, agree with saliency and if saliency can benefit image captioning. To this end, they proposed a captioning model in which image features are boosted with the corresponding saliency map by exploiting a moving sliding window and mean pooling as aggregation strategies. On a related line, a novel work has been recently presented to study the differences in human attention during free-viewing and image captioning tasks [66].

Grounded and diverse captioning

More principled approaches have been proposed for grounding a caption on the image [149, 158, 73, 87, 72]. Among these models, DenseCap [87] generates multiple descriptions for the same image, each of them describing a specific image region. Further, the Neural Baby Talk approach [124] extends the attentive model in a two-step design in which a word-level sentence template is firstly generated and then filled by object detectors with concepts found in the image.

Another related line of work is that of generating diverse descriptions. Some works have extended the beam-search algorithm to sample multiple captions from the same distribution [200, 3], while different GAN-based approaches have also appeared [33, 167, 206]. Most of these improve on diversity, but suffer on accuracy and do not provide controllability over the generation process. Others have conditioned the generation with a specific style or sentiment [131, 132, 47]. Deshpande *et al.* [34], instead, proposed a diverse image captioning model which uses a control input as a sequence of part-of-speech tags. This approach, while generating diversity, is hardly employable to effectively control the generation of the sentence.

2.2.2 Video captioning

The generation of natural language descriptions of visual content has received large interest also in the context of user-generated videos [38] and movie clips [159, 196]. First approaches described the input video through mean-pooled CNN features [197] or sequentially encoded by a recurrent layer [38, 196]. This strategy was then followed by the majority of video captioning approaches, either by incorporating attentive mechanisms [225] in the sentence decoder, by building a common visual-semantic embedding [143], or by adding external knowledge with language models [195] or visual classifiers [159].

Recent video captioning models have improved both components of the encoder-decoder approach by significantly changing their structure. Yu *et al.* [231] focused on the sentence decoder and proposed a hierarchical model containing a sentence and a paragraph generator. In particular, the sentence generator produces one simple short sentence that describes a specific short video interval by exploiting both temporal and spatial attention mechanisms. In contrast, Pan *et al.* [142] concentrated on the video encoding stage and introduced a hierarchical recurrent encoder to exploit temporal information of videos. In [10], instead, authors proposed a modification of the LSTM cell able to identify discontinuity points between frames or segments and to modify the temporal connections of the encoding layer accordingly.

On a different note, Krishna *et al.* [97] introduced the task of dense-captioning events, which involves both detecting and describing events in a video, and proposed a new model able to identify events of a video while simultaneously describing the detected events in natural language.

Linking visual tracks to names

One of the work presented in this thesis addresses the problem of identifying characters in movies or TV series with the final goal of generating a textual description containing characters' proper names (Sec. 5.2). In this context, computer vision researchers principally focus on linking people with their names by tracking faces in the video and assigning names to them [12, 43, 150, 169, 180]. For example, [43, 169] tackled this problem by automatically aligning subtitles and script texts of movies and TV series. In particular, Everingham *et al.* [43] aimed to associate speaker names present in the movie scripts to the correct faces appearing in the movie clips by detecting face tracks with lip motion. Sivic *et al.* [169] extended the previous work, limited in classifying frontal faces, by adding the detection and recognition of characters in profile views, improving the overall performance.

In [180], each TV series episode is instead modelled as a Markov Random Field, integrating cues from face, speech, and clothing. Bojanowski *et al.* [12] proposed a method to extract actor/action pairs from movie scripts and used them as constraints in a discriminative clustering framework. In [150], authors introduced a joint model for person naming and co-reference resolution which consists in resolving the identity of ambiguous mentions of people such as pronouns (e.g. "he" or "she") and nominals (e.g. "man").

Recently, Rohrbach *et al.* [161] addressed the problem of generating video descriptions with grounded and co-referenced people by proposing a deeply-learned model. This task aims at predicting the spatial location in which a given character appears, and at producing captions with proper names in the correct place. Miech *et al.* [134], instead, addressed the problem of weakly supervised learning of actions and actors from movies by applying an online optimization algorithm based on the Block-Coordinate Frank-Wolfe method. Finally, in [86] an end-to-end system for detecting and clustering faces by identity in full-length movies is proposed. However, this approach is far from the aforementioned works as it only aimed at clustering face tracks without naming the corresponding movie characters.

Several other methods have been proposed towards understanding social aspects of movies and TV series scenes for either classifying different types of interactions [145] or predicting whether people are looking at each other [36, 129]. As an example, Vicol *et al.* [198] introduced a novel dataset which provides graph-based annotations of social situations appearing in movie clips to capture who is present in the clip, their emotional and physical attributes, their relationships, and the interactions between them.

2.3 Cross-modal retrieval

Matching visual data and natural language is a challenging task in computer vision and multimedia. Since visual and textual data belong to two distinct modalities, one of the first approaches [94] has been that of generating a joint visual-semantic embedding space in which images and sentences could be compared. Even if other approaches exist, currently this is still one of the most commonly used solutions.

Following this line, [45] introduced a modification of the Hinge-based loss function to exploit hard negatives, *i.e.* worst matching pairs, during training. This has demonstrated to be effective to improve cross-modal retrieval performance and has been used in almost all subsequent works [41, 58, 77, 109]. Further, Wang *et al.* [205] used a two-branch network composed of an embedding and a similarity branch: while the embedding network translates image and text into a feature representation, the similarity network predicts how well the feature representations match. Differently, Dong *et al.* [39] suggested to tackle the retrieval problem exclusively in the visual space, introducing a deep neural model that learns to predict a visual feature representation from textual input.

Inspired by the use of multiple image descriptors to improve related visual-semantic tasks [4, 218], Lee *et al.* [109] have recently proposed a stacked cross-attention mechanism that matches images and textual descriptions by learning a latent correspondence between detected regions and words of the caption. Wang *et al.* [212] extended this model by integrating an encoding of the relative position of image regions, which has proven to further enhance the learning of the joint embedding. On the same line, Li *et al.* [112] proposed a reasoning model based on graph convolutional networks to generate a visual representation that captures key objects and semantic concepts of a scene. All of these supervised methods have been proven effective when trained on large scale datasets, and are not designed to work with scarce data.

Only a few works have instead applied image-text matching strategies to artistic data. Among them, Garcia *et al.* [48] used additional metadata such as title, author, genre, and period of the paintings to find corresponding image-text pairs. While this model matches images and textual descriptions in a supervised way, in this thesis we also address the problem in a semi-supervised setting, adapting the knowledge learned on a given source domain to align images and text belonging to a different target domain and without directly training the model on the target domain. This solution, which is known as domain adaptation, has been used in a wide variety of applications such as image classification [122], semantic segmentation [70, 28], object detection [80, 27], and image captioning [26, 222]. Typically, it is addressed by minimizing the distance between feature space statistics of the source and target, or by using domain adversarial objectives where a domain classifier is trained to distinguish between the source and target representations.

Chapter 3

Saliency prediction

Visual cognition science has shown that humans, when observing a scene without a specific task to perform, do not focus on each region of the image with the same intensity. Instead, attentive mechanisms guide their gazes on salient and relevant parts [156]. Emulating such selective visual mechanisms has been studied for more than 80 years by neuroscientists [20] and more recently by computer vision researches [81]. In this context, computational saliency has proven to be effective for a wide range of applications like image retargeting [165], object recognition [203], video compression [60], tracking [127], and other data-dependent tasks such as image captioning [173, 183].

Traditionally, algorithms for saliency prediction focused on identifying the fixation points that human viewers would focus on at first glance. Others have concentrated on highlighting the most important object regions in an image [110, 220, 83]. In this chapter, we focus on the first type of saliency models, that try to predict eye fixations over an image.

Inspired by biological studies, researchers have defined hand-crafted and multi-scale features that capture a large spectrum of stimuli: lower-level features (color, texture, contrast) [81, 61] or higher-level concepts (faces, people, text, horizon) [89]. However, given the large variety of aspects that can contribute to define visual saliency, it is difficult to design approaches that combine all these hand-tuned factors in an appropriate way.

In recent years, deep learning techniques have shown impressive results in

This chapter is related to publications [1, 2, 6, 7, 8, 9] reported in Appendix B, by the author of the thesis. See Appendix B for details.

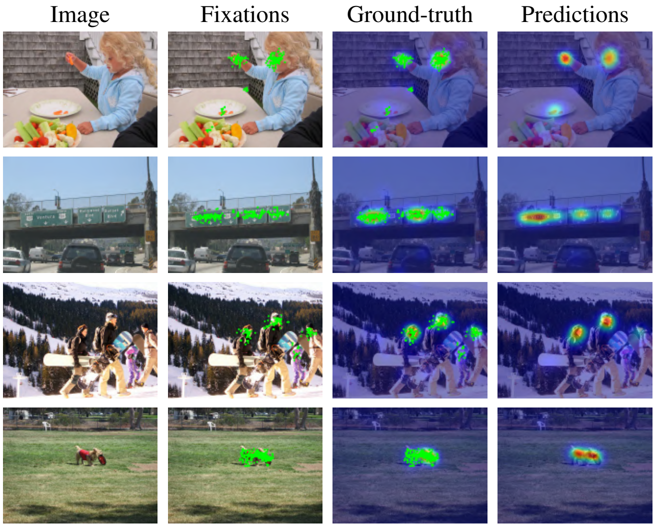


Figure 3.1: Visual saliency prediction aims at predicting where humans gazes will focus on a given image. Ground-truth data is collected by means of eye-tracking glasses or mouse clicks to get eye fixation points, which are then smoothed together to obtain the ground-truth saliency map.

several vision tasks. Motivated by these achievements, first attempts to predict saliency map with deep convolutional networks have been performed [199, 103]. These solutions suffered from the small amount of training data compared to the ones available in other contexts requiring the usage of limited number of layers or the usage of pre-trained architectures generated for other tasks. With the publication of a large scale saliency prediction dataset (*i.e.* SALICON [75]), collected thanks to crowd-sourcing techniques, saliency prediction has achieved a strong improvement resulting in several architectures based on deep learning [100, 141, 82, 140].

Although the use of deep learning techniques helps to go beyond the limitations of hand-crafted models, we are the first that investigate the incorporation of machine attention models [216, 56, 192] in saliency prediction. Machine attention [216] is a computational paradigm which sequentially attends to different parts of an input. This is usually achieved by exploiting a recurrent neural network, and by defining a compatibility measure between its internal state and regions of the input. This paradigm has been successfully applied to image captioning [216] and machine translation [30] to selectively focus on different parts of a sentence, and to action recognition [114] to focus on the relevant parts of a spatio-temporal volume. We argue that machine attention can also be effective for saliency prediction, as a powerful way to process saliency-specific features and to obtain an enhanced prediction.

Moreover, it is well known that when observers view complex scenes presented on computer monitors, there is a strong tendency to look more frequently around the center of the scene than around the periphery [181, 190]. This has been exploited in past works on saliency prediction by incorporating hand-crafted and pre-defined priors into saliency maps [89, 199, 103, 100, 105], thus limiting the saliency model to learn its own priors directly from data.

Contributions

In this chapter, we propose two different saliency prediction models. The first architecture, called Multi-Level Network (ML-Net), learns how to weight features coming from different levels of a CNN, and demonstrates the effectiveness of using medium level features. A new loss function is also designed to train the proposed network and to tackle the imbalance problem of saliency maps. The second architecture, called Saliency Attentive Model (SAM), incorporates an Attentive Convolutional Long Short-Term Memory network that iteratively focuses on relevant spatial locations to refine saliency features. To the best of our knowledge, we are among the first to incorporate attentive models in a saliency prediction architecture. In order to handle the tendency of humans to fix the center region of an image, both networks introduce an explicit prior component. In contrast to previous works, our two models can learn priors in an automatic way directly from training data. We experimentally validate our approaches on different publicly available benchmark datasets and we demonstrate their effectiveness in comparison with several other saliency prediction methods. As additional contribution, we have made the source codes of our two models publicly available[1,2].

The rest of the chapter is organized as follows: in Sec. 3.1 and 3.2, we respectively introduce the architectures of our ML-Net and SAM models, and the loss functions employed during training. Datasets and evaluation metrics used to evaluate saliency models are described in Sec 3.3, while all quantitative and qualitative experiments are presented in Sec. 3.4.

[1] https://github.com/marcellacornia/mlnet

[2] https://github.com/marcellacornia/sam

3.1 Multi-Level Network (ML-Net)

In this section, we present our first saliency prediction model, called ML-Net, that exploits multi-level features extracted from a CNN to predict the final saliency map.

3.1.1 Model architecture

Our saliency model is composed by three main parts: given an input image, a CNN extracts low, medium and high level features; then, an encoding network builds saliency-specific features and produces a temporary saliency map. A prior is then learned and applied to produce the final saliency prediction. Figure 3.2 reports a summary of the architecture.

Feature extraction network

The first component of our architecture is a fully convolutional network with 13 layers, which takes the input image and produces features maps for the encoding network.

We build our architecture on the popular 16 layers model from VGG [168], which is well known for its elegance and simplicity, and at the same time yields nearly state-of-the-art results in image classification and good generalization properties. However, like any standard CNN architecture, it has the significant disadvantage of reducing the size of feature maps at higher levels with respect to the input size. This is due to the presence of spatial pooling layers which have a stride greater than one: the output of each layer is a three-dimensional tensor with shape $k \times \left\lfloor \frac{H}{f} \right\rfloor \times \left\lfloor \frac{W}{f} \right\rfloor$, where k is the number of filters of the layer, and f is the downsampling factor of that stage of the network. In the VGG-16 model there are five max-pooling stages with kernel size $k = 2$ and stride 2. Given an input image with size $W \times H$, the output feature map has size $\lfloor \frac{W}{2^5} \rfloor \times \lfloor \frac{H}{2^5} \rfloor$, thus a fully convolutional model built upon the VGG-16 would output a saliency map rescaled by a factor of 32.

To limit this rescaling phenomenon, we remove the last pooling stage and decrease the stride of the last but one, while keeping unchanged its stride. This way, the output feature map of our feature extraction network are rescaled by a factor of 8 with respect to the input image. In the following, we refer to the output size of the feature extraction network as $w \times h$, with $w = \lfloor \frac{W}{8} \rfloor$ and $h = \lfloor \frac{H}{8} \rfloor$. For reference, a complete description of the network is reported in Figure 3.3.

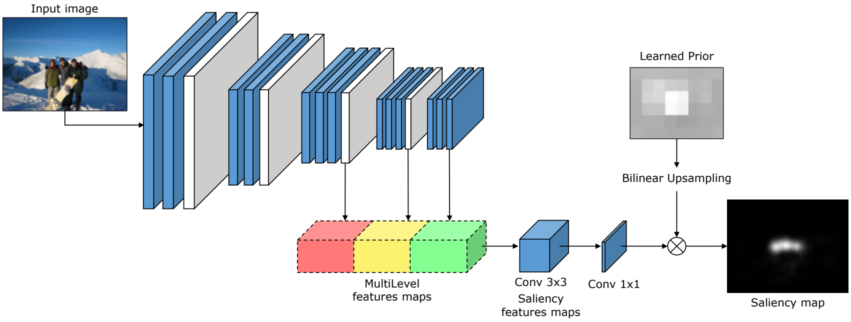


Figure 3.2: Overview of our deep multi-level network for saliency prediction. A CNN is used to compute low and high level features from the input image. Extracted features maps are then fed to an Encoding network, which learns a feature weighting function to generate saliency-specific feature maps. A prior image is also learned and applied to the predicted saliency map.

Encoding network

We take feature maps at three different locations: the output of the third pooling layer (which contains 256 feature maps), that of the last pooling layer (512 feature maps), and the output of the last convolutional layer (512 feature maps). In the following, we respectively call these maps, `conv3`, `conv4` and `conv5`, since they come from the third, fourth and fifth convolutional stage of the network. They all share the same spatial size, and are concatenated to form a tensor with 1280 channels, which is fed to a Dropout layer with retain probability 0.5, to improve generalization. A convolutional layer then learns 64 saliency-specific feature maps with a 3×3 kernel. A final 1×1 convolution learns to weight the importance of each saliency feature map to produce the final predicted feature map.

Prior learning

Psychological studies have shown that when observers look at images, their gazes are biased toward the center [181, 190]. This phenomenon is mainly due to the tendency of photographers to position objects of interest at the center of the image. Also, when people repeatedly watch images with salient information placed in the center, they naturally expect to find the most informative content of the image around its center [190]. Another important reason that encourages this behavior

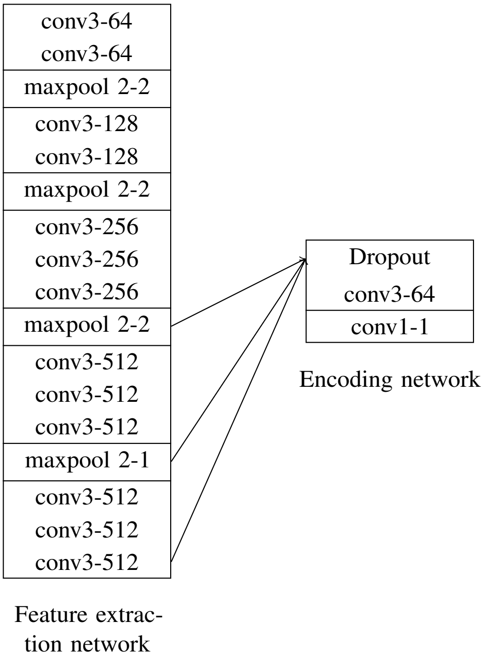


Figure 3.3: Architecture of the feature extraction and encoding networks. The convolutional layer parameters are denoted as "conv<receptive field size>-<number of channels>". The ReLU activation function is not shown for brevity.

is the interestingness of the scene [17]. Indeed, when there are no highly salient regions, humans are inclined to look at the center of the image.

Based on this evidence, the inclusion of center priors is a key component of several recent works of saliency prediction [89, 199, 103, 100, 105]. Differently from existing works, which included pre-defined priors, we let the network learn its own custom prior. To this end, we learn a coarse $w' \times h'$ mask (with $w' \ll w$ and $h' \ll h$), initialized to one, upsample and apply it to the predicted saliency map with pixel-wise multiplication.

Given the learned prior U with shape $w' \times h'$, we interpolate the pixels of U to produce an output prior map V of size $w \times h$. We compute a sampling grid G of shape $w' \times h'$ associating each element of U with real-valued coordinates into

V. If $G_{i,j} = (x_{i,j}, y_{i,j})$ then $U_{i,j}$ should be equal to V at $(x_{i,j}, y_{i,j})$; however since $(x_{i,j}, y_{i,j})$ are real-valued, we convolve with a sampling kernel and set

$$V_{x,y} = \sum_{i=1}^{w'} \sum_{j=1}^{h'} U_{i,j} k_x(x - x_{i,j}) k_y(y - y_{i,j}) \tag{3.1}$$

where $k_x(\cdot)$ and $k_y(\cdot)$ are bilinear kernels, corresponding to $k_x(d) = \max\left(0, \frac{w}{w'} - |d|\right)$ and $k_y(d) = \max\left(0, \frac{h}{h'} - |d|\right)$. w' and h' are set to $\lfloor w/10 \rfloor$ and $\lfloor h/10 \rfloor$ in all our tests.

3.1.2 Loss function

Our loss function is inspired by three objectives: predictions should be pixelwise similar to ground-truth maps, therefore a square error loss $\|\phi(\mathbf{x}_i) - \mathbf{y}\|^2$ is a reasonable choice. Secondly, predicted maps should be invariant to their maximum, and there is no point in forcing the network to produce values in a given numerical range, so predictions are normalized by their maximum. Third, the loss should give the same importance to high and low ground-truth values, even though the majority of ground-truth pixels are close to zero. For this reason, the deviation between predicted values and ground-truth values $\mathbf{y}_i$ is weighted by a linear function $\alpha - \mathbf{y}_i$, which tends to give more importance to pixels with high ground-truth fixation probability.

The overall loss function is thus

$$L(\phi(\mathbf{x}), \mathbf{y}) = \frac{1}{N} \sum_{i=1}^{N} \left\| \frac{\frac{\phi(\mathbf{x}_i)}{\max \phi(\mathbf{x}_i)} - \mathbf{y}_i}{\alpha - \mathbf{y}_i} \right\|^2 + \lambda \|\mathbf{1} - U\|^2 \tag{3.2}$$

where a L_2 regularization term is added to penalize the deviation of the prior mask U from its initial value, thus encouraging the network to adapt to ground-truth maps by changing convolutional weights rather than modifying the prior.

3.2 Saliency Attentive Model (SAM)

After presenting our first solution to address the task of saliency prediction, we here present the architecture of our second model, called SAM (Saliency Attentive Model).

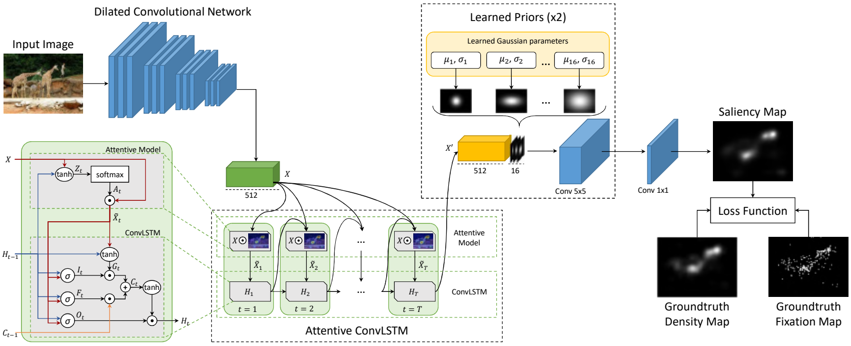


Figure 3.4: Overview of our Saliency Attentive Model (SAM). After computing a set of feature maps on the input image through a Dilated Convolutional Network, an Attentive Convolutional LSTM sequentially enhances saliency features thanks to an attentive recurrent mechanism. Predictions are then combined with multiple learned priors to model the center bias of human-eye fixations. During training, we encourage the network to minimize a combination of different loss functions, thus taking into account different quality aspects that predictions should meet.

3.2.1 Model architecture

The main novelty of our proposal is an Attentive Convolutional model, which recurrently processes saliency features at different locations, by selectively attending to different regions of a tensor, and for the first time uses an LSTM without the concept of time. Predictions are then combined with multiple learned priors which are used to model the human-gaze center bias. To extract feature maps from input images, we employ a Convolutional Neural Network model. Instead of using a pre-defined CNN, we propose a Dilated Convolutional Network to limit the rescaling effects which can worse saliency prediction performance. A new combination of different loss functions is finally used to train the whole network by simultaneously taking into account different quality aspects. The overall architecture of our model is shown in Figure 3.4.

Attentive Convolutional LSTM

Long Short-Term Memory networks [68] achieved good performances on several tasks in which time dependencies are a key component [38, 90, 215, 10], but they

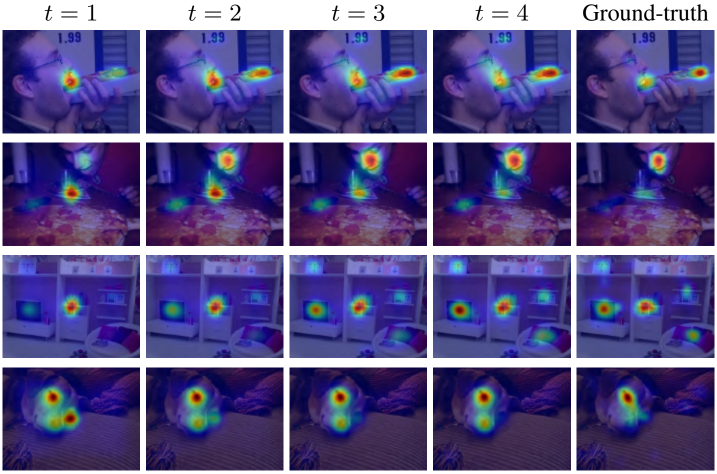


Figure 3.5: Progressive refinement of predictions performed by the Attentive ConvLSTM. The first and the second row show a progressive change of focus in the saliency map, so that regions which were wrongly predicted as salient are progressively corrected, and truly salient regions are correctly identified. The third and the fourth row, instead, respectively show an increase and a reduction of saliency in regions of the image that have been (or have not been) considered as salient at the first timestep. In all cases, the result is a progressive approach of the saliency map to the ground-truth.

can not be directly employed for saliency prediction, as they work on sequences of time varying vectors. We extend the traditional LSTM to work on spatial features: formally this is achieved by substituting dot products with convolutional operations in the LSTM equations. Moreover, we exploit the sequential nature of LSTM to process features in an iterative way, instead of using the model to deal with temporal dependencies in the input.

To explain our proposal of the attentive model, let's consider the LSTM scheme on the left part of Fig. 3.4. Here the LSTM takes as input a stack of features extracted from the input image (X in Fig. 3.4) and produces a refined stack of feature maps (X' in Fig. 3.4) entering in the learned prior module. The LSTM works by sequentially updating an internal state, according to the values of three sigmoid gates. Specifically, the update is driven by the following equations:

$$I_t = \sigma(W_i * \tilde{X}_t + U_i * H_{t-1} + b_i) \tag{3.3}$$

$$F_t = \sigma(W_f * \tilde{X}_t + U_f * H_{t-1} + b_f) \tag{3.4}$$

$$O_t = \sigma(W_o * \tilde{X}_t + U_o * H_{t-1} + b_o) \tag{3.5}$$

$$G_t = \tanh(W_c * \tilde{X}_t + U_c * H_{t-1} + b_c) \tag{3.6}$$

$$C_t = F_t \odot C_{t-1} + I_t \odot G_t \tag{3.7}$$

$$H_t = O_t \odot \tanh(C_t). \tag{3.8}$$

Here, the gates I_t, F_t, O_t, the candidate memory G_t, memory cell C_t, C_{t-1}, and hidden state H_t, H_{t-1} are 3-d tensors, each of them having 512 channels. $*$ represents the convolutional operator, all W and U are 2-d convolutional kernels, and all b are learned biases.

The input of the LSTM layer $\tilde{X}_t$ is computed, at each timestep (*i.e.* at each iteration), through an attentive mechanism. In particular, an attention map is generated by convolving the previous hidden state H_{t-1} and the input X, feeding the result to a $\tanh$ activation function and finally convolving with a one channel convolutional kernel:

$$Z_t = V_a * \tanh(W_a * X + U_a * H_{t-1} + b_a). \tag{3.9}$$

The output of this operations is a 2-d map from which we can compute a normalized spatial attention map through the *softmax* operator:

$$A_t^{ij} = p(att_{ij}|X, H_{t-1}) = \frac{\exp(Z_t^{ij})}{\sum_i \sum_j \exp(Z_t^{ij})} \tag{3.10}$$

where A_t^{ij} is the element of the attention map in position (i, j). The attention map is applied to the input X with an element-wise product between each channel of the feature maps and the attention map:

$$\tilde{X}_t = A_t \odot X. \tag{3.11}$$

Fig. 3.5 shows saliency predictions on four sample images, using the output of the ConvLSTM module at different timesteps as input of the rest of the model. As can be noticed, predictions are progressively refined by modifying the initial map given by the CNN. This refinement results in an significant enhancement of the predictions.

Learned Priors

As mentioned before, one of the most important cues of human gazes is that there is a strong tendency to look at the center of scene. In our previous saliency prediction model (Sec. 3.1), we have incorporated this important property of human eye fixations by learning a single prior map which is applied to the final prediction in a multiplicative way.

In this case, we instead let the network learn multiple prior maps. To reduce the number of parameters and facilitate the learning, we constraint each prior to be a 2d Gaussian function, whose mean and covariance matrix are instead freely learnable. This lets the network learn its own priors purely from data, without relying on assumptions from biological studies.

We model the center bias by means of a set of Gaussian functions with diagonal covariance matrix. Means and variances are learned for each prior map according to the following equation:

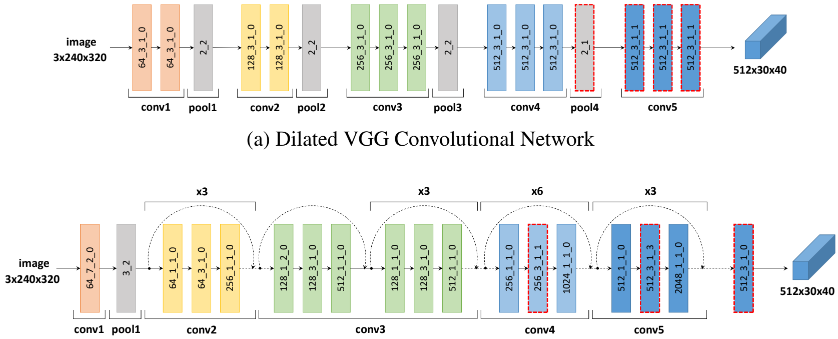
$$f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\left(\frac{(x-\mu_x)^2}{2\sigma_x^2} + \frac{(y-\mu_y)^2}{2\sigma_y^2}\right)\right). \tag{3.12}$$

Our network learns the parameters of N Gaussian functions (in our experiments $N = 16$) and generates the relative prior maps. Since the X' tensor has 512 channels, after the concatenation with learned prior maps, we obtain a tensor with 528 channels. The resulting tensor is fed through a convolutional layer with 512 filters. This operation adds more non-linearity to the model and proves to be effective with respect to other previous works. The entire prior learning module is replicated two times.

Dilated Convolutional Network

As previously mentioned, one of the main drawbacks of using CNNs to extract features for saliency prediction is that they considerably rescale the input image during the feature extraction phase, thus worsening the prediction accuracy. In the following, we devise a strategy which increases the output resolution of a CNN while preserving the scale at which convolutional filters operate and the number of parameters. This makes it possible to use pre-trained weights, and thus to reduce the need for fine-tuning convolutional filters after the network structure has been modified.

The intuition of the approach is that given a CNN of choice and one of its layers having stride $s > 1$, we can increase the output resolution by reducing



(a) Dilated VGG Convolutional Network

(b) Dilated Residual Convolutional Network

Figure 3.6: Overall architectures of Dilated Convolutional Networks based on the VGG-16 and ResNet-50 models. Convolutional and pooling blocks are respectively expressed in terms of `channels_kernel_stride_holes` and `kernel_stride`. On top of the ResNet model, we report the number of repetitions for each block. Red dashed edges indicate modified layers with respect to the original networks.

the stride of the layer, and adding dilation [230] to all the layers which follow the chosen layer. In this way, all convolutional filters still operate on the same scale they have been trained for. We apply this technique on two recent feature extraction networks: the VGG-16 [168] and the ResNet-50 [64].

The VGG-16 network is composed by 13 convolutional layers and 3 fully connected layers. The convolutional layers are divided in five convolutional blocks where, each of them is followed by a max-pooling layer with a stride of 2.

The ResNet-50, instead of having a series of stacked layers that process the input image as in common CNNs, performs a series of residual mappings between blocks composed by a few stacked layers. This is obtained using shortcut connections that realize an identity mapping, *i.e.* the input of the block is added to its output. Residual connections help to avoid the accuracy degradation problem [62] that occurs with the increase of the network depth, and are beneficial also in the saliency prediction case, since they improve the feature extraction capabilities of the network.

In particular, the ResNet-50 network consists of five convolutional blocks and a fully connected layer. The first block is composed by one convolutional layer followed by a max-pooling layer, both of them having a stride of 2, while the remaining four blocks are fully convolutional. All of these blocks, except the second one (`conv2`), reduce the dimension of feature maps with strides of 2.

Since the purpose of our network is to extract feature maps, we only consider convolutional layers and ignore fully connected layers which are present at the end of both networks. Moreover, it can be noticed that the downscaling factor of both of these architectures is particularly critical. For example, with an input image having a size of 240×320, the output dimension is 8×10, which is relatively small for the saliency prediction task. For this reason, we modify network structures to limit the rescaling phenomenon.

For the VGG-16 model, we also remove the last max-pooling layer and apply the aforementioned technique to the last but one pooling layer (see Figure 3.6a). On the contrary, for the ResNet-50 model we remove the stride and we introduce dilated convolutions in the last two blocks (see Figure 3.6b). In this case, since the technique is applied two times, we introduce holes of size 1 in the kernels of the block `conv4` and holes of size $2^2 - 1 = 3$ in the kernels of the block `conv5`. The output of the residual network is a tensor with 2048 channels. To limit the number of feature maps, we feed this tensor into another convolutional layer with 512 filters. Thanks to these expedients, our saliency maps are rescaled by a factor of 8 instead of 32 as in the original VGG-16 and ResNet-50 models. We include dilated convolutions also in prior layers, thus obtaining two convolutional layers with large receptive fields that allow us to capture the saliency of an object with respect to its neighborhood. We set the kernel size of these layers to 5 and the holes size to 3 achieving therefore a receptive field of 17×17. Strides of these layers are set to 1 and both of them are followed by a ReLU activation function.

The last layer of our model is a convolutional operation with one filter and a kernel size of 1 that extracts the final saliency map. Since the predicted map has lower dimensions than the original image, it is brought to its original size via bilinear upsampling.

3.2.2 Loss function

In order to capture several quality factors, saliency predictions are usually evaluated through different metrics. Inspired by this evaluation protocol, we introduce a new loss function given by a linear combination of three different saliency evaluation metrics. We define the overall loss function as follows:

$$\begin{split} L(\tilde{\mathbf{y}}, \mathbf{y}^{den}, \mathbf{y}^{fix}) = \\ \alpha L_1(\tilde{\mathbf{y}}, \mathbf{y}^{fix}) + \beta L_2(\tilde{\mathbf{y}}, \mathbf{y}^{den}) + \gamma L_3(\tilde{\mathbf{y}}, \mathbf{y}^{den}) \end{split} \tag{3.13}$$

where $\tilde{\mathbf{y}}$, $\mathbf{y}^{den}$ and $\mathbf{y}^{fix}$ are respectively the predicted saliency map, the ground-truth density distribution and the ground-truth binary fixation map, while α, β and γ are three scalars which balance the three loss functions. L_1, L_2 and L_3 are respectively the Normalized Scanpath Saliency, the Linear Correlation Coefficient and the Kullback-Leibler Divergence which are commonly used to evaluate saliency prediction models. These evaluation metrics are described in Sec. 3.3.2 and respectively defined in Eq. 3.14, 3.15, and 3.17.

In Section 3.4.3, we quantitatively justify the choice of our loss combination comparing our results with those obtained using single evaluation metrics as loss function. Moreover, we compare the proposed training strategy with several other probability distances used by previous saliency methods demonstrating that our solution is able to achieve a better balance among all evaluation metrics.

3.3 Evaluating saliency models

In the following, we describe datasets and evaluation metrics used for the saliency prediction task.

3.3.1 Datasets

Saliency prediction methods are usually evaluated on different saliency datasets which differ in terms of both image content and experimental settings.

SALICON [84]. This is the largest available dataset for saliency prediction. It contains $10,000$ training images, $5,000$ validation images and $5,000$ testing images, taken from the Microsoft COCO dataset [116]. Eye fixations are simulated with mouse movements: as shown in [84], there is a high degree of similarity between mouse-contingent saliency annotations and fixations recorded with eye-tracking systems. Ground-truth maps of the test set are not publicly available and predictions must be submitted to the SALICON challenge website[3] for evaluation.

In 2017, a new version of this dataset was released in which authors replaced the original velocity-based fixation detection algorithm, resulting in more eye-like fixations. In the following, we use both versions of the SALICON dataset and we refer to SALICON 2015 for the original release and to SALICON 2017 for the more recent version.

[3] https://competitions.codalab.org/competitions/3791

MIT1003 [89]. The MIT1003 dataset contains $1,003$ images coming from Flickr and LabelMe. Saliency maps have been created from eye-tracking data of 15 observers.

MIT300 [88]. The MIT300 dataset is a collection of 300 natural images with saliency maps generated from eye-tracking data of 39 users. Saliency maps of this entire dataset are held out and we used the MIT Saliency benchmark [21] for evaluating our predictions. To test our networks on this dataset, we fine-tune them on images of the MIT1003 randomly split in training and validation sets.

CAT2000 [17]. This dataset contains $4,000$ images coming from a large variety of categories such as *Cartoons*, *Art*, *Satellite*, *Low resolution images*, *Indoor*, *Outdoor*, *Line drawings*, etc. It is composed of 20 different categories with 200 images for each of them. Saliency maps of the testing set, composed by $2,000$ images, are not available and also in this case we submitted our predicted saliency maps to the MIT Saliency benchmark [21] for evaluation.

TORONTO [19]. It contains 120 color images from indoor and outdoor environments. Saliency maps have been created from eye-tracking data of 20 subjects.

PASCAL-S [113]. The dataset is composed of 850 natural images coming from the validation set of the PASCAL VOC 2010 [44], with eye fixations of 8 different subjects.

DUT-OMRON [221]. It consists of $5,168$ images with the largest height or width of 400 pixels. Saliency maps have been obtained from eye-tracking data of 5 observers. A post-processing step is applied to remove outlier eye fixation points that do not lie on a meaningful object.

3.3.2 Evaluation metrics

A large variety of metrics to evaluate saliency prediction models exist and the main difference between them concerns the ground-truth representation. In fact, saliency evaluation metrics can be categorized in location-based and distribution-based metrics [157, 22, 104]. The first category considers saliency maps at discrete fixation locations, while the second treats both ground-truth fixation maps and predicted saliency maps as continuous distributions.

The most widely used location-based metrics, which compare the predicted saliency map $\tilde{\mathbf{y}}$ with respect to the ground-truth binary fixation map $\mathbf{y}^{fix}$, are the Area under the ROC curve, in its different variants of Judd (AUC) and shuffled (sAUC), and the Normalized Scanpath Saliency (NSS). The AUC metric chooses non-fixation points with a uniform distribution, while the shuffled AUC uses human fixations of other images in the dataset as non-fixation distribution. In this way, centered distribution of human fixations of the dataset is taken into account. In general, the AUC metrics do not penalize low-valued false positives giving a high score for high-valued predictions placed at fixated locations and ignoring the others.

The NSS metric was defined specifically for the evaluation of saliency models [148]. The idea is to quantify the saliency map values at the eye fixation locations and to normalize it with the saliency map variance:

$$\mathit{NSS}\left(\tilde{\mathbf{y}}, \mathbf{y}^{fix}\right) = \frac{1}{N}\sum_{i}\frac{\tilde{\mathbf{y}}_i - \mu(\tilde{\mathbf{y}})}{\sigma(\tilde{\mathbf{y}})} \cdot \mathbf{y}_i^{fix} \tag{3.14}$$

where i indexes the i^{th} pixel, $N = \sum_i \mathbf{y}_i^{fix}$ is the total number of fixated pixels and $\tilde{\mathbf{y}}$ is normalized to have a zero mean and unit standard deviation. The NSS is sensitive in an equivalent manner to both false positives and false negatives.

For the distribution-based category, the most used evaluation metrics, which instead compare the predicted saliency map $\tilde{\mathbf{y}}$ with respect to the ground-truth density map $\mathbf{y}^{den}$, are the Linear Correlation Coefficient (CC), the Similarity (SIM), the Earth Mover Distance (EMD), and the Kullback-Leibler Divergence (KL-Div).

The CC metric is the Pearson's correlation coefficient and treats the saliency and ground-truth density maps as random variables measuring the linear relationship between them. It is computed as:

$$\mathit{CC}\left(\tilde{\mathbf{y}}, \mathbf{y}^{den}\right) = \frac{\sigma\left(\tilde{\mathbf{y}}, \mathbf{y}^{den}\right)}{\sigma\left(\tilde{\mathbf{y}}\right) \cdot \sigma\left(\mathbf{y}^{den}\right)} \tag{3.15}$$

where $\sigma\left(\tilde{\mathbf{y}}, \mathbf{y}^{den}\right)$ is the covariance of $\tilde{\mathbf{y}}$ and $\mathbf{y}^{den}$. It ranges between -1 and 1, and a score close to -1 or 1 indicates a perfect linear relationship between the two maps.

The Similarity metric [88] is computed as the sum of pixel-wise minimums between the predicted saliency map $\tilde{\mathbf{y}}$ and the ground-truth density map $\mathbf{y}^{den}$, after normalizing the two maps:

$$\mathit{SIM}\left(\tilde{\mathbf{y}}, \mathbf{y}^{den}\right) = \sum_{i} min(\tilde{\mathbf{y}}_{\mathbf{i}}, \mathbf{y}_i^{den}) \tag{3.16}$$

where i indexes the i^{th} pixel, $\tilde{\mathbf{y}}$ and $\mathbf{y}^{den}$ are supposed to be probability distributions and sum up to one. A similarity score of 1 indicates that the predicted map is identical to the ground-truth one.

The EMD represents the minimal cost to transform the probability distribution of the predicted saliency map $\tilde{\mathbf{y}}$ into the one of the ground-truth $\mathbf{y}^{den}$. Therefore, a larger EMD indicates a larger difference between the two maps. It penalizes false positives proportionally to the spatial distance from the ground-truth.

The KL Divergence evaluates the loss of information when the distribution $\tilde{\mathbf{y}}$ is used to approximate the distribution $\mathbf{y}^{den}$, therefore taking a probabilistic interpretation of saliency and ground-truth density maps. Formally:

$$\textit{KL-Div}\left(\tilde{\mathbf{y}}, \mathbf{y}^{den}\right) = \sum_i \mathbf{y}_i^{den} \log\left(\frac{\mathbf{y}_i^{den}}{\tilde{\mathbf{y}}_i + \epsilon} + \epsilon\right) \tag{3.17}$$

where i indexes the i^{th} pixel and ϵ is a regularization constant. The KL-Div is a dissimilarity metric and a lower value indicates a better approximation of the ground-truth by the predicted saliency map.

3.4 Experimental results

In this section, we provide experimental results of the two saliency models previously described. In particular, we demonstrate the effectiveness of the main components of the two networks by showing extensive analyses and experiments, and we compare their performance with respect to several state-of-the-art saliency prediction models. First, we describe training and implementation details of both models.

3.4.1 Training and implementation details

We evaluate our solutions on SALICON, MIT300 and CAT2000 datasets. For the first dataset, we train the networks on its training set and we use the $5,000$ validation images to validate the model. For the second and the third dataset, we pre-train the networks on SALICON 2015 dataset and then fine-tune them on MIT1003 dataset and CAT2000 training set respectively, as suggested by the MIT Saliency Benchmark organizers. In particular, to test our models on the MIT300 dataset, we use 903 randomly selected images of the MIT1003 to fine-tune the networks and the remaining 100 as validation set. For the CAT2000 dataset, instead, we randomly choose $1,800$ images of training set for the fine-tuning and we use the remaining 200 (10 for each category) as validation set.

ML-Net details. For training and testing this model, images from all datasets are zero-padded to fit a $4:3$ aspect ratio, and then resized to 640×480. During training, we use a batch size equal to 10 and the Stochastic Gradient Descent as optimizer, with Nesterov momentum 0.9, weight decay 0.0005 and learning rate 10^{-3}. Parameters α and λ of Eq. 3.2 are respectively set to 1.1 and $1/(w' \cdot h')$ in all experiments. Weights of the Feature Extraction Network are initialized with those of the VGG-16 model trained on ImageNet [163].

SAM details. For the SALICON, MIT1003 and MIT300 datasets, we resize input images to 240×320. Since images from MIT1003 and MIT300 have different sizes, we apply zero padding bringing images to have an aspect ratio of $4 : 3$ and then resize them to have the selected input size. Instead, images from CAT2000 dataset have all the same input size of 1080×1920. For this reason, we resize all images of this dataset to 180×320. Predictions of all datasets are slightly blurred with a Gaussian filter. After a validation process, we set the standard deviation of the Gaussian kernel to 7.

Weights of the Dilated Convolutional Networks are initialized with those of the VGG-16 and ResNet-50 models trained on ImageNet [163]. For the Attentive ConvLSTM, following the initialization proposed in [8], we initialize the recurrent weights matrices U_i, U_f, U_o and U_c as random orthogonal matrices. All W matrices and U_a are initialized by sampling each element from the Gaussian distribution of mean 0 and variance 0.05^2. The matrix V_a and all bias vectors are initialized to zero. Weights of all other convolutional layers of our model are initialized according to [50].

At training time, we randomly sample a minibatch containing 10 training samples, and encourage the network to minimize the proposed loss function through the RMSprop optimizer [184]. Batch normalization is preserved in the ResNet-50 part of the model, and we do not add batch normalization layers elsewhere. Loss parameters α, β, and γ (Eq. 3.13) are respectively set to -1, -2, and 10 balancing the contribution of each loss function. Differently from the KL-Div that is a dissimilarity metric and its value should be minimized, the CC and the NSS are to be maximized to predict better saliency maps. To this end, we set α and β as negative weights. The choice of these balance weights is driven by the goal of having good results on all evaluation metrics and by taking into account the numerical range that the single metrics have at convergence. During the training phase, we set the initial learning rate to 10^{-5} and we decrease it by a factor of 10 every two epochs for the model based on the ResNet, and every three epochs for that based on the VGG network.

3.4.2 Evaluation of ML-Net model

To assess the contribution of each CNN level to the final performance of our ML-Net model, described in Sec. 3.1, we perform a feature importance analysis.

Feature importance analysis

Our method relies on a non-linear combination of features extracted at different levels of a CNN. To validate our multilevel approach, we first evaluate the relative importance of features coming from each level. In the following, we define the importance of a feature as the extent to which a variation of the feature can affect the predicted map. Let us start by considering a linear model where different levels of a CNN are combined to obtain a pixel of the saliency map $\phi_i(\mathbf{x})$

$$\phi_i(\mathbf{x}) = w_i^T \mathbf{x} + \theta_i \tag{3.18}$$

where w_i and θ_i are the weight vector and the bias relative to pixel i, while $\mathbf{x}$ represents the activation coming from different levels of the feature extraction CNN, and $\phi_i(\mathbf{x})$ is the predicted saliency pixel. It is easy to see that the magnitude of elements in w_i defines the importance of the corresponding features. In the extreme case of a pixel of the feature map which is always multiplied by 0, it is straightforward to see that part of the feature map is ignored by the model, and has therefore no importance, while a pixel with high absolute values in w_i will have a considerable effect on the predicted saliency pixel.

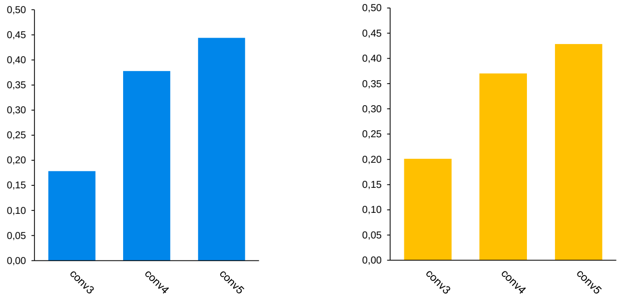
In our model, $\phi_i(\cdot)$ is a highly non-linear function of the input, due to the presence of the encoding network and of the prior, thus the above reasoning is not directly applicable. Instead, given an image $\mathbf{x}_j$, we can approximate $\phi_i(\mathbf{x}_j)$ in the neighborhood of $\mathbf{x}_j$ as follows

$$\phi_i(\mathbf{x}_j) \approx \nabla\phi_i(\mathbf{x}_j)^T \mathbf{x} + \theta \tag{3.19}$$

An intuitive explanation of this approximation is that the magnitude of the partial derivatives indicates which features need to be changed to affect the model. Also notice that Eq. 3.19 is equivalent to a first order Taylor expansion.

To get an estimation of the importance of each pixel in an activation map regardless of the choice of $\mathbf{x}_j$, we can average the element-wise absolute values of the gradient computed in the neighborhood of each validation sample.

$$w_i = \frac{1}{N}\sum_{j=1}^{N}\left[\left|\frac{\partial\phi_i}{\partial x^1}(\mathbf{x}_j)\right|, \left|\frac{\partial\phi_i}{\partial x^2}(\mathbf{x}_j)\right|, \cdots, \left|\frac{\partial\phi_i}{\partial x^d}(\mathbf{x}_j)\right|\right] \tag{3.20}$$



(a) Contribution to mean

(b) Contribution to variance

Figure 3.7: Contribution of features extracted from `conv3`, `conv4` and `conv5` to prediction mean and variance.

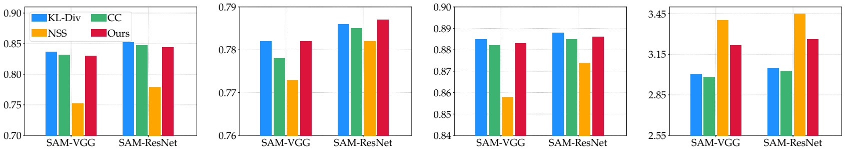
where d is the dimensionality of $\mathbf{x}_j$. Then, to get the relative importance of each activation map, we average the values of w_i corresponding to that map, and L_1 normalize the resulting importances.

To get an estimate of the importance of feature maps extracted from the CNN, we should compute $\phi_i(\mathbf{x}_j)$ for every test image j and for every saliency pixel i. To reduce the amount of required computation, instead of computing the gradient of each saliency pixel, we compute the gradient of the mean and variance of the output saliency map, in the neighborhood of each test sample. Applying Eq. 3.20, we get an indication of the contribution of each feature pixel to the mean and variance of the predicted map.

Figure 3.7 reports the relative importance of activation maps coming from `conv3`, `conv4` and `conv5` on the model trained on the SALICON 2015 dataset [84]. It is easy to notice that all features give a valuable contribution to the final result, and that while high level features are still the most relevant ones, medium level features have a considerable role in the prediction of the saliency map. This confirms our strategy to incorporate activations coming from different levels.

3.4.3 Evaluation of SAM model

In this section, we provide analyses and experiments to validate the contribution of each component of our Saliency Attentive Model, described in Sec. 3.2.



(a) CC (b) sAUC (c) AUC (d) NSS

Figure 3.8: Comparison between the proposed loss function and its components used individually as loss functions. We report results for both SAM-VGG and SAM-ResNet on SALICON 2015 validation set [84]. Each plot corresponds to a different evaluation metric (CC, sAUC, AUC and NSS), while the four color bars represent the loss functions used. As it can be observed, our loss function achieves the best balance between metrics.

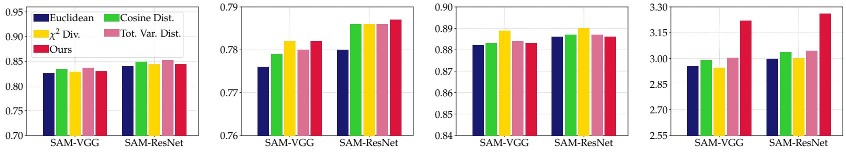
Comparison between different loss functions

In Fig. 3.8 we compare results obtained by using single loss functions (KL-Div, CC, NSS) and our combination defined in Eq. 3.13. Results are reported for both versions of our model on SALICON 2015 validation set. We call SAM-VGG the model based on the VGG network and SAM-ResNet that based on the ResNet network.

As it can be seen, our combined loss achieves on average better results on all metrics. In particular, when the model is trained using the KL-Div or the CC metrics as loss function, the performances are good especially on the CC, while the model fails on the NSS. When the model is trained using the NSS metric, instead, it achieves better results only on the NSS and fails on all other metrics.

To further validate the effectiveness of the proposed loss function, we compare it with traditional loss functions and probability distances used by other previous saliency models [101, 141, 82]. Fig. 3.9 shows the comparison between our combination of saliency metrics and four other loss functions: the Euclidean loss, the Cosine Distance, the χ^2 Divergence and the Total Variation Distance. Also in this case, our loss function achieves a better balance among all metrics. The gap with respect to all other traditional losses is particularly evident on the NSS metric, while, on all other metrics, the proposed combined loss, if it does not reach the best results, it is very close to them.

Overall, our combined loss reaches competitive results on all metrics differently from the other loss functions. For this reason, results of all following



(a) CC (b) sAUC (c) AUC (d) NSS

Figure 3.9: Comparison between the proposed combination of saliency metrics and more traditional loss functions such as Euclidean Loss, χ^2 Divergence, Cosine Distance and Total Variation Distance. Each plot corresponds to a different evaluation metric (CC, sAUC, AUC and NSS). The five color bars represent the performance of our model trained with the considered loss functions. We report results of both SAM-VGG and SAM-ResNet models on SALICON 2015 validation set [84].

experiments are obtained by training the SAM model with our combination of loss functions.

Model ablation analysis

We evaluate the contribution of each component of the architecture, on SALICON, MIT1003 and CAT2000 validation sets. To this end, we construct five different variations: the plain CNN architecture without the last fully convolutional layer (as a baseline), the Dilated Convolutional Network (DCN), the DCN with the proposed ConvLSTM model, the DCN with the proposed learned priors module, and the final version of our model with all its components.

Tables 3.1 and 3.2 show the results of the ablation analysis using both versions of our model on three different datasets. The results emphasize that the overall architecture is able to predict better saliency maps in both SAM-VGG and SAM-ResNet variants and each proposed component gives an important contribution to the final performance on all considered datasets. In particular, on the SALICON dataset, it can be seen that there is a constant improvement on all metrics. For example, the VGG baseline achieves a result of 0.743 in terms of CC, while the DCN achieves a relative improvement of $\frac{0.801-0.743}{0.743} = 7.8\%$. This result is further improved by 1% when adding the Attentive ConvLSTM or by 2.9% when adding the learned priors. The overall architecture adds an important improvement of 2.6% to the DCN with the Attentive ConvLSTM and 0.7% to the DCN with

Dataset	Model	SAM-VGG			
		CC	sAUC	AUC	NSS
SALICON	Plain CNN	0.743	0.765	0.870	2.333
	Dilated Convolutional Network	0.801	**0.786**	0.876	3.122
	DCN + Attentive ConvLSTM	0.809	0.784	0.878	3.142
	DCN + Learned Priors	0.824	0.782	0.882	3.209
	DCN + Attentive ConvLSTM + Learned Priors	**0.830**	0.782	**0.883**	**3.219**
MIT1003	Plain CNN	0.638	**0.625**	0.889	2.147
	Dilated Convolutional Network	0.718	0.596	0.906	2.704
	DCN + Attentive ConvLSTM	0.749	0.601	0.908	2.812
	DCN + Learned Priors	0.750	0.621	0.908	2.805
	DCN + Attentive ConvLSTM + Learned Priors	**0.757**	0.613	**0.910**	**2.852**
CAT2000	Plain CNN	0.751	0.546	0.862	1.886
	Dilated Convolutional Network	0.791	**0.548**	0.870	2.067
	DCN + Attentive ConvLSTM	0.851	0.537	0.874	2.253
	DCN + Learned Priors	0.877	0.532	0.876	2.328
	DCN + Attentive ConvLSTM + Learned Priors	**0.879**	0.530	**0.877**	**2.347**

Table 3.1: Ablation analysis of SAM-VGG model on SALICON 2015 [84], MIT1003 [89], and CAT2000 [17] validation sets.

learned priors. The ResNet baseline, instead, achieves a CC result of 0.771 that is improved by a 6.7% when adding the dilated convolutions. The Attentive ConvLSTM and the learned priors respectively add an improvement of 2.2% and 2.1%. These results are further improved using the overall architecture with all proposed components by 0.4% and 0.5%.

It is also noteworthy that, with our pipeline, a VGG-based network and a ResNet-based network achieve almost the same performance, so one of the two models can be equally chosen according to speed and memory allocation needs, without considerably affecting prediction performance.

Fig. 3.10 shows some saliency maps predicted by our SAM-ResNet model and by only some of its main components with respect to the ground-truth. As it can be seen, there is a constant improvement of predictions which, by adding our key components, are more qualitatively similar to the ground-truth.

Contribution of the attentive model and learned priors

Table 3.3 reports the performance of our model when using the output of the Attentive ConvLSTM module at different timesteps as input for the rest of the

Dataset	Model	SAM-ResNet			
		CC	sAUC	AUC	NSS
SALICON	Plain CNN	0.771	0.762	0.876	2.404
	Dilated Convolutional Network	0.823	0.774	0.879	3.187
	DCN + Attentive ConvLSTM	0.841	0.786	0.885	3.256
	DCN + Learned Priors	0.840	0.784	0.885	3.235
	DCN + Attentive ConvLSTM + Learned Priors	**0.844**	**0.787**	**0.886**	**3.260**
MIT1003	Plain CNN	0.667	**0.631**	0.895	2.255
	Dilated Convolutional Network	0.748	0.609	0.902	2.845
	DCN + Attentive ConvLSTM	0.756	0.613	0.912	2.860
	DCN + Learned Priors	0.746	0.613	0.908	2.816
	DCN + Attentive ConvLSTM + Learned Priors	**0.768**	0.617	**0.913**	**2.893**
CAT2000	Plain CNN	0.819	**0.538**	0.870	2.052
	Dilated Convolutional Network	0.881	0.527	0.877	2.368
	DCN + Attentive ConvLSTM	0.882	0.528	0.878	2.367
	DCN + Learned Priors	0.885	0.528	0.878	**2.377**
	DCN + Attentive ConvLSTM + Learned Priors	**0.888**	0.534	**0.879**	2.375

Table 3.2: Ablation analysis of SAM-ResNet model on SALICON 2015 [84], MIT1003 [89], and CAT2000 [17] validation sets.

model. Results clearly show that the refinement carried out by the Attentive model results in better performance. No further significant improvements were observed for $t > 4$: while CC, sAUC and AUC almost saturated, NSS slightly decreased after four iterations.

To assess the effectiveness of our prior learning strategy, we compare it with the prior strategy described in Sec. 3.1.1, in which a low resolution prior map is learned and applied element-wise to the predicted saliency map, after performing bilinear upsampling. It is worthwhile to note that our ML-Net and SAM models are the only two attempts to incorporate the center bias in a deep learning model without the use of hand-crafted prior maps. Results on SALICON validation set are reported in Table 3.4. Using multiple Gaussian learned priors, instead of learning an entire prior map, with no pre-defined structure, shows to be beneficial according to all metrics.

3.4.4 Comparison with state of the art

We quantitatively compare our two models with state-of-the-art competitors on SALICON, MIT300 and CAT2000 test sets. Not all saliency methods report ex-

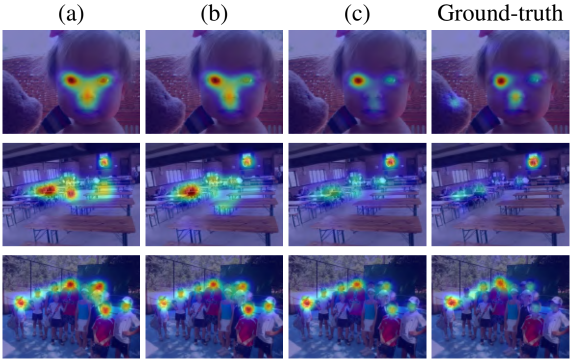


Figure 3.10: Examples of saliency maps predicted by the DCN (a), the DCN with the Attentive ConvLSTM (b), and the DCN with the Attentive ConvLSTM and learned priors (c) compared with the ground-truth (d).

perimental results on all considered datasets. For this reason, comparison methods are different depending on each dataset. We decide to sort model performances by the NSS metric as suggested by the MIT Saliency Benchmark [21, 22, 104].

Table 3.5 shows the results on the SALICON 2015 dataset in terms of CC, sAUC, AUC and NSS. As it can be observed, our SAM-ResNet outperforms all competitors by a big margin especially on CC and NSS metrics and obtains the best result also on the sAUC. In particular, our method overcomes the other ResNet-based model [118] with an improvement of 1.5% according to NSS metric, 1.3% and 0.4% according to CC and sAUC. For a fair comparison with other methods, we also include the results achieved by our SAM-VGG model. The improvement with respect all other VGG-based methods is even more significant than that obtained by the SAM-ResNet model. In details, our SAM-VGG overcomes all other VGG-based methods with an improvement of 12.7% and 5.6% according to NSS and CC metrics. Regarding our ML-Net model, it achieves the second best performance in terms of the NSS metric with respect to all other VGG-based competitors.

We also test our SAM model on the latest version of the SALICON dataset. The results are shown in Table 3.6 in comparison with different competitors. Also in this case, our model is able to achieve the best performance on all considered saliency metrics overcoming all other methods by a big margin. As expected, the ResNet version obtains better results than the VGG-based model. Nevertheless,

	T	CC	sAUC	AUC	NSS
SAM-VGG	1	0.821	0.777	**0.884**	3.168
	2	0.827	0.777	0.883	3.224
	3	0.828	0.781	0.883	**3.226**
	4	**0.830**	**0.782**	0.883	3.219
SAM-ResNet	1	0.785	0.737	0.879	3.050
	2	0.829	0.764	**0.886**	3.214
	3	0.842	0.779	**0.886**	3.256
	4	**0.844**	**0.787**	**0.886**	**3.260**

Table 3.3: Results on SALICON validation set 2015 [84] when using the output of the Attentive ConvLSTM module at different timesteps as input of the rest of the model.

	CC	sAUC	AUC	NSS
SAM-VGG (single prior)	0.811	**0.783**	0.878	3.150
SAM-VGG (multiple learned priors)	**0.830**	0.782	**0.883**	**3.219**
SAM-ResNet (single prior)	0.840	0.785	0.884	3.249
SAM-ResNet (multiple learned priors)	**0.844**	**0.787**	**0.886**	**3.260**

Table 3.4: Comparison results between multiple learned priors and the single prior map used in our ML-Net model. Results are reported on SALICON 2015 validation set [84].

the version based on VGG-16 is still able to surpass the competitors on all metrics, thus further confirming the effectiveness of our solution.

The results on MIT300 and CAT2000 datasets are respectively reported in Tables 3.7 and 3.8. Our SAM model achieves state-of-the-art results on all metrics, except for the sAUC, on the CAT2000 dataset surpassing other methods by an important margin especially on SIM, CC, NSS and EMD metrics. On the MIT300 dataset, instead, we obtain results very close to the best ones. Our SAM model does not obtain a big gain in performance on AUC metrics. This can be explained considering that the AUC metrics are primarily based on true positives without significantly penalizing false positives. For this reason, hazy or blurred saliency maps like the ones predicted by [105] achieve high AUC values [18, 238], despite being visually very different from the ground-truth annotations.

Starting from our SAM model trained on the two releases of the SALICON, we also evaluate the effectiveness of the proposal on other three popular saliency

	CC	sAUC	AUC	NSS
SAM-ResNet	**0.842**	**0.779**	0.883	**3.204**
DSCLRCN [118]	0.831	0.776	0.884	3.157
SAM-VGG	0.825	0.774	0.881	3.143
ML-Net	0.743	0.768	0.866	2.789
MixNet [37]	0.730	0.771	0.861	2.767
SU [101]	0.780	0.760	0.880	2.610
SalGAN [140]	0.781	0.772	0.781	2.459
SalNet [141]	0.622	0.724	0.858	1.859
DeepGazeII [105]	0.509	0.761	**0.885**	1.336

Table 3.5: Comparison results on SALICON 2015 test set [84]. The results in bold indicate the best performing method on each evaluation metric. Methods are sorted by the NSS metric.

	CC	SIM	AUC	NSS
SAM-ResNet	**0.899**	**0.793**	**0.865**	**1.990**
SAM-VGG	0.891	0.786	0.864	1.971
EAD [65]	0.871	0.760	0.852	1.896
SalGAN [140]	0.844	0.728	0.857	1.816
SALICON [75]	0.659	0.600	0.808	1.557
SalNet [141]	0.763	0.639	0.840	1.555

Table 3.6: Comparison results on SALICON 2017 test set [84]. Methods are sorted by the NSS metric. Results of comparison methods are from [65].

datasets: TORONTO [19], PASCAL-S [113] and DUT-OMRON [221]. For a fair comparison with other methods, we do not finetune our model on a subset of these datasets. The comparison results are reported in Table 3.9. Again, we observe that our model is able to quantitatively overcome the drawbacks of different existing proposals. As a side note, here the performance of the VGG-based model is often very similar to that of the ResNet-based one. Also, it shall be observed that the 2017 version of SALICON shows better generalization capabilities on all metrics with the exception of the NSS metric. This can be partially explained by the fact that the ground-truth maps of SALICON 2015 are less blurred than in the second version of the dataset: this helps the NSS measure, which normalizes the prediction to have zero mean and unit variance, thus increasing the weight of predicted pixels when the prediction is less blurred.

	SIM	CC	sAUC	AUC	NSS	EMD	KL-Div
DSCLRCN [118]	**0.68**	**0.80**	0.72	0.87	**2.35**	2.17	0.95
SAM-ResNet	**0.68**	0.78	0.70	0.87	2.34	2.15	1.27
SAM-VGG	0.67	0.77	0.71	0.87	2.30	2.14	1.13
DeepFix [100]	0.67	0.78	0.71	0.87	2.26	**2.04**	0.63
SALICON [75]	0.60	0.74	**0.74**	0.87	2.12	2.62	**0.54**
PDP [82]	0.60	0.70	0.73	0.85	2.05	2.58	0.92
ML-Net	0.59	0.67	0.70	0.85	2.05	2.63	1.10
SalGAN [140]	0.63	0.73	0.72	0.86	2.04	2.29	1.07
DVA [207]	0.58	0.68	0.71	0.85	1.98	3.06	0.64
iSEEL [182]	0.57	0.65	0.68	0.84	1.78	2.72	0.65
SalNet [141]	0.52	0.58	0.69	0.83	1.51	3.31	0.81
BMS [235]	0.51	0.55	0.65	0.83	1.41	3.35	0.81
Mr-CNN [119]	0.48	0.48	0.69	0.79	1.37	3.71	1.08
DeepGazeII [105]	0.46	0.52	0.72	**0.88**	1.29	3.98	0.96
GBVS [61]	0.48	0.48	0.63	0.81	1.24	3.51	0.87
Judd [89]	0.42	0.47	0.60	0.81	1.18	4.45	1.12
eDN [199]	0.41	0.45	0.62	0.82	1.14	4.56	1.14

Table 3.7: Comparison results on MIT300 dataset [88]. The results in bold indicate the best performing method on each evaluation metric. Methods are sorted by the NSS metric.

	SIM	CC	sAUC	AUC	NSS	EMD	KL-Div
SAM-ResNet	**0.77**	**0.89**	0.58	**0.88**	**2.38**	**1.04**	0.56
SAM-VGG	0.76	**0.89**	0.58	**0.88**	**2.38**	1.07	0.54
DeepFix [100]	0.74	0.87	0.58	0.87	2.28	1.15	**0.37**
MixNet [37]	0.66	0.76	0.58	0.86	1.92	1.63	0.62
iSEEL [182]	0.62	0.66	**0.59**	0.84	1.67	1.78	0.92
BMS [235]	0.61	0.67	**0.59**	0.85	1.67	1.95	0.83
eDN [199]	0.52	0.54	0.55	0.85	1.30	2.64	0.97
Judd [89]	0.46	0.54	0.56	0.84	1.30	3.60	0.94
GBVS [61]	0.51	0.50	0.58	0.80	1.23	2.99	0.80

Table 3.8: Comparison results on CAT2000 test set [17]. The results in bold indicate the best performing method on each evaluation metric. Methods are sorted by the NSS metric.

3.4.5 Qualitative results

Qualitative results obtained by our models on SALICON 2015 and MIT1003 validations sets, together with those of other state-of-the-art models, are respect-

	DUT-OMRON				TORONTO				PASCAL-S			
	CC	SIM	AUC	NSS	CC	SIM	AUC	NSS	CC	SIM	AUC	NSS
Itti [81]	0.46	0.39	0.83	1.54	0.48	0.45	0.80	1.30	0.42	0.36	0.82	1.30
GBVS [61]	0.53	0.43	0.87	1.71	0.57	0.49	0.83	1.52	0.45	0.36	0.84	1.36
eDN [199]	-	-	-	1.33	0.50	0.40	0.85	1.25	-	-	-	1.42
Mr-CNN [119]	-	-	-	-	0.49	0.47	0.80	1.41	-	-	-	-
DVA [207]	0.67	0.53	0.91	3.09	0.72	0.58	**0.86**	2.12	0.66	0.52	0.89	2.26
SAM-VGG$_{2015}$	0.65	0.53	0.91	2.91	0.69	0.59	**0.86**	2.14	0.72	0.60	**0.90**	**2.48**
SAM-VGG$_{2017}$	0.69	0.53	0.91	2.95	**0.74**	**0.63**	**0.86**	**2.15**	0.73	**0.61**	0.89	2.31
SAM-ResNet$_{2015}$	0.69	**0.56**	0.91	**3.21**	0.69	0.59	**0.86**	2.12	0.69	0.59	0.89	2.34
SAM-ResNet$_{2017}$	**0.70**	0.54	**0.92**	2.97	**0.74**	0.62	**0.86**	2.14	**0.74**	**0.61**	**0.90**	2.34

Table 3.9: Comparison results on DUT-OMRON [221], TORONTO [19] and PASCAL-S [113] dataset. Results of comparison methods are from [207]. The subscript notations $_{2015}$ and $_{2017}$ indicate that the models are trained on the SALICON 2015 and SALICON 2017, respectively.

ively shown in Fig. 3.11 and 3.12. As it can be noticed, our networks are able to predict high saliency values on people, faces, objects and other predominant cues. The SAM model also produces good saliency maps when images do not contain strong saliency regions, such as when saliency is concentrated in the center of the scene or when images portray a landscape. We also notice that the model can sometimes infer the relative importance of different people in the same scene, a human behaviour which saliency models still struggle to replicate, as discussed in [23].

To visually highlight the differences between the two versions of the SALICON dataset, we report in Fig. 3.13 some qualitative results of our SAM-ResNet model on both versions of the dataset. As it can be seen, saliency maps of the newest version of the dataset are in general more blurred and less focused on specific areas of the images. Nevertheless, our model is able to predict saliency maps that are visually very similar to the ground-truth in both scenarios, thus confirming its effectiveness also from a qualitative point of view.

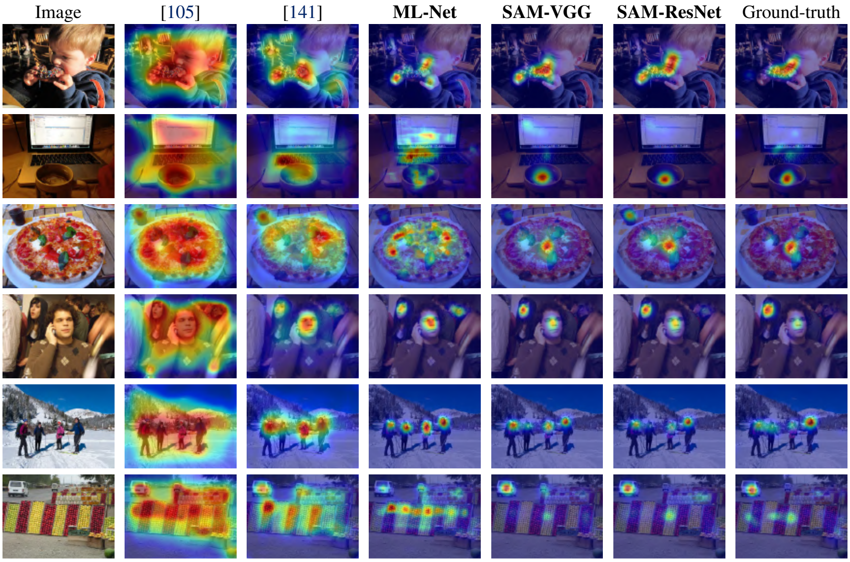


Figure 3.11: Qualitative results and comparison with other state-of-the-art models on SALICON 2015 validation set [84].

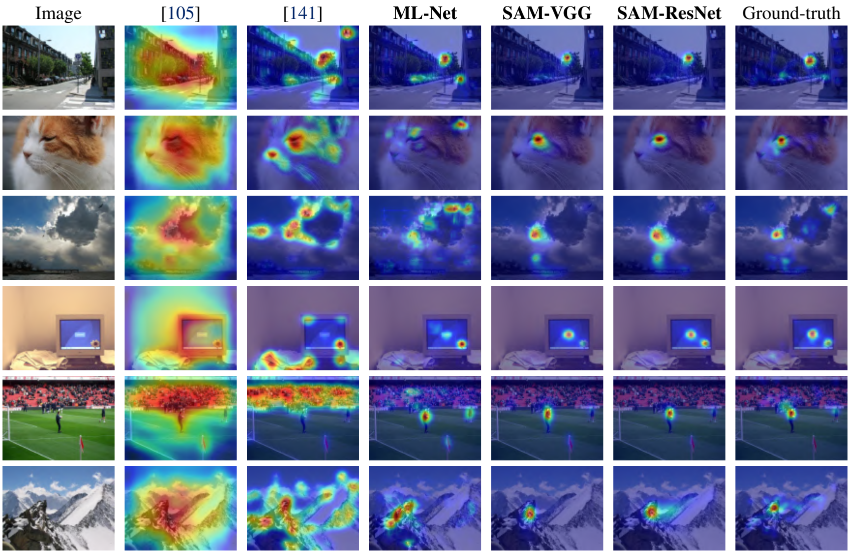


Figure 3.12: Qualitative results and comparison with other state-of-the-art models on MIT1003 validation set [89].

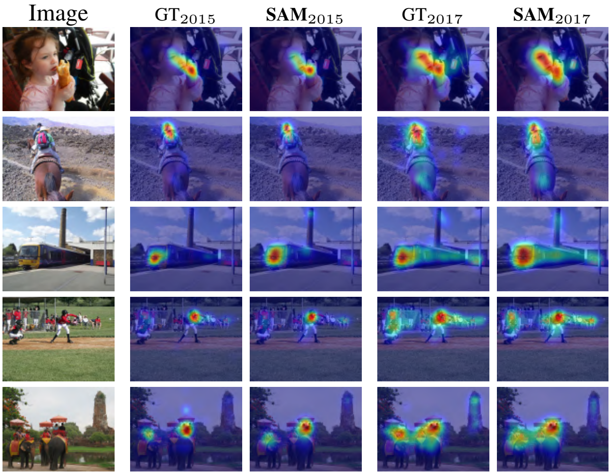


Figure 3.13: Qualitative results on both 2015 and 2017 releases of the SALICON dataset [84].

Chapter 4

Image captioning

A core problem in computer vision and artificial intelligence is that of building a system that can replicate the human ability of understanding a visual stimuli and describing it in natural language. Indeed, this kind of system would have a great impact on society, opening up to a new progress in human-machine interaction and collaboration. Recent advancements in computer vision and machine translation, together with the availability of large datasets, have made it possible to generate natural sentences describing images.

This task, which is called image captioning, has been recently gaining much attention thanks to the spread of deep learning architectures which can effectively describe images in natural language [201, 90, 217, 202]. Image captioning approaches are usually capable of learning a correspondence between an input image and a probability distribution over time, from which captions can be sampled either using a greedy decoding strategy [202], or more sophisticated techniques like beam search and its variants [3].

Contributions

In this chapter, we present two different solutions to address the image captioning task. The first model exploits the conditioning given by a saliency prediction model on which parts of the image are salient and which are contextual, during the generation of the sentence. In particular, it incorporates an attentive mechanism

This chapter is related to publications [4, 11, 24, 25] reported in Appendix B, by the author of the thesis. See Appendix B for details.

that can focus on different parts of the input image while weighing the contribution of salient and contextual regions. We demonstrate through extensive analyses and experiments the effectiveness of incorporating saliency information in a captioning model with respect to different baselines and other saliency-based captioning approaches.

While the language model of the first solution is based on recurrent neural networks, the second approach is instead based on the Transformer model that abandons the recurrent relations in favour of the use of fully-attentive mechanisms. The proposed architecture, called Meshed-Memory Transformer, improves both the image encoding and the language generation steps: it learns a multi-level representation of the relationships between image regions integrating learned a priori knowledge, and uses a mesh-like connectivity at decoding stage to exploit low- and high-level features. Experimentally, we investigate the performance of our solution and different fully-attentive models in comparison with recurrent ones. We show that our model achieves a new state of the art on the most important dataset for image captioning (*i.e.* COCO [116]) reaching the first place on the leaderboard of the online test server[1].

In details, the rest of the chapter is organized as follows: in Sec. 4.1, we introduce our first captioning model that incorporates saliency prediction to enhance image descriptions. After motivating the use of saliency information into a captioning model, we describe the architecture of our solution and we show its effectiveness with quantitative and qualitative experiments. Then, in Sec. 4.2, we investigate the use of fully-attentive models and we propose a second captioning architecture that is based on the Transformer model and achieves a new state of the art on standard image captioning.

[1] https://competitions.codalab.org/competitions/3221

4.1 Boosting captioning with saliency

While the progress of captioning techniques is encouraging, the human ability in the construction and formulation of a sentence is still far from being adequately emulated in today's image captioning systems. When humans describe a scene, they look at an object before naming it in a sentence [57], and they do not focus on each region with the same intensity, as selective mechanisms attract their gaze on salient and relevant parts of the scene [156]. Also, they care about the context using peripheral vision, so that the description of an image alludes not only to the main objects in the scene, and to how they relate to each other, but also to the context in which they are placed in the image.

As seen in the previous chapter, an intensive research effort has been carried out in the computer vision community to predict where humans look in an image. Indeed, we have demonstrated that, with the advent of deep neural networks and large annotated datasets, saliency prediction techniques have obtained impressive results generating maps that are very close to the ones computed with eye-tracking devices. Despite the encouraging progress in image captioning and visual saliency, and their close connections, these two fields of research have remained almost separate. In fact, only few attempts have been recently presented in this direction [173, 183]. In particular, Sugano *et al.* [173] presented a gaze-assisted attention mechanism for image caption based on human eye fixations (*i.e.* the static states of gaze upon a specific location). Although this strategy confirms the importance of using eye fixations, it requires gaze information from a human operator. Therefore, it can not be applied on general visual data archives, in which this information is missing. To overcome this limit, Tavakoli *et al.* [183] presented an image captioning method based on saliency maps, which can be automatically predicted from the input image.

In this section, we present an approach which incorporates saliency prediction to effectively enhance the quality of image description. We propose a captioning architecture based on recurrent neural networks which can focus on different regions of the input image by means of an attentive mechanism. This attentive behaviour, differently from previous works [216], is conditioned by two different attention paths: the former focused on salient spatial regions, predicted by a saliency model, and the latter focused on contextual regions, which are computed as well from saliency maps. Experimental results on public image captioning datasets demonstrate that our solution is able to properly exploit saliency cues. Also, we show that this is done without losing the key properties of the generated captions, such as their diversity and the vocabulary size. By visualizing the states

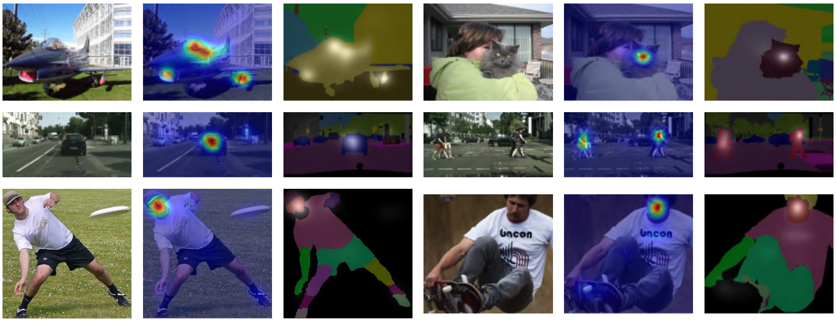


Figure 4.1: Ground-truth semantic segmentation and saliency predictions from our SAM model (Sec. 3.2) on sample images from Pascal-Context [136] (first row), Cityscapes [31] (second row) and LIP [53] (last row).

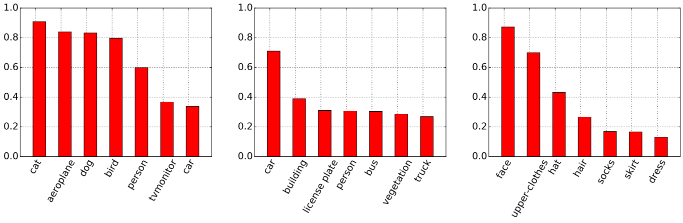
of both attentive paths, we finally show that the trained model has learned to attend to both salient and contextual regions during the generation of the caption, and that attention focuses produced by the network effectively correspond, step by step, to generated words.

4.1.1 What is hit by saliency?

As previously mentioned, human gazes are attracted by both low-level cues such as color, contrast and texture, and high-level concepts such as faces and text [89, 23]. Current state-of-the-art saliency prediction methods are able to effectively incorporate all these factors and predict saliency maps which are very close to those obtained from human eye fixations. In this section, we qualitatively investigate which parts of an image are actually hit or ignored by saliency models, by jointly analyzing saliency and semantic segmentation maps. This motivates the need of using saliency predictions as an additional conditioning for captioning models.

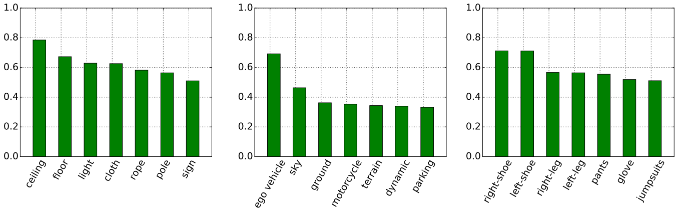
To compute saliency maps, we employ our Saliency Attentive Model described in Section 3.2, which has shown good results on popular saliency benchmarks. It is worthwhile to mention, anyway, that the qualitative conclusions of this section can be applied to any state-of-the-art saliency model.

Since semantic segmentation algorithms are not always completely accurate, we perform the analysis on three semantic segmentation datasets, in which regions



(a) Pascal-Context [136] (b) Cityscapes [31] (c) LIP [53]

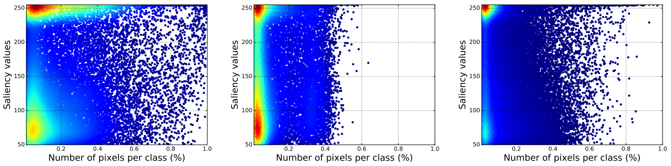
Figure 4.2: Most salient classes on Pascal-Context, Cityscapes and LIP datasets.



(a) Pascal-Context [136] (b) Cityscapes [31] (c) LIP [53]

Figure 4.3: Least salient classes on Pascal-Context, Cityscapes and LIP datasets.

have been segmented by human annotators: Pascal-Context [136], Cityscapes [31], and the Look into Person (LIP) [53]. While the first one contains natural images without a specific target, the other two are focused, respectively, on urban streets and human body parts. In particular, the Pascal-Context provides additional annotations for the Pascal VOC 2010 dataset [44] which contains $10,103$ training and validation images and $9,637$ testing images. It goes beyond the original Pascal semantic segmentation task by providing annotations for the whole scene, and images are annotated by using more than 400 different labels. The Cityscapes dataset, instead, is composed by a set of video sequences recorded in street scenes from 50 different cities. It provides high quality pixel-level annotations for $5,000$ frames and coarse annotations for $20,000$ frames. The dataset is annotated with 30 street-specific classes, such as *car*, *road*, *traffic sign*, etc. Finally, the LIP dataset is focused on the semantic segmentation of people and provides more than $50,000$ images annotated with 19 semantic human part labels. Images contain



(a) Pascal-Context [136] (b) Cityscapes [31] (c) LIP [53]

Figure 4.4: Distribution of object sizes and saliency values (best seen in color).

person instances cropped from the Microsoft COCO dataset [116] and split in training, validation and testing sets with $30,462$, $10,000$ and $10,000$ images respectively. For our analyses we only consider train and validation images for the Pascal-Context and LIP datasets, and the $5,000$ pixel-level annotated frames for the Cityscapes dataset. Figure 4.1 shows, for some sample images, the predicted saliency map and the corresponding semantic segmentation on the three datasets.

We firstly investigate which are the most and the least salient classes for each dataset. Since there are semantic classes with a low number of occurrences with respect to the total number of images, we only consider relevant semantic classes (*i.e.* classes with at least N occurrences). Due to the different dataset sizes, we set N to 500 for the Pascal-Context and LIP datasets, and to 200 for the Cityscapes dataset. To collect the number of times that the predicted saliency hits a semantic class, we binarize each map by thresholding the values of its pixels. A low threshold value leads to a binarized map with dilated salient regions, while an high threshold creates small salient regions around the fixation points. For this reason, we use two different threshold values to analyze the most and the least salient classes. We choose a threshold near 0 to find the least salient classes for each dataset, and a value near 255 to find instead the most salient ones.

Figures 4.2 and 4.3 show the most and the least salient classes in terms of the percentage of times that saliency hits a region belonging to a class. As it can be seen, there are different distributions depending on the considered dataset. For example, for the Pascal-Context, the most salient classes are animals (such as cats, dogs and birds), people and vehicles (such as airplanes and cars), while the least salient ones result to be ceiling, floor and light. As for the Cityscapes dataset, cars are absolutely the most salient class with a 70% of times in which is hit by saliency. All other classes, instead, do not reach the 40%. On the LIP dataset, the most salient classes are all human body parts in the upper body, while the least salient ones are all in the lower body. As expected, people faces are those most hit by saliency with an absolute number of occurrences near to 90%. It can be observed as a general pattern that the most important or visible objects in the scene are hit by saliency, while objects in the background, and the context itself of the image are usually ignored. This leads to the hypothesis that both salient and non salient regions are important to generate the description of an image, given that we generally want the context to be included in the caption, and that the distinction between salient regions and context, given by a saliency prediction model, can improve captioning results.

We also investigate the existence of a relation between the size of an object and its saliency values. In Figure 4.4, we plot the joint distribution of object sizes and saliency values on the three datasets, where the size of an object is simply computed as the number of its pixels normalized by the size of the image. As it can be seen, most of the low saliency instances are small; however, high saliency values concentrate on small objects as well as on large ones. In summary, there is not always a proportionality between the size of an object and its saliency, so the importance of an object can not be assessed by simply looking at its size. In the image captioning scenario that we want to tackle, larger objects correspond to larger activations in the last layers of a convolutional architecture, while smaller objects correspond to smaller activations. Since salient and non salient regions can have comparable activations, the supervision given by a saliency prediction model on whether a pixel belongs or not to a salient region can be beneficial during the generation of the caption.

4.1.2 Saliency and context aware attention

Following the qualitative findings of the previous section, we develop a model in which saliency is exploited to enhance image captioning. Here, a generative recurrent neural network is conditioned, step by step, on salient spatial regions, predicted by a saliency model, and on contextual features which account for the role on non-salient regions in the generation of the caption. In the following, we describe the overall model. An overview is presented in Figure 4.5.

Each input image I is firstly encoded through a Fully Convolutional Network, which provides a stack of high-level features on a spatial grid $\{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_L\}$, each corresponding to a spatial location of the image. At the same time, we extract a saliency map for the input image using our SAM model (Sec. 3.2), and downscale it to fit the spatial size of the convolutional features, so to obtain a spatial grid $\{s_1, s_2, ..., s_L\}$ of salient regions, where $s_i \in [0, 1]$. Correspondingly, we also

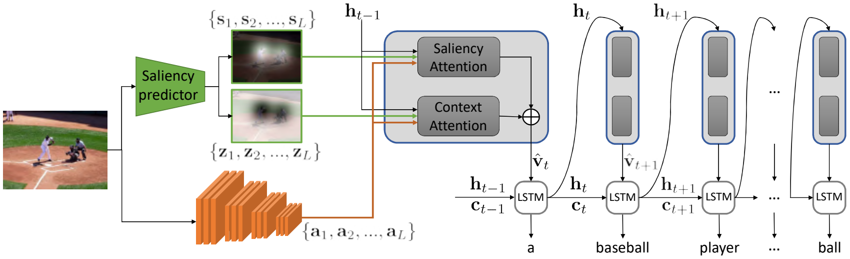


Figure 4.5: Overview of the proposed model. Two different attention paths are built for salient regions and contextual regions, to help the model build captions which describe both components (best seen in color).

define a spatial grid of contextual regions, $\{z_1, z_2, ..., z_L\}$ where $z_i = 1 - s_i$. Under the model, visual features at different locations will be selected or inhibited according to their saliency value.

The generation of the caption is carried out word-by-word by feeding and sampling words from an LSTM layer, which, at every timestep, is conditioned on features extracted from the input image and on the saliency map. Formally, the behaviour of the generative LSTM is driven by the following equations:

$$\mathbf{i}_t = \sigma(W_{vi}\hat{\mathbf{v}}_t + W_{wi}\mathbf{w}_t + W_{hi}\mathbf{h}_{t-1} + \mathbf{b}_i) \tag{4.1}$$

$$\mathbf{f}_t = \sigma(W_{vf}\hat{\mathbf{v}}_t + W_{wf}\mathbf{w}_t + W_{hf}\mathbf{h}_{t-1} + \mathbf{b}_f) \tag{4.2}$$

$$\mathbf{o}_t = \sigma(W_{vo}\hat{\mathbf{v}}_t + W_{wo}\mathbf{w}_t + W_{ho}\mathbf{h}_{t-1} + \mathbf{b}_o) \tag{4.3}$$

$$\mathbf{g}_t = \phi(W_{vg}\hat{\mathbf{v}}_t + W_{wg}\mathbf{w}_t + W_{hg}\mathbf{h}_{t-1} + \mathbf{b}_g) \tag{4.4}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t \tag{4.5}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \phi(\mathbf{c}_t) \tag{4.6}$$

where, at each timestep, $\hat{\mathbf{v}}_t$ denotes the visual features extracted from I, by considering the map of salient regions $\{s_i\}_i$, and those of contextual regions $\{z_i\}_i$. $\mathbf{w}_t$ is the input word, and $\mathbf{h}$ and $\mathbf{c}$ are respectively the internal state and the memory cell of the LSTM. $\odot$ denotes the element-wise Hadamard product, σ is the sigmoid function, ϕ is the hyperbolic tangent `tanh`, W_* are learned weight matrices and $\mathbf{b}_*$ are learned biases vectors.

To provide the generative network with visual features, we draw inspiration from the machine attention literature [216] and compute the fixed-length feature vector $\hat{\mathbf{v}}_t$ as a linear combination of spatial features $\{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_L\}$ with time-varying weights α_{ti}, normalized over the spatial extent via a *softmax* operator:

$$\hat{\mathbf{v}}_t = \sum_{i=1}^{L} \alpha_{ti}\mathbf{a}_i, \tag{4.7}$$

$$\alpha_{ti} = \frac{\exp\left(e_{ti}\right)}{\sum_{k=1}^{L} \exp\left(e_{tk}\right)}. \tag{4.8}$$

At each timestep the attention mechanism selects a region of the image, based on the previous LSTM state, and feeds it to the LSTM, so that the generation of a word is conditioned on that specific region, instead of being driven by the entire image.

Ideally, we want weights α_{ti} to be aware of the saliency and contextual value of location $\mathbf{a}_i$, and to be conditioned on the current status of the LSTM, which can be well encoded by its internal state $\mathbf{h}_t$. In this way, the generative network can focus on different locations of the input image according to their belonging to a salient or contextual region, and to the current generation state. Of course, simply multiplying attention weights with saliency values would result in a loss of context, which is fundamental for caption generation. We instead split attention weights e_{ti} into two contributions, one for saliency and one for context regions, and employ two different fully connected networks to learn the two contributions (Figure 4.5). Conceptually, this is equivalent to building two separate attention paths, one for salient regions and for contextual regions, which are merged to produce the final attention. Overall, the model obeys to the following equation:

$$e_{ti} = s_i \cdot e_{ti}^{sal} + z_i \cdot e_{ti}^{ctx} \tag{4.9}$$

where e_{ti}^{sal} and e_{ti}^{ctx} are, respectively, the attention weights for salient and context regions. Attention weights for saliency and context are computed as follows:

$$e_{ti}^{sal} = v_{e,sal}^T \cdot \phi(W_{ae,sal} \cdot \mathbf{a}_i + W_{he,sal} \cdot \mathbf{h}_{t-1}) \tag{4.10}$$

$$e_{ti}^{ctx} = v_{e,ctx}^T \cdot \phi(W_{ae,ctx} \cdot \mathbf{a}_i + W_{he,ctx} \cdot \mathbf{h}_{t-1}) \tag{4.11}$$

Notice that our model learns different weights for saliency and contextual regions, and combines them into a final attentive map in which the contributions of salient and non-salient regions are merged together. Similarly to the classical Soft Attention approach [216], the proposed generative LSTM can focus on every region of the image, but the attentive process is aware of the saliency of each location, so that the focus on salient and contextual regions is driven by the output of the saliency predictor.

Sentence generation

Words are encoded with one-hot vectors having size equal to that of the vocabulary, and are then projected into an embedding space via a learned linear transformation. Because sentences have different lengths, they are also marked with special begin-of-string and end-of-string tokens, to keep the model aware of the beginning and end of a particular sentence.

Given an image and sentence $(\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_T)$, encoded with one-hot vectors, the generative LSTM is conditioned step by step on the first t words of the caption, and is trained to produce the next word of the caption. The objective function which we optimize is the log-likelihood of correct words over the sequence

$$\max_{\mathbf{w}} \sum_{t=1}^{T} \log \Pr(\mathbf{y}_t | \hat{\mathbf{v}}_t, \mathbf{y}_{t-1}, \mathbf{y}_{t-2}, ..., \mathbf{y}_0) \tag{4.12}$$

where $\mathbf{w}$ are all the parameters of the model. The probability of a word is modeled via a softmax layer applied on the output of the LSTM. To reduce the dimensionality, a linear embedding transformation is used to project one-hot word vectors into the input space of the LSTM and, vice versa, to project the output of the LSTM to the dictionary space.

$$\Pr(\mathbf{y}_t | \hat{\mathbf{v}}_t, \mathbf{y}_{t-1}, \mathbf{y}_{t-2}, ..., \mathbf{y}_0) \propto \exp(\mathbf{y}_t^T W_p \mathbf{h}_t) \tag{4.13}$$

where W_p is a matrix for transforming the LSTM output space to the word space and $\mathbf{h}_t$ is the output of the LSTM.

At test time, the LSTM is given a begin-of-string tag as input for the first timestep, then the most probable word according to the predicted distribution is sampled and given as input for the next timestep, until an end-of-string tag is predicted.

4.1.3 Experimental settings

Datasets and metrics

To validate the effectiveness of the proposed Saliency and Context aware Attention, we perform experiments on five popular image captioning datasets: SALICON [84], Microsoft COCO [116], Flickr8k [69], Flickr30k [229], and PASCAL-50S [193].

Microsoft COCO is composed by more than $120,000$ images divided in training and validation sets, where each of them is provided with at least five sentences generated by using Amazon Mechanical Turk. SALICON is a subset of this one, created for the visual saliency prediction task. Since its images are from the Microsoft COCO dataset, at least five captions for each image are available. Overall, it contains $10,000$ training images, $5,000$ validation images and $5,000$ testing images where eye fixations for each image are simulated with mouse movements. In our experiments, we only use train and validation sets for both datasets. The Flickr8k and the Flickr30k datasets are composed by $8,000$ and $30,000$ images respectively. Both of them come with five annotated sentences for each image. In our experiments, we randomly choose $1,000$ validation images and $1,000$ test images for each of these two datasets. The PASCAL-50S dataset provides additional annotations for the UIUC PASCAL sentences [153]. It is composed of $1,000$ images from the PASCAL-VOC dataset, each of them annotated with 50 human-written sentences, instead of 5 as in the original dataset. Due to the limited number of samples and for a fair comparison with other captioning methods, we first pre-train the model on the Microsoft COCO dataset, then we test it on the images of this dataset without a specific fine-tuning.

For evaluation, we employ four automatic metrics which are usually employed in image captioning: BLEU [144], ROUGE$_L$ [115], METEOR [9] and CIDEr [193]. BLEU is a modified form of precision between n-grams to compare a candidate translation against multiple reference translations. We evaluate our predictions with BLEU using mono-grams, bi-grams, three-grams and four-grams. ROUGE$_L$ computes an F-measure considering the longest co-occurring in sequence n-grams. METEOR, instead, is based on the harmonic mean of unigram precision and recall, with recall weighted higher than precision. It also has several features that are not found in other metrics, such as stemming and synonymy matching, along with the standard exact word matching. CIDEr, finally, computes the average cosine similarity between n-grams found in the generated caption and those found in reference sentences, weighting them using TF-IDF. To ensure a fair evaluation, we use the Microsoft COCO evaluation toolkit[2] to compute all scores.

Implementation details

Each image is encoded through a convolutional network, which computes a stack of high-level features. We employ the popular ResNet-50 [64], trained on ImageNet [163], to compute the feature maps over the input image. In particular, the ResNet-50 is composed by 49 convolutional layers, divided in 5 convolutional

[2] https://github.com/tylin/coco-caption

blocks, and 1 fully connected layer. Since we want to maintain the spatial dimensions, we extract the feature maps from the last convolutional layer and ignore the fully connected layer. The output of the ResNet model is a tensor with 2048 channels. To limit the number of feature maps and the number of learned parameters, we fed this tensor into another convolutional layer with 512 filters and a kernel size of 1, followed by a ReLU activation function. Differently from the weights of the ResNet-50 which are kept fixed, the weights of this last convolutional layer are initialized according to [50] and fine-tuned over the considered datasets. In the LSTM, following the initialization proposed in [8], the weight matrices applied to the inputs are initialized by sampling each element from the Gaussian distribution of 0 mean and 0.01^2 variance, while the weight matrices applied to the internal states are initialized by using the orthogonal initialization. The vectors $v_{e,sal}$ and $v_{e,ctx}$ as well as all bias vectors $\mathbf{b}_*$ are instead initialized to zero.

As mentioned before, to predict the saliency map for each input image, we exploit our Saliency Attentive Model (SAM), which is able to predict accurate saliency maps according to different saliency benchmarks. We note however, that we do not expect a significant performance variation when using other state-of-the-art saliency methods.

In our experiments, we use different input image sizes depending on the considered dataset. For the SALICON dataset, since its images have all the same size of 480×640, we keep the original size of these images, thus obtaining $L = 15 \times 20 = 300$. For all other datasets, which are composed of images with different sizes, we set the input size to 480×480 obtaining $L = 15 \times 15 = 225$. Since saliency maps are exploited inside the proposed saliency-context attention model, we resize the SALICON saliency maps to have a size of 15×20 while, for all other datasets, we resize them to 15×15.

All experiments are performed by using the Adam optimizer [93] with Nestorov momentum [178] using an initial learning rate of 0.001 and batch size 64. The hidden state dimension is set to 1024 while the embedding size to 512. For all datasets, we choose a vocabulary size equal to the number of words which appear at least 5 times in training and validation captions.

4.1.4 Comparison with baselines

To assess the performance of our method, and to investigate the hypotheses behind it, we first compare with the classic Soft Attention approach, and we then build three baselines in which saliency is used to condition the generative process.

Soft Attention [216]. The visual input to the LSTM is computed via the Soft

Attention mechanism to attend at different locations of the image, without considering salient and non-salient regions. A single feed forward network is in charge of producing attention values, which can be obtained by replacing Eq. 4.9 with

$$e_{ti} = v_e^T \cdot \phi(W_{ae} \cdot \mathbf{a}_i + W_{he} \cdot \mathbf{h}_{t-1}). \tag{4.14}$$

This approach is equivalent to the one proposed in [216], although some implementation details are different. In order to achieve a fair evaluation, we use activations from the ResNet-50 model instead of the VGG-19, and we do not include the doubly stochastic regularization trick. For this reason, the numerical results that we report are not directly comparable with those in the original paper (ours are in general higher than the original ones).

Saliency pooling. Visual features from the CNN are multiplied at each location by the corresponding saliency value, and then summed, without any attention mechanism. In this case the visual input of the LSTM is not time dependent, and salient regions are given more focus than non-salient ones. Comparing with Eq. 4.7, it can be seen as a variation of the Soft Attention in which the network always focuses on salient regions.

$$\hat{\mathbf{v}}_t = \hat{\mathbf{v}} = \sum_{i=1}^{L} s_i \mathbf{a}_i \tag{4.15}$$

Attention on saliency. This is an extension of the Soft Attention approach in which saliency is used to modulate attention values at each location. The attention mechanism, therefore, is conditioned to attend salient regions with higher probability, and to ignore non-salient regions.

$$e_{ti} = s_i \cdot v_e^T \cdot \phi(W_{ae} \cdot \mathbf{a}_i + W_{he} \cdot \mathbf{h}_{t-1}) \tag{4.16}$$

Attention on saliency and context (with weight sharing). The attention mechanism is aware of salient and context regions, but weights used to compute the attentive scores of salient and context are shared, excluding the v_e^T vectors. Notice that, if those were shared too, this baseline would be equivalent to the Soft Attention one.

$$e_{ti} = s_i \cdot e_{ti}^{sal} + (1 - s_i) \cdot e_{ti}^{ctx} \tag{4.17}$$

$$e_{ti}^{sal} = v_{e,sal}^T \cdot \phi(W_{ae} \cdot \mathbf{a}_i + W_{he} \cdot \mathbf{h}_{t-1}) \tag{4.18}$$

$$e_{ti}^{ctx} = v_{e,ctx}^T \cdot \phi(W_{ae} \cdot \mathbf{a}_i + W_{he} \cdot \mathbf{h}_{t-1}) \tag{4.19}$$

Dataset	Model	B-1	B-2	B-3	B-4	M	R	C
SALICON	Soft Attention	69.0	50.9	36.1	25.4	22.5	49.9	70.8
	Saliency+Context Attention	**69.2**	**51.4**	**37.2**	**26.9**	**22.9**	**50.4**	**73.3**
COCO	Soft Attention	70.6	53.0	38.3	27.5	24.3	51.8	87.9
	Saliency+Context Attention	**70.8**	**53.6**	**39.1**	**28.4**	**24.8**	**52.1**	**89.8**
Flick8k (validation)	Soft Attention	59.9	41.8	27.9	18.2	19.8	45.0	47.7
	Saliency+Context Attention	**62.8**	**44.5**	**30.2**	**19.9**	**20.3**	**46.5**	**50.1**
Flickr8k (test)	Soft Attention	61.0	43.2	29.6	20.1	20.8	46.5	53.2
	Saliency+Context Attention	**63.5**	**45.6**	**31.5**	**21.2**	**21.1**	**47.5**	**54.1**
Flickr30k (validation)	Soft Attention	**61.9**	**43.3**	29.7	20.2	19.9	44.8	43.2
	Saliency+Context Attention	61.3	**43.3**	**30.1**	**20.9**	**20.2**	**45.0**	**44.5**
Flickr30k (test)	Soft Attention	**61.9**	43.4	29.9	20.5	19.8	44.5	44.2
	Saliency+Context Attention	61.5	**43.8**	**30.5**	**21.3**	**20.0**	**45.2**	**46.4**
PASCAL-50S	Soft Attention	**82.4**	70.0	57.0	45.1	32.8	65.9	70.4
	Saliency+Context Attention	**82.4**	**70.2**	**57.5**	**45.7**	**32.9**	**66.3**	**70.7**

Table 4.1: Image captioning results. The conditioning of saliency and context (`Saliency+Context Attention`) enhances the generation of the caption with respect to the traditional machine attention mechanism. `Soft Attention` here indicates our reimplementation of [216], using the same visual features of our model.

It is straightforward also to notice that our proposed approach is equivalent to the last baseline, without weight sharing.

In Table 4.1 we first compare the performance of our method with respect to the Soft Attention approach, to assess the superior performance of the proposal with respect to the published state of the art. We report results on all the datasets, both on validation and test sets, with respect to all the evaluation metrics previously described. As it can be seen, the proposed approach always overcomes by a significant margin the Soft Attention approach, thus experimentally confirming the benefit of having two separate attention paths, one for salient and one for non-salient regions, and the role of saliency as a conditioning for captioning. In particular, on the METEOR metric, the relative improvement ranges from $\frac{32.9-32.8}{32.8} = 0.30\%$ on the PASCAL-50S to $\frac{20.3-19.8}{19.8} = 2.53\%$ of the Flickr8k validation set.

In Table 4.2, instead, we compare our approach with the three baselines that incorporate saliency. Firstly, it can be observed that the Saliency pooling baseline

Dataset	Model	B-1	B-2	B-3	B-4	M	R	C
SALICON	Saliency pooling	66.1	47.8	33.7	24.0	21.1	47.9	62.4
	Attention on saliency	68.8	51.3	37.0	26.5	22.7	50.1	71.3
	Saliency+Cont. Att. (weight sh.)	68.9	51.3	36.8	26.3	22.6	50.2	71.4
	Saliency+Context Attention	**69.2**	**51.4**	**37.2**	**26.9**	**22.9**	**50.4**	**73.3**
COCO	Saliency pooling	68.6	50.9	36.3	25.8	23.3	50.2	81.4
	Attention on saliency	70.4	53.2	38.6	27.6	24.1	51.6	86.6
	Saliency+Cont. Att. (weight sh.)	70.4	53.1	38.8	28.2	24.7	**52.1**	89.4
	Saliency+Context Attention	**70.8**	**53.6**	**39.1**	**28.4**	**24.8**	**52.1**	**89.8**
Flickr8k (Validation)	Saliency pooling	56.1	37.7	24.3	15.6	18.3	42.8	37.0
	Attention on saliency	58.7	40.4	26.8	17.6	19.7	45.1	44.7
	Saliency+Cont. Att. (weight sh.)	62.0	43.9	29.6	19.8	20.2	45.7	**50.2**
	Saliency+Context Attention	**62.8**	**44.5**	**30.2**	**19.9**	**20.3**	**46.5**	50.1
Flickr8k (Test)	Saliency pooling	56.5	37.8	24.6	16.2	18.5	42.9	37.7
	Attention on saliency	59.6	42.2	28.7	19.5	20.7	46.1	50.1
	Saliency+Cont. Att. (weight sh.)	62.4	44.2	29.9	19.7	21.1	46.7	51.7
	Saliency+Context Attention	**63.5**	**45.6**	**31.5**	**21.2**	**21.1**	**47.5**	**54.1**
Flickr30k (Validation)	Saliency pooling	58.7	40.5	27.1	18.4	18.3	43.0	34.2
	Attention on saliency	**63.0**	**44.5**	**30.8**	**21.3**	19.4	44.7	43.5
	Saliency+Cont. Att. (weight sh.)	62.0	43.8	30.0	20.5	19.7	44.6	43.3
	Saliency+Context Attention	61.3	43.3	30.1	20.9	**20.2**	**45.0**	**44.5**
Flickr30k (Test)	Saliency pooling	58.3	40.6	27.5	18.6	18.7	43.0	36.2
	Attention on saliency	**62.5**	**44.2**	**30.5**	21.0	19.6	44.9	45.0
	Saliency+Cont. Att. (weight sh.)	61.7	43.7	30.0	20.4	19.6	44.2	43.1
	Saliency+Context Attention	61.5	43.8	**30.5**	**21.3**	**20.0**	**45.2**	**46.4**
PASCAL-50S	Saliency pooling	79.9	67.1	53.6	41.8	31.4	64.1	65.3
	Attention on saliency	**82.4**	**70.3**	57.4	45.5	32.7	**66.3**	70.2
	Saliency+Cont. Att. (weight sh.)	82.0	69.7	56.4	44.2	32.7	65.2	70.0
	Saliency+Context Attention	**82.4**	70.2	**57.5**	**45.7**	**32.9**	**66.3**	**70.7**

Table 4.2: Comparison with image captioning with saliency baselines. While the use of machine attention strategies is beneficial (see Saliency pooling vs. Attention on Saliency), saliency and context are both important for captioning. The use of different attention paths for saliency and context also enhances the performance (see Saliency+Context Attention (with weight sharing) vs. Saliency+Context Attention).

usually performs worse than the Soft Attention, thus demonstrating that always attending to salient locations is not sufficient to achieve good captioning results. When plugging in attention, like in the Saliency Attention baseline, numerical

Dataset	Model	B-4	M	R	C
SALICON	Sugano *et al.* [173]	24.5	21.9	**52.4**	63.8
	Ours	**26.9**	**22.9**	50.4	**73.3**
COCO	Tavakoli *et al.* [183] (GBVS)	**28.7**	23.5	51.2	84.1
	Tavakoli *et al.* [183] (iSEEL)	28.3	23.5	50.8	83.6
	Ours	28.4	**24.8**	**52.1**	**89.8**
Flickr30k (Test)	Ramanishka *et al.* [151]	–	18.3	–	–
	Ours	21.3	**20.0**	45.2	46.4
PASCAL-50S	Tavakoli *et al.* [183] (GBVS)	40.0	30.2	63.5	61.5
	Tavakoli *et al.* [183] (iSEEL)	39.6	30.2	63.2	61.4
	Ours	**45.7**	**32.9**	**66.3**	**70.7**

Table 4.3: Comparison with existing saliency-boosted captioning models.

results are a bit higher, thanks to a time-dependent attention, but still far from the performance achieved by the complete model. It can also be noticed that, even though this baseline does not take into account the context, it sometimes achieves better results than the Soft Attention model (such as in the case of SALICON, with respect to the METEOR metric). Finally, we notice that the baseline with attention on saliency and context, and with weight sharing, is better than Saliency Attention, further confirming the benefit of including the context. Having two completely separated attention paths, such as in our model, is anyway important, as demonstrated by the numerical results of this last baseline with respect to those of our method.

4.1.5 Comparison with state of the art

We also compare to existing captioning models that incorporate saliency during the generation of image descriptions. In particular, we compare to the model proposed in [173], which exploited human fixation points, to the work by Tavakoli *et al.* [183] which reports experiments on Microsoft COCO and on PASCAL-50S, and to the proposal by Ramanishka *et al.* [151] which used convolutional activations as a proxy for saliency.

Table 4.3 shows the results on the three considered datasets in term of BLEU@4, METEOR, ROUGE$_L$ and CIDEr. We compare our solutions to both versions of the model presented in [183]. The GBVS version exploits saliency maps calculated by using a traditional bottom-up model [61], while the other one includes saliency maps extracted from a deep convolutional network [182].

Overall, results show that the proposed Saliency and Context Attention model can overcome the other methods on different metrics, thus confirming the strategy of including two attention paths. In particular, on the METEOR metric, we obtain a relative improvement of 4.57% on the SALICON dataset, 5.53% on the Microsoft COCO and 8.94% on the PASCAL-50S.

4.1.6 Analysis of generated captions

We further collect statistics on captions generated by our method and by the Soft Attention model, to quantitatively assess the quality of generated captions. Firstly, we define three metrics which evaluate the vocabulary size and the difference between the corpus of captions generated by the two models and the ground-truth:

- *Vocabulary size*: number of unique words generated in all captions;
- *Percentage of novel sentences*: percentage of generated sentences which are not seen in the training set;
- *Percentage of different sentences*: percentage of images which are described differently by the two models;

Then, we measure the diversity of the set of captions generated by each of the two models, via the following two metrics [167]:

- *Div-1*: ratio of number of unique unigrams in a set of captions to the number of words in the same set. Higher is more diverse.
- *Div-2*: ratio of number of unique bigrams in a set of captions to the number of words in the same set. Higher is more diverse.

In Table 4.4 we compare the set of captions generated by our model with that generated by the Soft Attention baseline. Although our model features a slight reduction of the vocabulary size on SALICON, COCO and PASCAL-50S, captions generated by the two models are very often different, thus confirming that the two approaches have learned different captioning models. Moreover, the diversity and the number of novel sentences of the Soft Attention approach are entirely preserved.

Dataset	Model	Div-1	Div-2	Vocab.	% novel sent.	% different sent.
SALICON	Soft Attention	0.0136	0.0498	658	95.22%	95.34%
	Saliency+Context Attention	0.0125	0.0549	614	93.12%	
COCO	Soft Attention	0.0038	0.0187	1490	81.81%	93.80%
	Saliency+Context Attention	0.0037	0.0182	1444	78.02%	
Flickr8k (Validation)	Soft Attention	0.0367	0.1026	389	98.30%	97.90%
	Saliency+Context Attention	0.0400	0.1094	411	99.30%	
Flickr8k (Test)	Soft Attention	0.0385	0.1041	404	98.50%	97.60%
	Saliency+Context Attention	0.0419	0.1119	423	99.60%	
Flickr30k (Validation)	Soft Attention	0.0577	0.1445	699	99.90%	98.62%
	Saliency+Context Attention	0.0565	0.1439	715	99.61%	
Flickr30k (Test)	Soft Attention	0.0580	0.1508	682	99.90%	98.20%
	Saliency+Context Attention	0.0585	0.1549	711	99.70%	
PASCAL-50S	Soft Attention	0.0475	0.1379	465	97.10%	94.80%
	Saliency+Context Attention	0.0468	0.1359	456	96.40%	

Table 4.4: Statistics on vocabulary size and diversity of the generated captions. Including saliency and context in two different machine attention paths (`Saliency+Context attention`) produces different captions with respect to the traditional machine attention approach (`Soft Attention`), while preserving almost the same diversity statistics.

4.1.7 Analysis of attentive states

The selection of a location in our model is based on the competition between the saliency attentive path and the context attentive path (see Eq. 4.9). To investigate how the two paths interact and contribute to the generation of a word, in Figure 4.6 we report, for several images from the Microsoft COCO dataset, the changes in attention weights between the two paths. Specifically, for each image we report the average of e_{ti}^{sal} and e_{ti}^{ctx} values at each timestep, along with a visualization of its saliency map. It is interesting to see how the model was able to correctly exploit the two attention paths for generating different parts of the caption, and how generated words correspond in most cases to the attended regions. For example, in the case of the first image ("a group of zebras graze in a grassy field"), the saliency attentive path is more active than the context path during the generation of words corresponding to the "group of zebras", which are captured by saliency. Instead, when the model has to describe the context ("in a grassy field"), the saliency attentive path has lower weights with respect to the context attentive path. The same can be observed for all the reported images; it can also be noticed that

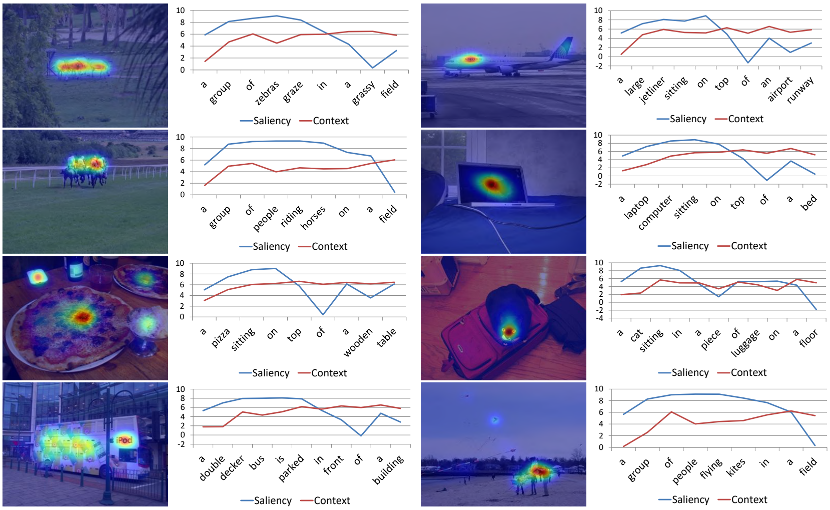


Figure 4.6: Examples of attention weights changes between saliency and context along with the generation of captions (best seen in color). Images are from the Microsoft COCO dataset [116].

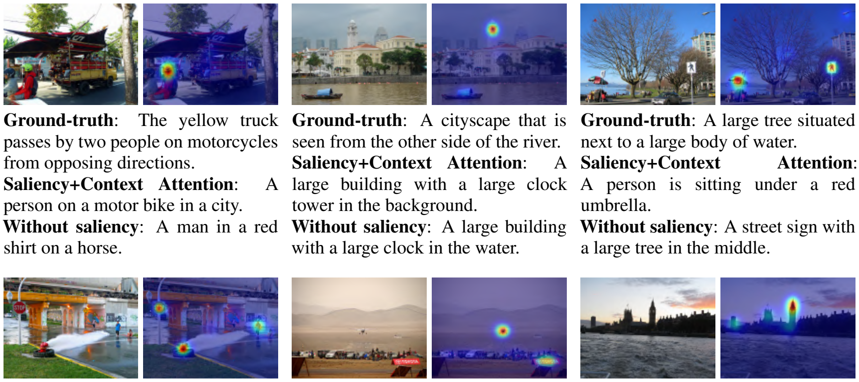
generated captions tend to describe both salient objects and the context, and that usually the salient part, which is also the most important, is described before the context.

4.1.8 Qualitative results

Finally, in Figure 4.8 we report some sample results on images taken from the COCO dataset. For each image we report the corresponding saliency map, and captions generated by our model and by the Soft Attention baseline compared to the ground-truth. It can be seen that, on average, captions generated by our model are more consistent with the corresponding image and the human-generated caption, and that, as also observed in the previous section, salient parts are described as well as the context. The incorporation of saliency and context also helps the model to avoid failures due to hallucination, such as in the case of the fourth image, in which the Soft Attention model predicts a remote control which is not depicted in the image. Other failure cases, which are avoided by our model, include the repetition of words and the failure to describe the context. We speculate that the presence of two separate attention paths, which the model has learned to attend during the generation of the caption, helps to avoid such failures more effectively than the classic machine attention approach.

Failure cases

For completeness, some failure cases of the proposed model are reported in Figure 4.7. The majority of failures occurs when the salient regions of the image are not described in the corresponding ground-truth caption (as for example in the first row), thus causing a performance loss. Some problems arise also in presence of complex scenes (such as in the fourth image). However, we observe that the Soft Attention baseline fails as well to predict correct and complete captions in these cases.



Ground-truth: The yellow truck passes by two people on motorcycles from opposing directions.
Saliency+Context Attention: A person on a motor bike in a city.
Without saliency: A man in a red shirt on a horse.

Ground-truth: A cityscape that is seen from the other side of the river.
Saliency+Context Attention: A large building with a large clock tower in the background.
Without saliency: A large building with a large clock in the water.

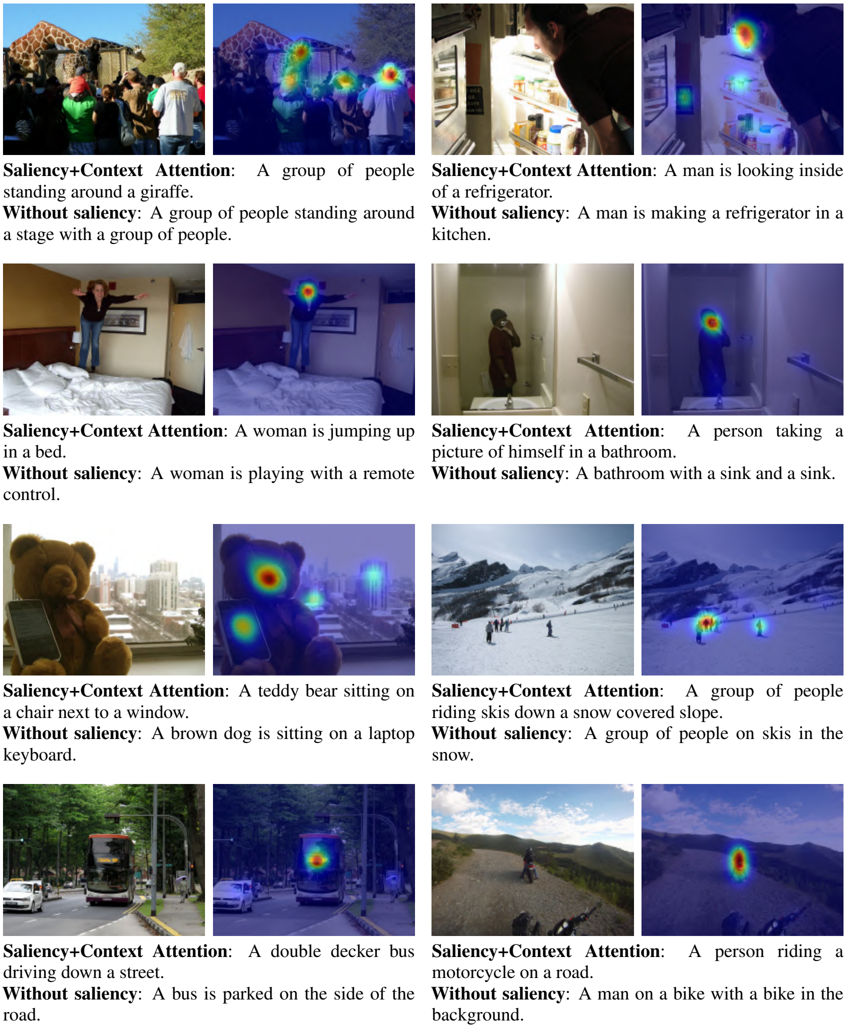
Ground-truth: A large tree situated next to a large body of water.
Saliency+Context Attention: A person is sitting under a red umbrella.
Without saliency: A street sign with a large tree in the middle.

Ground-truth: A busted fire hydrant spewing water out onto a street.
Saliency+Context Attention: A person standing in a front of a large cruise ship.
Without saliency: A man is standing in a dock near a large truck.

Ground-truth: A small airplane flying over a field filled with people.
Saliency+Context Attention: A group of people walking around a large jet.
Without saliency: A large group of people standing on top of a lush green field.

Ground-truth: The view of city buildings is seen from the river.
Saliency+Context Attention: A large clock tower towering over the water.
Without saliency: A large building with a large clock tower in the water.

Figure 4.7: Failure cases on sample images of the Microsoft COCO dataset [116].



Saliency+Context Attention: A group of people standing around a giraffe.
Without saliency: A group of people standing around a stage with a group of people.

Saliency+Context Attention: A man is looking inside of a refrigerator.
Without saliency: A man is making a refrigerator in a kitchen.

Saliency+Context Attention: A woman is jumping up in a bed.
Without saliency: A woman is playing with a remote control.

Saliency+Context Attention: A person taking a picture of himself in a bathroom.
Without saliency: A bathroom with a sink and a sink.

Saliency+Context Attention: A teddy bear sitting on a chair next to a window.
Without saliency: A brown dog is sitting on a laptop keyboard.

Saliency+Context Attention: A group of people riding skis down a snow covered slope.
Without saliency: A group of people on skis in the snow.

Saliency+Context Attention: A double decker bus driving down a street.
Without saliency: A bus is parked on the side of the road.

Saliency+Context Attention: A person riding a motorcycle on a road.
Without saliency: A man on a bike with a bike in the background.

Figure 4.8: Sample results on the Microsoft COCO dataset [116].

4.2 Boosting captioning with fully-attentive models

In the previous section, we have introduced a captioning architecture that during the generation of the image description combines two attentive paths, one for salient regions and the other for context. In the last few years, however, attentive models have been improved by replacing this type of attention over a grid of features with attention over image regions coming from an object detector [4, 204, 237]. In these models, the generative process attends a set of regions which are softly selected while generating the caption.

Regarding the language model, the use of recurrent neural networks has remained the dominant approach, with the exception of the investigation of convolutional language models [6], which however did not become a leading choice. The recent advent of fully-attentive models, in which the recurrent relation is abandoned in favour of the use of self-attention, offers unique opportunities in terms of set and sequence modeling performances, as testified by the Transformer [192] and BERT [35] models and their applications to retrieval [172] and video understanding [175]. Also, this setting offers novel architectural modeling capabilities, as for the first time the attention operator is used in a multi-layer and extensible fashion. Nevertheless, the multimodal nature of image captioning demands for specific architectures, different from those employed for the understanding of a single modality.

Following these premises, in this section we investigate the design of a novel fully-attentive approach for image captioning. Our architecture takes inspiration from the Transformer model [192] for machine translation and incorporates two key novelties with respect to all previous image captioning algorithms: (*i*) image regions and their relationships are encoded in a multi-level fashion, in which low-level and high-level relations are taken into account. When modeling these relationships, our model can learn and encode a priori knowledge by using persistent *memory vectors*. (*ii*) The generation of the sentence, done with a multi-layer architecture, exploits both low- and high-level visual relationships instead of having just a single input from the visual modality. This is achieved through a learned gating mechanism, which weights multi-level contributions at each stage. As this creates a mesh connectivity schema between encoder and decoder layers, we name our model *Meshed-Memory Transformer* – $\mathcal{M}^2$ Transformer for short. Figure 4.9 depicts a schema of the architecture.

Experimentally, we explore different fully-attentive baselines and recent proposals, gaining insights on the performance of fully-attentive models in image captioning. Our $\mathcal{M}^2$ Transformer, when tested on the COCO benchmark, achieves

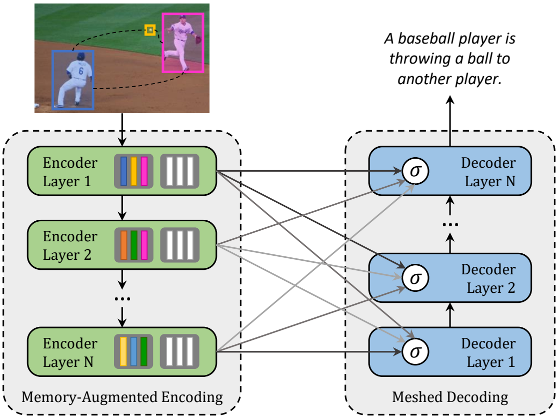


Figure 4.9: Our image captioning approach encodes relationships between image regions exploiting learned a priori knowledge. Multi-level encodings of image regions are connected to a language decoder through a meshed and learnable connectivity.

a new state of the art on the "Karpathy" test set, on both single-model and ensemble configurations. Most importantly, it surpasses existing proposals on the online test server, ranking first among published algorithms.

To foster researches in this field and the reproducibility of our work, the source code and trained models of our $\mathcal{M}^2$ Transformer are publicly available[3].

4.2.1 Meshed-Memory Transformer

Our model can be conceptually divided into an encoder and a decoder module, both made of stacks of attentive layers. While the encoder is in charge of processing regions from the input image and devising relationships between them, the decoder reads from the output of each encoding layer to generate the output caption word by word. All intra-modality and cross-modality interactions between word and image-level features are modeled via scaled dot-product attention, without using recurrence. Attention operates on three sets of vectors, namely a set of queries $\boldsymbol{Q}$, keys $\boldsymbol{K}$, and values $\boldsymbol{V}$, and takes a weighted sum of value vectors according to a similarity distribution between query and key vectors. In the case of scaled

[3] https://github.com/aimagelab/meshed-memory-transformer

dot-product attention, the operator can be formally defined as

$$\mathsf{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d}}\right)\boldsymbol{V}, \tag{4.20}$$

where $\boldsymbol{Q}$ is a matrix of n_q query vectors, $\boldsymbol{K}$ and $\boldsymbol{V}$ both contain n_k keys and values, all with the same dimensionality, and d is a scaling factor.

Memory-augmented encoder

Given a set of image regions $\boldsymbol{X}$ extracted from an input image, attention can be used to obtain a permutation invariant encoding of $\boldsymbol{X}$ through the self-attention operations used in the Transformer [192]. In this case, queries, keys, and values are obtained by linearly projecting the input features, and the operator can be defined as

$$\mathcal{S}(\boldsymbol{X}) = \mathsf{Attention}(W_q\boldsymbol{X}, W_k\boldsymbol{X}, W_v\boldsymbol{X}), \tag{4.21}$$

where W_q, W_k, W_v are matrices of learnable weights. The output of the self-attention operator is a new set of elements $\mathcal{S}(\boldsymbol{X})$, with the same cardinality as $\boldsymbol{X}$, in which each element of $\boldsymbol{X}$ is replaced with a weighted sum of the values, *i.e.* of linear projections of the input (Eq. 4.20).

Noticeably, attentive weights depend solely on the pairwise similarities between linear projections of the input set itself. Therefore, the self-attention operator can be seen as a way of encoding pairwise relationships inside the input set. When using image regions (or features derived from image regions) as the input set, $\mathcal{S}(\cdot)$ can naturally encode the pairwise relationships between regions that are needed to understand the input image before describing it[4].

This peculiarity in the definition of self-attention has, however, a significant limitation. Because everything depends solely on pairwise similarities, self-attention cannot model a priori knowledge on relationships between image regions. For example, given one region encoding a man and a region encoding a basketball ball, it would be difficult to infer the concept of *player* or *game* without any a priori knowledge. Again, given regions encoding eggs and toasts, the knowledge that the picture depicts a *breakfast* could be easily inferred using a priori knowledge on relationships.

Memory-augmented attention. To overcome this limitation of self-attention, we propose a memory-augmented attention operator. In our proposal, the set of

[4] Taking another perspective, self-attention is also conceptually equivalent to an attentive encoding of graph nodes [194].

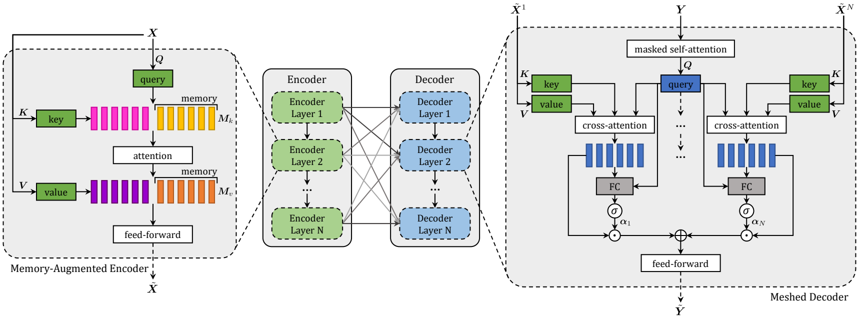


Figure 4.10: Architecture of the $\mathcal{M}^2$ Transformer. Our model is composed of a stack of memory-augmented encoding layers, which encodes multi-level visual relationships with a priori knowledge, and a stack of decoder layers, in charge of generating textual tokens. For the sake of clarity, AddNorm operations are not shown. Best seen in color.

keys and values used for self-attention is extended with additional "slots" which can encode a priori information. To stress that a priori information should not depend on the input set $\boldsymbol{X}$, the additional keys and values are implemented as plain learnable vectors which can be directly updated via SGD. Formally, the operator is defined as:

$$\begin{aligned} \mathcal{M}_{\text{mem}}(\boldsymbol{X}) &= \mathsf{Attention}(W_q\boldsymbol{X}, \boldsymbol{K}, \boldsymbol{V}) \\ \boldsymbol{K} &= [W_k\boldsymbol{X}, \boldsymbol{M}_k] \\ \boldsymbol{V} &= [W_v\boldsymbol{X}, \boldsymbol{M}_v], \end{aligned} \tag{4.22}$$

where $\boldsymbol{M}_k$ and $\boldsymbol{M}_v$ are learnable matrices with n_m rows, and $[\cdot,\cdot]$ indicates concatenation. Intuitively, by adding learnable keys and values, through attention it will be possible to retrieve learned knowledge which is not already embedded in $\boldsymbol{X}$. At the same time, our formulation leaves the set of queries unaltered. Intuitively again, this will help to avoid hallucination, given that knowledge is always retrieved because of similarities with queries which are seen in the image.

Just like the self-attention operator, our memory-augmented attention can be applied in a multi-head fashion. In this case, the memory-augmented attention operation is repeated h times, using different projection matrices W_q, W_k, W_v and different learnable memory slots $\boldsymbol{M}_k, \boldsymbol{M}_v$ for each head. Then, we concatenate the results from different heads and apply a linear projection.

Encoding layer. We embed our memory-augmented operator into a Transformer-like layer: the output of the memory-augmented attention is applied to a position-wise feed-forward layer composed of two affine transformations with a single non-linearity, which are independently applied to each element of the set. Formally,

$$\mathcal{F}(\boldsymbol{X})_i = U\sigma(V\boldsymbol{X}_i + b) + c, \tag{4.23}$$

where $\boldsymbol{X}_i$ indicates the i-th vector of the input set, and $\mathcal{F}(\boldsymbol{X})_i$ the i-th vector of the output. Also, $\sigma(\cdot)$ is the ReLU activation function, V and U are learnable weight matrices, b and c are bias terms.

Each of these sub-components (memory-augmented attention and position-wise feed-forward) is then encapsulated within a residual connection and a layer norm operation. The complete definition of an encoding layer can be finally written as:

$$\begin{aligned} \boldsymbol{Z} &= \mathsf{AddNorm}(\mathcal{M}_{\text{mem}}(\boldsymbol{X})) \\ \tilde{\boldsymbol{X}} &= \mathsf{AddNorm}(\mathcal{F}(\boldsymbol{Z})), \end{aligned} \tag{4.24}$$

where $\mathsf{AddNorm}$ indicates the composition of a residual connection and of a layer normalization.

Full encoder. Given the aforementioned structure, multiple encoding layers are stacked in sequence, so that the i-th layer consumes the output set computed by layer $i-1$. This amounts to creating multi-level encodings of the relationships between image regions, in which higher encoding layers can exploit and refine relationships already identified by previous layers, eventually using a priori knowledge. A stack of N encoding layers will therefore produce a multi-level output $\tilde{\mathcal{X}} = (\tilde{\boldsymbol{X}}^1, ..., \tilde{\boldsymbol{X}}^N)$, obtained from the outputs of each encoding layer.

Meshed decoder

Our decoder is conditioned on both previously generated words and region encodings, and is in charge of generating the next tokens of the output caption. Here, we exploit the aforementioned multi-level representation of the input image while still building a multi-layer structure. To this aim, we devise a meshed attention operator which, unlike the cross-attention operator of the Transformer, can take advantage of all encoding layers during the generation of the sentence.

Meshed cross-attention. Given an input sequence of vectors $\boldsymbol{Y}$, and outputs from all encoding layers $\tilde{\mathcal{X}}$, the Meshed Attention operator connects $\boldsymbol{Y}$ to all elements in $\tilde{\mathcal{X}}$ through gated cross-attentions. Instead of attending only the last encoding layer, we perform a cross-attention with all encoding layers. These multi-level contributions are then summed together after being modulated. Formally, our meshed attention operator is defined as

$$\mathcal{M}_{\text{mesh}}(\tilde{\mathcal{X}}, \boldsymbol{Y}) = \sum_{i=1}^{N} \boldsymbol{\alpha}_i \odot \mathcal{C}(\tilde{\boldsymbol{X}}^i, \boldsymbol{Y}), \tag{4.25}$$

where $\mathcal{C}(\cdot, \cdot)$ stands for the encoder-decoder cross-attention, computed using queries from the decoder and keys and values from the encoder:

$$\mathcal{C}(\tilde{\boldsymbol{X}}^i, \boldsymbol{Y}) = \mathsf{Attention}(W_q \boldsymbol{Y}, W_k \tilde{\boldsymbol{X}}^i, W_v \tilde{\boldsymbol{X}}^i), \tag{4.26}$$

and $\boldsymbol{\alpha}_i$ is a matrix of weights having the same size as the cross-attention results. Weights in $\boldsymbol{\alpha}_i$ modulate both the single contribution of each encoding layer, and the relative importance between different layers. These are computed by measuring the relevance between the result of the cross-attention computed with each encoding layer and the input query, as follows:

$$\boldsymbol{\alpha}_i = \sigma\left(W_i \left[\boldsymbol{Y}, \mathcal{C}(\tilde{\boldsymbol{X}}^i, \boldsymbol{Y})\right] + b_i\right), \tag{4.27}$$

where $[\cdot, \cdot]$ indicates concatenation, σ is the sigmoid activation, W_i is a $2d \times d$ weight matrix, and b_i is a learnable bias vector.

Architecture of decoding layers. As for encoding layers, we apply our meshed attention in a multi-head fashion. As the prediction of a word should only depend on previously predicted words, the decoder layer comprises a masked self-attention operation which connects queries derived from the t-th element of its input sequence $\boldsymbol{Y}$ with keys and values obtained from the left-hand subsequence, *i.e.* $\boldsymbol{Y}_{\leq t}$. Also, the decoder layer contains a position-wise feed-forward layer (as in Eq. 4.23), and all components are encapsulated within AddNorm operations. The final structure of the decoder layer can be written as:

$$\begin{aligned} \boldsymbol{Z} &= \mathsf{AddNorm}(\mathcal{M}_{\text{mesh}}(\mathsf{AddNorm}(\mathcal{S}_{\text{mask}}(\boldsymbol{Y}))) \\ \tilde{\boldsymbol{Y}} &= \mathsf{AddNorm}(\mathcal{F}(\boldsymbol{Z})), \end{aligned} \tag{4.28}$$

where $\boldsymbol{Y}$ is the input sequence of vectors and $\mathcal{S}_{\text{mask}}$ indicates a masked self-attention over time. Finally, our decoder stacks together multiple decoder layers, helping to refine both the understanding of the textual input and the generation of next tokens. Overall, the decoder takes as input word vectors, and the t-th element of its output sequence encodes the prediction of a word at time $t+1$, conditioned on $\boldsymbol{Y}_{\leq t}$. After taking a linear projection and a softmax operation, this encodes a probability over words in the dictionary.

4.2.2 Training

Following a standard practice in image captioning [152, 155, 4], we pre-train our model with a word-level cross-entropy loss (XE) and finetune the sequence generation using reinforcement learning. When training with XE, the model is trained to predict the next token given previous ground-truth words; in this case, the input sequence for the decoder is immediately available and the computation of the entire output sequence can be done in a single pass, parallelizing all operations over time.

When training with reinforcement learning, we employ a variant of the self-critical sequence training approach [155] on sequences sampled using beam search [4]: to decode, we sample the top-k words from the decoder probability distribution at each timestep, and always maintain the top-k sequences with highest probability. As sequence decoding is iterative in this step, the aforementioned parallelism over time cannot be exploited. However, intermediate keys and values used to compute the output token at time t can be reused in the next iterations.

Following previous works [4], we use the CIDEr-D score as reward, as it well correlates with human judgment [193]. We baseline the reward using the mean of the rewards rather than greedy decoding as done in previous methods [155, 4], as we found it to slightly improve the final performance. The final gradient expression for one sample is thus:

$$\nabla_\theta L(\theta) = -\frac{1}{k}\sum_{i=1}^{k}\big((r(\boldsymbol{w}^i) - b)\nabla_\theta \log p(\boldsymbol{w}^i)\big) \tag{4.29}$$

where $\boldsymbol{w}^i$ is the i-th sentence in the beam, $r(\cdot)$ is the reward function, and $b = \left(\sum_i r(\boldsymbol{w}^i)\right)/k$ is the baseline, computed as the mean of the rewards obtained by the sampled sequences. At prediction time, we decode again using beam search, and keep the sequence with highest predicted probability among those in the last beam.

4.2.3 Experimental settings

Datasets

We first evaluate our model on the COCO dataset [116], which is the most commonly used test-bed for image captioning. Then, we assess the captioning of novel objects by testing on the recently proposed nocaps dataset [1].

COCO. As previously mentioned, the dataset contains more than $120,000$ images, each of them annotated with 5 different captions. We follow the splits provided by Karpathy *et al.* [90], where $5,000$ images are used for validation, $5,000$ for testing and the rest for training. We also evaluate the model on the COCO online test server, composed of $40,775$ images for which annotations are not made publicly available.

nocaps. The dataset consists of $15,100$ images taken from the Open Images [106] validation and test sets, each annotated with 11 human-generated captions. Images are divided into validation and test splits, respectively composed of $4,500$ and $10,600$ elements. Images can be further grouped into three subsets depending on the nearness to COCO, namely in-domain, near-domain, and out-of-domain images. In-domain images contain only objects that are described in the COCO captions, out-of-domain images contain object classes that do not appear in COCO captions, and near-domain images contain both in-domain and out-of-domain object classes. Under this setting, we use COCO as training data and evaluate our results on the nocaps test server.

Metrics

Following the standard evaluation protocol, we employ the full set of captioning metrics: BLEU [144], METEOR [9], ROUGE [115], CIDEr [193], and SPICE [2]. A detailed description of these evaluation metrics can be found in Sec. 4.1.3.

Implementation details

To represent image regions, we use Faster R-CNN [154] with ResNet-101 [64] finetuned on the Visual Genome dataset [98, 4], thus obtaining a 2048-dimensional feature vector for each region. To represent words, we use one-hot vectors and linearly project them to the input dimensionality of the model d. We also employ sinusoidal positional encodings [192] to represent word positions inside the sequence and sum the two embeddings before the first decoding layer.

Pre-training with XE is done following the learning rate scheduling strategy of [192] with a warmup equal to $10,000$ iterations. Then, during CIDEr-D optimization, we use a fixed learning rate of 5×10^{-6}. We train all models using the Adam optimizer [93], a batch size of 50, and a beam size equal to 5.

Decoding optimization. As mentioned in Section 4.2.2, during the decoding stage computation cannot be parallelized over time as the input sequence is iteratively built. A naive approach would be to feed the model at each iteration with the previous $t-1$ generated words, $\{w_0, w_1, ..., w_{t-1}\}$ and sample the next predicted word w_t after computing the results of each attention and feed-forward layer over all timesteps. This in practice requires to re-compute the same queries, keys, values and attentive states multiple times, with intermediate results depending on w_t being recomputed $T-t$ times, where T is the length of the sampled sequence (in our experiments T is equal to 20).

In our implementation, we revert to a more computationally friendly approach in which we re-use intermediate results computed at previous timesteps. Each attentive layer of the decoder internally stores previously computed keys and values. At each timestep of the decoding, the model is fed only with w_{t-1}, and we only compute queries, keys and values depending on w_{t-1}.

In PyTorch, this can be implemented by exploiting the `register_buffer` method of `nn.Module`, and creating buffers to hold previously computed results. When running on a NVIDIA 2080Ti GPU, we found this to reduce training and inference times by approximately a factor of 3.

Vocabulary and tokenization. We convert all captions to lowercase, remove punctuation characters and tokenize using the spaCy NLP toolkit[5]. To build vocabularies, we remove all words which appear less than 5 times in training and validation splits. For each image, we use a maximum number of region feature vectors equal to 50.

Model dimensionality and weight initialization. In our model, we set the dimensionality d of each layer to 512, the number of heads to 8, and the number of memory vectors to 40. Using 8 attentive heads, the size of queries, keys and values in each head is set to $d/8 = 64$. We employ dropout with keep probability 0.9 after each attention and feed-forward layer. In our meshed attention operator (Eq. 4.25), we normalize the output with a scaling factor of $\sqrt{N}$.

Weights of attentive layers are initialized from the uniform distribution proposed by Glorot *et al.* [50], while weights of feed-forward layers are initialized using [63]. All biases are initialized to 0. Memory vectors for keys and values

[5] `https://spacy.io/`

are initialized from a normal distribution with zero mean and, respectively, $1/d_k$ and $1/m$ variance, where d_k is the dimensionality of keys and m is the number of memory vectors.

Novel object captioning. To train the model on the nocaps dataset, instead of using one-hot vectors, we represent words with GloVe word embeddings [147]. Two fully-connected layers are added to convert between the GloVe dimensionality and d before the first decoding layer and after the last decoding layer. Before the final softmax, we multiply with the transpose of the word embeddings.

Following [1], we use an object detector trained on Open Images [6] and filter detections by removing 39 Open Images classes that contain parts of objects or which are seldom mentioned. We also discard overlapping detections by removing the higher-order of two objects based on the class hierarchy, and we use the top-3 detected objects as constraints based on the detection confidence score. Differently from [1], we do not consider the plural forms or other word phrases of object classes, thus taking into account only the original class names. After decoding, we select the predicted caption with highest probability that satisfies the given constraints.

4.2.4 Ablation study

Performance of the Transformer. In previous works, the Transformer model has been applied to captioning only in its original configuration with six layers and self/cross attention, with the structure of connections that has been successful for uni-modal scenarios like machine translation. As we speculate that captioning requires specific architectures, we compare variations of the original Transformer with our approach.

Firstly, we investigate the impact of the number of encoding and decoding layers on captioning performance. As it can be seen in Table 4.5, the original Transformer (six layers) achieves 121.8 CIDEr, slightly superior to the Up-Down approach [4] which uses a two-layer recurrent language model with additive attention and includes a global feature vector (120.1 CIDEr). Varying the number of layers, we observe a significant increase in performance when using three encoding and three decoding layers, which leads to 123.6 CIDEr. We hypothesize that this is due to the reduced training set size and to the lower semantic complexities of sentences in captioning with respect to those of language understanding tasks. Following this finding, all subsequent experiments will use three layers.

[6] Specifically, the `tf_faster_rcnn_inception_resnet_v2_atrous_oidv2` model from the Tensorflow model zoo.

	B-1	B-4	M	R	C	S
Transformer (w/ 6 layers as in [192])	79.1	36.2	27.7	56.9	121.8	20.9
Transformer (w/ 3 layers)	79.6	36.5	27.8	57.0	123.6	21.1
Transformer (w/ AoA [74])	80.3	38.8	29.0	58.4	129.1	**22.7**
$\mathcal{M}^2$ Transformer$^{\text{1-to-1}}$ (w/o mem.)	80.5	38.2	28.9	58.2	128.4	22.2
$\mathcal{M}^2$ Transformer$^{\text{1-to-1}}$	80.3	38.2	28.9	58.2	129.2	22.5
$\mathcal{M}^2$ Transformer (w/o mem.)	80.4	38.3	29.0	58.2	129.4	22.6
$\mathcal{M}^2$ Transformer (w/ softmax)	80.3	38.4	29.1	58.3	130.3	22.5
$\mathcal{M}^2$ **Transformer**	**80.8**	**39.1**	**29.2**	**58.6**	**131.2**	22.6

Table 4.5: Ablation study and comparison with Transformer-based alternatives. All results are reported after the REINFORCE optimization stage.

Attention on Attention baseline. We also evaluate a recent proposal that can be straightforwardly applied to the Transformer as an alternative to standard dot-product attention. Specifically, we evaluate the addition of the "Attention on Attention" (AoA) approach [74] to the attentive layers, both in the encoder and in the decoder. Noticeably, in [74] this has been done with a Recurrent language model with attention, but the approach is sufficiently general to be applied to any attention stage. In this case, the result of dot-product attention is concatenated with the initial query and fed to two fully connected layers to obtain an information vector and a sigmoidal attention gate, then the two vectors are multiplied together. The final result is used as an alternative to the standard dot-product attention. This addition to a standard Transformer with three layers leads to 129.1 CIDEr (Table 4.5), thus underlying the usefulness of the approach also in Transformer-based models.

Meshed connectivity. We then evaluate the role of the meshed connections between encoder and decoder layers. In Table 4.5, we firstly introduce a reduced version of our approach in which the i-th decoder layer is only connected to the corresponding i-th encoder layer (1-to-1), instead of being connected to all encoders. As it can be noticed, using this 1-to-1 connectivity schema already brings an improvement with respect to using the output of the last encoder layer as in the standard Transformer (123.6 CIDEr vs 129.2 CIDEr), thus confirming that exploiting a multi-level encoding of image regions is beneficial. When we instead use our meshed connectivity schema, that exploits relationships encoded at all levels and weights them with a sigmoid gating, we observe a further performance improvement, from 129.2 CIDEr to 131.2 CIDEr. This amounts to a total

Memories	B-1	B-4	M	R	C	S
No memory	80.4	38.3	29.0	58.2	129.4	22.6
20	80.7	38.9	29.0	58.4	129.9	22.7
40	**80.8**	**39.1**	**29.2**	**58.6**	**131.2**	22.6
60	80.0	37.9	28.9	58.1	129.6	22.5
80	80.0	38.2	29.0	58.3	128.9	**22.9**

Table 4.6: Captioning results of $\mathcal{M}^2$ Transformer using different numbers of memory vectors.

Layers	B-1	B-4	M	R	C	S
2	80.5	38.6	29.0	58.4	128.5	**22.8**
3	**80.8**	**39.1**	**29.2**	**58.6**	**131.2**	22.6
4	**80.8**	38.6	29.1	58.5	129.6	22.6

Table 4.7: Captioning results of $\mathcal{M}^2$ Transformer using different numbers of encoder and decoder layers.

improvement of 7.6 CIDEr points with respect to the standard Transformer. Also, the result of our full model is superior to that obtained using the AoA.

As an alternative to the sigmoid gating approach for weighting the contributions from different encoder layers (Eq. 4.25), we also test with a softmax gating schema. In this case, the element-wise sigmoid applied to each encoder is replaced with the application of a softmax operation over the rows of $\boldsymbol{\alpha}_i$. Using this alternative brings to a reduction of around 1 CIDEr point, underlying that it is beneficial to exploit the full potentiality of a weighted sum of the contributions from all encoding layers, rather than forcing a peaky distribution in which one layer is given more importance than the others.

Role of persistent memory. We evaluate the role of memory vectors in both the 1-to-1 configuration and in the final configuration with meshed connections. As it can be seen from Table 4.5, removing memory vectors brings to a reduction in performance of around 1 CIDEr point in both connectivity settings, thus confirming the usefulness of exploiting a priori learned knowledge when encoding image regions.

In Table 4.6, we report the performance of our approach when using a varying number of memory vectors. As it can be seen, the best result in terms of BLEU, METEOR, ROUGE and CIDEr is obtained with 40 memory vectors, while 80 memory vectors provide a slightly superior result in terms of SPICE.

	B-1	B-4	M	R	C	S
SCST [155]	-	34.2	26.7	55.7	114.0	-
Up-Down [4]	79.8	36.3	27.7	56.9	120.1	21.4
RFNet [85]	79.1	36.5	27.7	57.3	121.9	21.2
Up-Down+HIP [227]	-	38.2	28.4	58.3	127.2	21.9
GCN-LSTM [226]	80.5	38.2	28.5	58.3	127.6	22.0
SGAE [223]	**80.8**	38.4	28.4	58.6	127.8	22.1
ORT [67]	80.5	38.6	28.7	58.4	128.3	**22.6**
AoANet [74]	80.2	38.9	**29.2**	**58.8**	129.8	22.4
$\mathcal{M}^2$ **Transformer**	**80.8**	**39.1**	**29.2**	58.6	**131.2**	**22.6**

Table 4.8: Comparison with the state of the art on the “Karpathy” test split, in single-model setting.

Encoder and decoder layers. We also investigate the performance of the $\mathcal{M}^2$ Transformer when changing the number of encoding and decoding layers. Table 4.7 shows that the best performance is obtained with three encoding and decoding layers, thus confirming the initial findings on the base Transformer model. As our model can deal with a different number of encoding and decoding layers, we also experimented with non symmetric encoding-decoding architectures, without however noticing significant improvements in performance.

4.2.5 Comparison with state of the art

We compare the performances of our approach with those of several recent proposals for image captioning. The models we compare to include SCST [155], which uses attention over the grid of features and a one-layer LSTM language model; Up-Down [4], which introduces attention over regions, and uses a two-layer LSTM language model. Also, we compare to the RFNet approach [85], which uses a recurrent fusion network to merge different CNN features; GCN-LSTM [226], which exploits pairwise relationships between image regions through a Graph Convolutional Neural Network; SGAE [223], which instead uses auto-encoding scene graphs. Further, we compare with the original AoANet [74] approach, which uses attention on attention for encoding image regions and an LSTM language model. Finally, we compare with ORT [67], which uses a plain Transformer, and weights attention scores in the region encoder with pairwise distances between detections.

We evaluate our approach on the COCO “Karpathy” test split, using both

	B-1	B-4	M	R	C	S
Ensemble/Fusion of 2 models						
GCN-LSTM [226]	80.9	38.3	28.6	58.5	128.7	22.1
SGAE [223]	81.0	39.0	28.4	58.9	129.1	22.2
ETA [111]	81.5	**39.9**	28.9	59.0	127.6	22.6
GCN-LSTM+HIP [227]	-	39.1	28.9	**59.2**	130.6	22.3
$\mathcal{M}^2$ **Transformer**	**81.6**	39.8	**29.5**	**59.2**	**133.2**	**23.1**
Ensemble/Fusion of 4 models						
SCST [155]	-	35.4	27.1	56.6	117.5	-
RFNet [85]	80.4	37.9	28.3	58.3	125.7	21.7
AoANet [74]	81.6	40.2	29.3	59.4	132.0	22.8
$\mathcal{M}^2$ **Transformer**	**82.0**	**40.5**	**29.7**	**59.5**	**134.5**	**23.5**

Table 4.9: Comparison with the state of the art on the “Karpathy” test split, using an ensemble of models.

single model and ensemble configurations, and on the online COCO evaluation server.

Single model. In Table 4.8 we report the performance of our method in comparison with the aforementioned competitors, using captions predicted from a single model and optimization on the CIDEr-D score. As it can be observed, our method surpasses all other approaches in terms of BLEU-4, METEOR and CIDEr, while being competitive on BLEU-1 and SPICE with the best performer, and slightly worse on ROUGE with respect to AoANet [74]. In particular, it advances the current state of the art on CIDEr by 1.4 points.

Ensemble model. Following the common practice [155, 74] of building an ensemble of models, we also report the performances of our approach when averaging the output probability distributions of multiple and independently trained instances of our model. In Table 4.9, we use ensembles of two and four models, trained from different random seeds. Noticeably, when using four models our approach achieves the best performance according to all metrics, with an increase of 2.5 CIDEr points with respect to the current state of the art [74].

Online Evaluation. Finally, we also report the performance of our method on the online COCO test server[7]. In this case, we use the ensemble of four models previously described, trained on the “Karpathy” training split. The evaluation is done

[7] https://competitions.codalab.org/competitions/3221

on the COCO test split, for which ground-truth annotations are not publicly available. Results are reported in Table 4.10, in comparison with the top-performing approaches of the leaderboard. For fairness of comparison, they also used an ensemble configuration. As it can be seen, our method surpasses the current state of the art on all metrics, achieving an advancement of 1.4 CIDEr points with respect to the best performer.

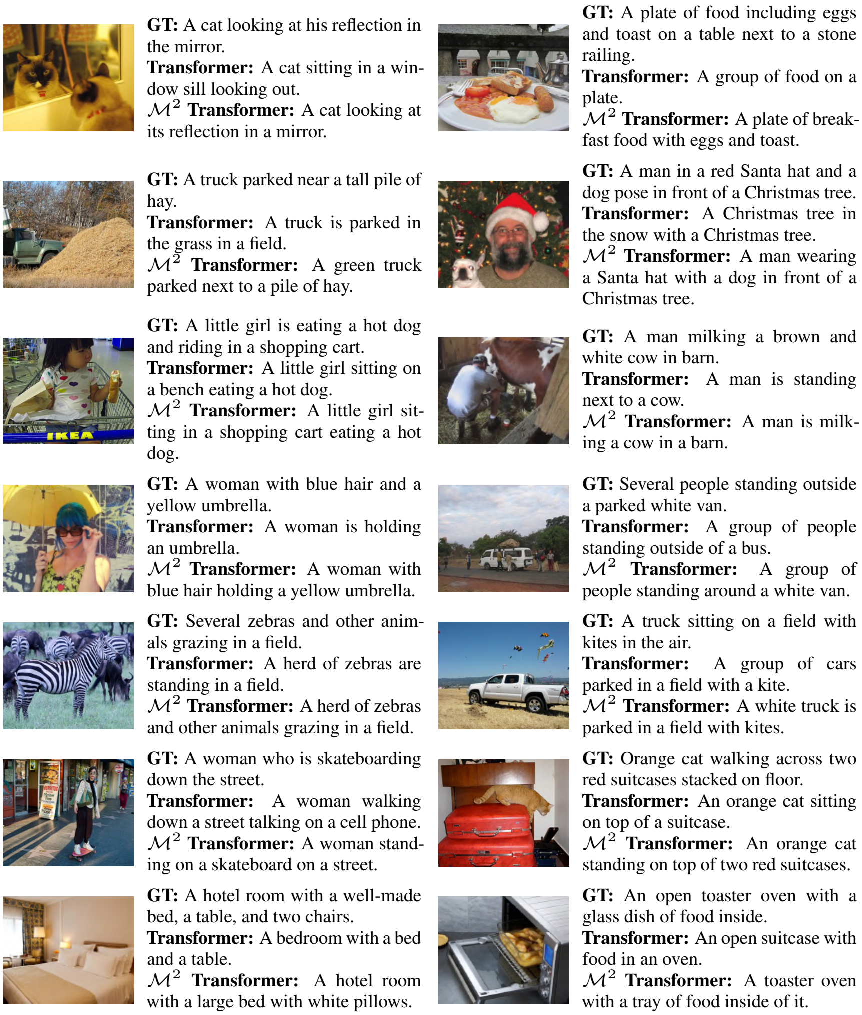
4.2.6 Qualitative results and visualization

Figure 4.11 proposes qualitative results generated by our model and the original Transformer. On average, our model is able to generate more accurate and descriptive captions, integrating fine-grained details and object relations.

Finally, to better understand the effectiveness of our $\mathcal{M}^2$ Transformer, we investigate the contribution of detected regions to the model output. Differently from recurrent-based captioning models, in which attention weights over regions can be easily extracted, in our model the contribution of one region with respect to the output is given by more complex non-linear dependencies. Therefore, we revert to attribution methods and we employ the Integrated Gradients approach [177], which approximates the integral of gradients with respect to the given input. Specifically, the Integrated Gradients approach produces an attribution score for each feature channel of each input region. To obtain the attribution of each region, we average over the feature channels, and re-normalize the obtained scores by their sum. For visualization purposes, we apply a contrast stretching function to project scores in the 0-1 interval. Results are presented in Figure 4.12, where we observe that our approach correctly grounds image regions to words, also in presence of object details and small detections.

	BLEU-1		BLEU-2		BLEU-3		BLEU-4		METEOR		ROUGE		CIDEr	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
SCST [155]	78.1	93.7	61.9	86.0	47.0	75.9	35.2	64.5	27.0	35.5	56.3	70.7	114.7	116.7
Up-Down [4]	80.2	95.2	64.1	88.8	49.1	79.4	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
RDN [92]	80.2	95.3	-	-	-	-	37.3	69.5	28.1	37.8	57.4	73.3	121.2	125.2
RFNet [85]	80.4	95.0	64.9	89.3	50.1	80.1	38.0	69.2	28.2	37.2	58.2	73.1	122.9	125.1
GCN-LSTM [226]	80.8	95.9	65.5	89.3	50.8	80.3	38.7	69.7	28.5	37.6	58.5	73.4	125.3	126.5
SGAE [223]	81.0	95.3	65.6	89.5	50.7	80.4	38.5	69.7	28.2	37.2	58.6	73.6	123.8	126.5
ETA [111]	81.2	95.0	65.5	89.0	50.9	80.4	38.9	70.2	28.6	38.0	58.6	73.9	122.1	124.4
AoANet [74]	81.0	95.0	65.8	89.6	51.4	81.3	39.4	71.2	29.1	38.5	58.9	74.5	126.9	129.6
GCN-LSTM+HIP [227]	**81.6**	95.9	66.2	90.4	51.5	81.6	39.3	71.0	28.8	38.1	59.0	74.1	127.9	130.2
$\mathcal{M}^2$ **Transformer**	**81.6**	**96.0**	**66.4**	**90.8**	**51.8**	**82.7**	**39.7**	**72.8**	**29.4**	**39.0**	**59.2**	**74.8**	**129.3**	**132.1**

Table 4.10: Leaderboard of various methods on the online MS-COCO test server.



GT: A cat looking at his reflection in the mirror.
Transformer: A cat sitting in a window sill looking out.
$\mathcal{M}^2$ **Transformer:** A cat looking at its reflection in a mirror.

GT: A plate of food including eggs and toast on a table next to a stone railing.
Transformer: A group of food on a plate.
$\mathcal{M}^2$ **Transformer:** A plate of breakfast food with eggs and toast.

GT: A truck parked near a tall pile of hay.
Transformer: A truck is parked in the grass in a field.
$\mathcal{M}^2$ **Transformer:** A green truck parked next to a pile of hay.

GT: A man in a red Santa hat and a dog pose in front of a Christmas tree.
Transformer: A Christmas tree in the snow with a Christmas tree.
$\mathcal{M}^2$ **Transformer:** A man wearing a Santa hat with a dog in front of a Christmas tree.

GT: A little girl is eating a hot dog and riding in a shopping cart.
Transformer: A little girl sitting on a bench eating a hot dog.
$\mathcal{M}^2$ **Transformer:** A little girl sitting in a shopping cart eating a hot dog.

GT: A man milking a brown and white cow in barn.
Transformer: A man is standing next to a cow.
$\mathcal{M}^2$ **Transformer:** A man is milking a cow in a barn.

GT: A woman with blue hair and a yellow umbrella.
Transformer: A woman is holding an umbrella.
$\mathcal{M}^2$ **Transformer:** A woman with blue hair holding a yellow umbrella.

GT: Several people standing outside a parked white van.
Transformer: A group of people standing outside of a bus.
$\mathcal{M}^2$ **Transformer:** A group of people standing around a white van.

GT: Several zebras and other animals grazing in a field.
Transformer: A herd of zebras are standing in a field.
$\mathcal{M}^2$ **Transformer:** A herd of zebras and other animals grazing in a field.

GT: A truck sitting on a field with kites in the air.
Transformer: A group of cars parked in a field with a kite.
$\mathcal{M}^2$ **Transformer:** A white truck is parked in a field with kites.

GT: A woman who is skateboarding down the street.
Transformer: A woman walking down a street talking on a cell phone.
$\mathcal{M}^2$ **Transformer:** A woman standing on a skateboard on a street.

GT: Orange cat walking across two red suitcases stacked on floor.
Transformer: An orange cat sitting on top of a suitcase.
$\mathcal{M}^2$ **Transformer:** An orange cat standing on top of two red suitcases.

GT: A hotel room with a well-made bed, a table, and two chairs.
Transformer: A bedroom with a bed and a table.
$\mathcal{M}^2$ **Transformer:** A hotel room with a large bed with white pillows.

GT: An open toaster oven with a glass dish of food inside.
Transformer: An open suitcase with food in an oven.
$\mathcal{M}^2$ **Transformer:** A toaster oven with a tray of food inside of it.

Figure 4.11: Examples of captions generated by our approach and the original Transformer model, as well as the corresponding ground-truths.

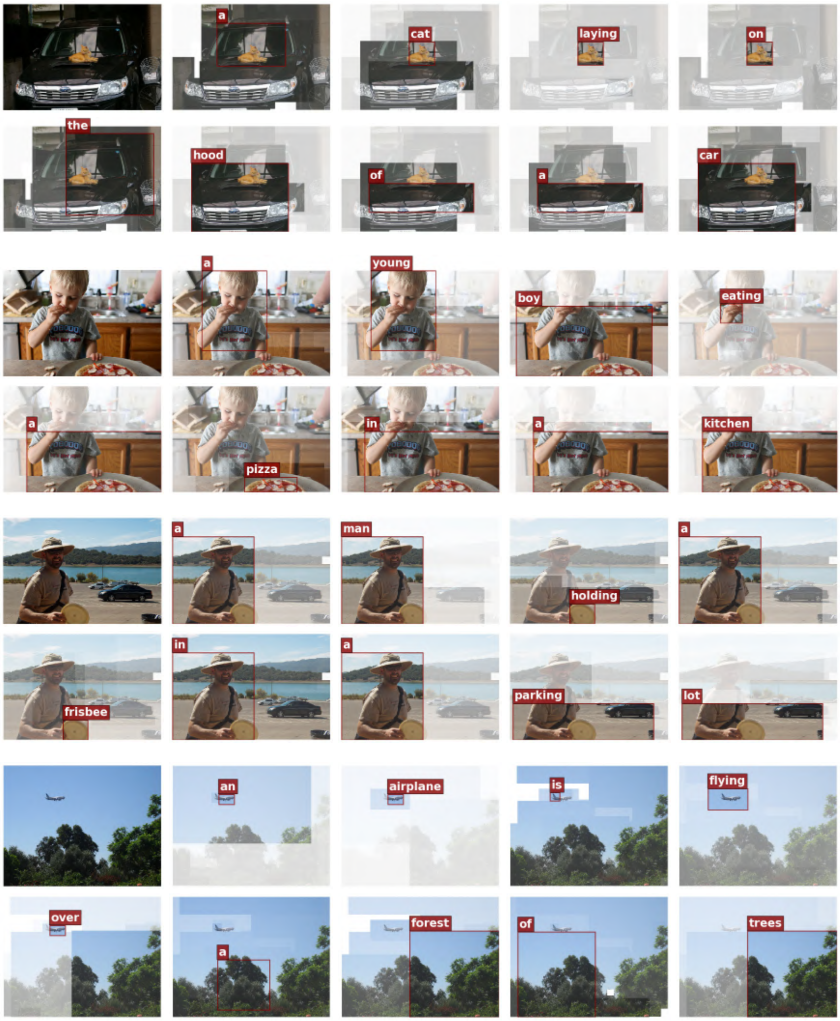


Figure 4.12: Visualization of attention states for four sample captions. For each generated word, we show the attended image regions, outlining the region with the maximum output attribution in red.

	In-Domain		Out-of-Domain		Overall	
	CIDEr	SPICE	CIDEr	SPICE	CIDEr	SPICE
NBT + CBS [1]	62.1	10.1	62.4	8.9	60.2	9.5
Up-Down + CBS [1]	80.0	12.0	66.4	9.7	73.1	11.1
Transformer	78.0	11.0	29.7	7.8	54.7	9.8
$\mathcal{M}^2$ Transformer	**85.7**	**12.1**	38.9	8.9	64.5	11.1
Transformer + CBS	74.3	11.0	62.5	9.2	66.9	10.3
$\mathcal{M}^2$ Transformer + CBS	81.2	12.0	**69.4**	**10.0**	**75.0**	**11.4**

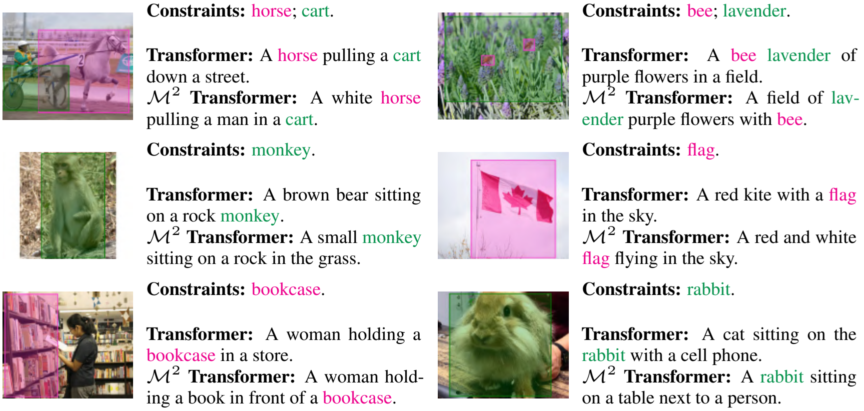
Table 4.11: Performances on nocaps validation set, for in-domain and out-of-domain captioning.

4.2.7 Describing novel objects

We also assess the performance of our approach when dealing with images containing object categories that are not seen in the training set. We compare with the Up-Down model [4] and Neural Baby Talk [124], when using GloVe word embeddings and Constrained Beam Search (CBS) [3] to address the generation of out-of-vocabulary words and constrain the presence of categories detected by an object detector. To compare with our model, we use a simplified implementation of the procedure described in [1] to extract constraints, without using word phrases (*e.g.* plurals).

Results are shown in Table 4.11: as it can be seen, the original Transformer is significantly less performing than Up-Down on both in-domain and out-of-domain categories, while our approach can properly deal with novel categories, surpassing the Up-Down baseline in both in-domain and out-of-domain images. As expected, the use of CBS significantly enhances the performances, in particular on out-of-domain captioning.

Figure 4.13 reports sample captions produced by our approach on images from the nocaps dataset. On each image, we compare to the baseline Transformer and show the constraints provided by the object detector. Overall, the $\mathcal{M}^2$ Transformer is able to better incorporate the constraints while maintaining the fluency and properness of the generated sentences.



Constraints: horse; cart.

Transformer: A horse pulling a cart down a street.
$\mathcal{M}^2$ **Transformer:** A white horse pulling a man in a cart.

Constraints: bee; lavender.

Transformer: A bee lavender of purple flowers in a field.
$\mathcal{M}^2$ **Transformer:** A field of lavender purple flowers with bee.

Constraints: monkey.

Transformer: A brown bear sitting on a rock monkey.
$\mathcal{M}^2$ **Transformer:** A small monkey sitting on a rock in the grass.

Constraints: flag.

Transformer: A red kite with a flag in the sky.
$\mathcal{M}^2$ **Transformer:** A red and white flag flying in the sky.

Constraints: bookcase.

Transformer: A woman holding a bookcase in a store.
$\mathcal{M}^2$ **Transformer:** A woman holding a book in front of a bookcase.

Constraints: rabbit.

Transformer: A cat sitting on the rabbit with a cell phone.
$\mathcal{M}^2$ **Transformer:** A rabbit sitting on a table next to a person.

Figure 4.13: Sample nocaps images and corresponding predicted captions generated by our model and the original Transformer. For each image, we report the Open Images object classes predicted by the object detector and used as constraints during the generation of the caption.

Chapter 5

Controllable captioning

In the previous chapter, we have shown how it is possible to automatically describe an image in natural language using both recurrent neural networks and fully-attentive models. However, the behavior of these architectures is not always easy to be explained from the exterior thus limiting the applicability of captioning algorithms in complex scenarios. As an image can be described in infinite ways depending on the goal and the context at hand, an higher degree of controllability over the generation process may be desired.

Contributions

Following these premises, in the first part of this chapter (Sec. 5.1), we introduce a novel framework for image captioning which can generate diverse descriptions by allowing both grounding and controllability. Given a control signal in the form of a sequence or set of image regions, we generate the corresponding caption through a recurrent architecture which predicts textual chunks explicitly grounded on regions, following the constraints of the given control. Experiments are conducted on Flickr30k Entities and on COCO Entities, an extended version of COCO in which we add grounding annotations collected in a semi-automatic manner. Results demonstrate that our method achieves state-of-the-art performances on controllable image captioning, in terms of caption quality and diversity. The

This chapter is related to publications [3, 17, 18] reported in Appendix B, by the author of the thesis. See Appendix B for details.

source code of our model and the additional annotations of the COCO dataset are publicly available[1].

While this approach focuses on captioning architectures controllable through a sequence or a set of regions, another lever of controllability that may be required is that of identifying people present in the scene and naming them with their proper names. This can be achieved in the context of video captioning in which large scale movie description datasets are available with the annotations of character identities. However, current approaches for movie description lack the ability to name characters with their proper names, and can only indicate people with a generic "someone" tag. In the second part of this chapter (Sec. 5.2), we present two contributions towards the development of video description architectures with naming capabilities: firstly, we collect and release[2] an extension of the popular Montreal Video Annotation Dataset in which the visual appearance of each character is linked both through time and to textual mentions in captions. We annotate, in a semi-automatic manner, a total of 63k face tracks and 34k textual mentions on 92 movies. Moreover, to showcase the features of the dataset and quantify the complexity of the naming task, we investigate multimodal architectures to replace the "someone" tags with proper character names in existing video captions.

5.1 Controlling caption generation via detection selection

Despite recent advancements, standard captioning models still lack controllability and explainability – *i.e.*, their behavior can hardly be influenced and explained. As an example, in the case of attention-driven models, the architecture implicitly selects which regions to focus on at each timestep, but it cannot be supervised from the exterior. While an image can be described in multiple ways, such an architecture provides no way of controlling which regions are described and what importance is given to each region. This lack of controllability creates a distance between human and machine intelligence, as humans can manage the variety of ways in which an image can be described, and select the most appropriate one depending on the task and the context at hand. Most importantly, this also limits the applicability of captioning algorithms to complex scenarios in which some control over the generation process is needed. As an example, a captioning-based

[1] https://github.com/aimagelab/show-control-and-tell

[2] https://github.com/aimagelab/mvad-names-dataset

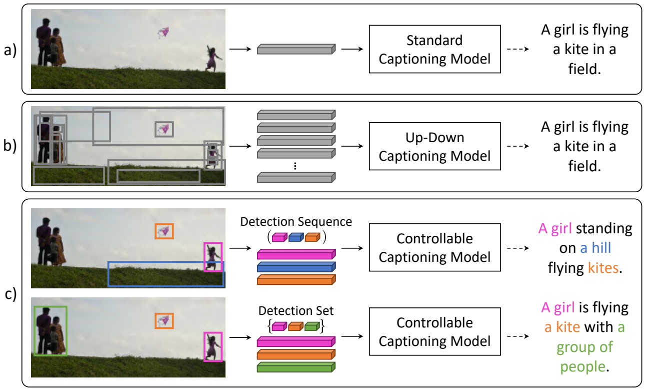


Figure 5.1: Comparison between (a) captioning models with global visual feature [202], (b) attentive models which integrate features from image regions [4] and (c) our *Show, Control and Tell*. Our method can produce multiple captions for a given image, depending on a control signal which can be either a sequence or a set of image regions. Moreover, chunks of the generated sentences are explicitly grounded on regions.

driver assistance system would need to focus on dangerous objects on the road to alert the driver, rather than describing the presence of trees and cars when a risky situation is detected. Eventually, such systems would also need to be explainable, so that their behavior could be easily interpreted in case of failures.

In this section, we introduce *Show, Control and Tell*, that explicitly addresses these shortcomings (Fig. 5.1). It can generate diverse natural language captions depending on a control signal which can be given either as a sequence or as a set of image regions which need to be described. As such, our method is capable of describing the same image by focusing on different regions and in a different order, following the given conditioning. Our model is built on a recurrent architecture which considers the decomposition of a sentence into noun chunks and models the relationship between image regions and textual chunks, so that the generation process can be explicitly grounded on image regions. To the best of our knowledge, this is the first captioning framework controllable from image regions. We evaluate our solution with respect to a set of carefully designed baselines, on Flickr30k Entities and on COCO, which we semi-automatically augment with grounding image regions for training and evaluation purposes. We demonstrate

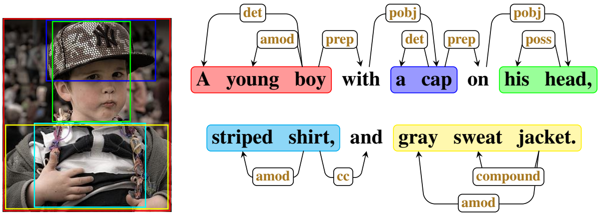


Figure 5.2: Example of a dependency tree for a caption. Noun chunks are marked with rounded boxes; chunks corresponding to image regions are depicted using the same color.

that our proposed method achieves state-of-the-art results for controllable image captioning on Flick30k and COCO both in terms of diversity and caption quality, even when compared with methods which focus on diversity.

5.1.1 Preliminaries

Sentences are natural language structures which are hierarchical by nature [128]. At the lowest level, a sentence might be thought as a sequence of words: in the case of a sentence describing an image, we can further distinguish between *visual* words, which describe something visually present in the image, and *textual* words, that refer to entities which are not present in the image [124]. Analyzing further the syntactic dependencies between words, we can recover a higher abstraction level in which words can be organized into a tree-like structure: in a dependency tree [52, 71, 25], each word is linked together with its modifiers (Fig. 5.2).

Given a dependency tree, nouns can be grouped with their modifiers, thus building *noun chunks*. For instance, the caption depicted in Fig. 5.2 can be decomposed into a sequence of different noun chunks: “a young boy”, “a cap”, “his head”, “striped shirt”, and “gray and sweat jacket”. As noun chunks, just like words, can be visually grounded into image regions, a caption can also be mapped to a sequence of regions, each corresponding to a noun chunk. A chunk might also be associated with multiple image regions of the same class if more than one possible mapping exists.

The number of ways in which an image can be described results in different sequences of chunks, linked together to form a fluent sentence. Therefore, captions also differ in terms of the set of considered regions, the order in which they are described, and their mapping to chunks given by the linguistic abilities of the annotator.

Following these premises, we define a model which can recover the variety of ways in which an image can be described, given a control input expressed as a sequence or set of image regions. We begin by presenting the former case, and then show how our model deals with the latter scenario.

5.1.2 Generating controllable captions

Given an image $\boldsymbol{I}$ and an ordered sequence of set of regions $\boldsymbol{R} = (\boldsymbol{r}_0, \boldsymbol{r}_1, ..., \boldsymbol{r}_N)$[3], the goal of our captioning model is to generate a sentence $\boldsymbol{y} = (y_0, y_1, ..., y_T)$ which in turns describes all the regions in $\boldsymbol{R}$ while maintaining the fluency of language.

Our model is conditioned on both the input image $\boldsymbol{I}$ and the sequence of region sets $\boldsymbol{R}$, which acts as a control signal, and jointly predicts two output distributions which correspond to the word-level and chunk-level representation of the sentence: the probability of generating a word at a given time, *i.e.* $p(y_t|\boldsymbol{R}, \boldsymbol{I}; \boldsymbol{\theta})$, and that of switching from one chunk to another, *i.e.* $p(g_t|\boldsymbol{R}, \boldsymbol{I}; \boldsymbol{\theta})$, where g_t is a boolean chunk-shifting gate. During the generation, the model maintains a pointer to the current region set $\boldsymbol{r}_i$ and can shift to the next element in $\boldsymbol{R}$ by means of the gate g_t.

To generate the output caption, we employ a recurrent neural network with adaptive attention. At each timestep, we compute the hidden state $\boldsymbol{h}_t$ according to the previous hidden state $\boldsymbol{h}_{t-1}$, the current image region set $\boldsymbol{r}_t$ and the current word w_t, such that $\boldsymbol{h}_t = \text{RNN}(w_t, \boldsymbol{r}_t, \boldsymbol{h}_{t-1})$. At training time, $\boldsymbol{r}_t$ and w_t are the ground-truth region set and word corresponding to timestep t; at test time, w_t is sampled from the first distribution predicted by the model, while the choice of the next image region is driven by the values of the chunk-shifting gate sampled from the second distribution:

$$\boldsymbol{r}_{t+1} \leftarrow \boldsymbol{R}[i], \quad \text{where } i = \min\left(\sum_{k=1}^{t} g_k, N\right), \; g_k \in \{0, 1\} \tag{5.1}$$

where $\{g_k\}_k$ is the sequence of sampled gate values, and N is the number of region sets in $\boldsymbol{R}$.

[3] For generality, we will always consider sequences of sets of regions, to deal with the case in which a chunk in the target sentence can be associated to multiple regions in training and evaluation data.

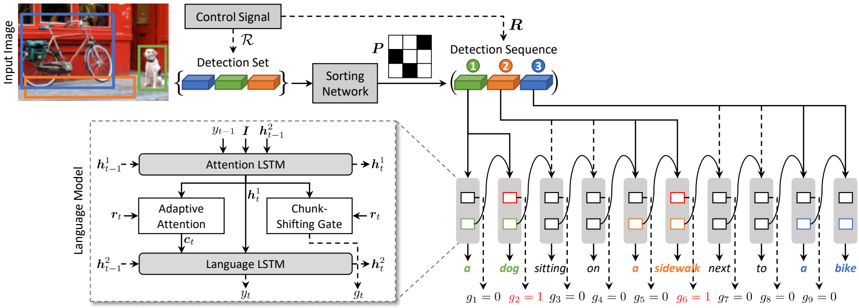


Figure 5.3: Overview of the approach. Given an image and a control signal, the figure shows the process to generate the controlled caption and the architecture of the language model.

Chunk-shifting gate. We compute $p(g_t|\boldsymbol{R})$ via an adaptive mechanism in which the LSTM computes a compatibility function between its internal state and a latent representation which models the state of the memory at the end of a chunk. The compatibility score is compared to that of attending one of the regions in $\boldsymbol{r}_t$, and the result is used as an indicator to switch to the next region set in $\boldsymbol{R}$.

The LSTM is firstly extended to obtain a chunk sentinel $\boldsymbol{s}^c_t$, which models a component extracted from the memory encoding the state of the LSTM at the end of a chunk. The sentinel is computed as:

$$\boldsymbol{l}^c_t = \sigma(\boldsymbol{W}_{ig}\boldsymbol{x}_t + \boldsymbol{W}_{hg}\boldsymbol{h}_{t-1}) \tag{5.2}$$

$$\boldsymbol{s}^c_t = \boldsymbol{l}^c_t \odot \tanh(\boldsymbol{m}_t) \tag{5.3}$$

where $\boldsymbol{W}_{ig} \in \mathbb{R}^{d\times k}$, $\boldsymbol{W}_{hg} \in \mathbb{R}^{d\times d}$ are learnable weights, $\boldsymbol{m}_t \in \mathbb{R}^d$ is the LSTM cell memory and $\boldsymbol{x}_t \in \mathbb{R}^k$ is the input of the LSTM at time t; $\odot$ represents the Hadamard element-wise product and σ the sigmoid logistic function.

We then compute a compatibility score between the internal state $\boldsymbol{h}_t$ and the sentinel vector through a single-layer neural network; analogously, we compute a compatibility function between $\boldsymbol{h}_t$ and the regions in $\boldsymbol{r}_t$.

$$z^c_t = \boldsymbol{w}^T_h \tanh(\boldsymbol{W}_{sg}\boldsymbol{s}^c_t + \boldsymbol{W}_g\boldsymbol{h}_t) \tag{5.4}$$

$$\boldsymbol{z}^r_t = \boldsymbol{w}^T_h \tanh(\boldsymbol{W}_{sr}\boldsymbol{r}_t + (\boldsymbol{W}_g\boldsymbol{h}_t)\mathbb{1}^T) \tag{5.5}$$

where n is the number of regions in $\boldsymbol{r}_t$, $\mathbb{1} \in \mathbb{R}^n$ is a vector with all elements set to 1, $\boldsymbol{w}_h^T$ is a row vector, and all $\boldsymbol{W}_*$, $\boldsymbol{w}_*$ are learnable parameters. Notice that the representation extracted from the internal state is shared between all compatibility scores, as if the region set and the sentinel vector were part of the same attentive distribution. Contrarily to an attentive mechanism, however, there is no value extraction.

The probability of shifting from one chunk to the next one is defined as the probability of attending the sentinel vector $\boldsymbol{s}_t^c$ in a distribution over $\boldsymbol{s}_t^c$ and the regions in $\boldsymbol{r}_t$:

$$p(g_t = 1|\boldsymbol{R}) = \frac{\exp z_t^c}{\exp z_t^c + \sum_{i=1}^{n} \exp \boldsymbol{z}_{ti}^r} \tag{5.6}$$

where $\boldsymbol{z}_{ti}^r$ indicates the i-th element in $\boldsymbol{z}_t^r$, and we dropped the dependency between n and t for clarity. At test time, the value of gate $g_t \in \{0, 1\}$ is then sampled from $p(g_t|\boldsymbol{R})$ and drives the shifting to the next region set in $\boldsymbol{R}$.

Adaptive attention with visual sentinel. While the chunk-shifting gate predicts the end of a chunk, thus linking the generation process with the control signal given by $\boldsymbol{R}$, once $\boldsymbol{r}_t$ has been selected a second mechanism is needed to attend its regions and distinguish between visual and textual words. To this end, we build an adaptive attention mechanism with a visual sentinel [123].

The visual sentinel vector models a component of the memory to which the model can fall back when it chooses to not attend a region in $\boldsymbol{r}_t$. Analogously to Eq. 5.2, it is defined as:

$$\boldsymbol{l}_t^v = \sigma(\boldsymbol{W}_{is}\boldsymbol{x}_t + \boldsymbol{W}_{hs}\boldsymbol{h}_{t-1}) \tag{5.7}$$

$$\boldsymbol{s}_t^v = \boldsymbol{l}_t^v \odot \tanh(\boldsymbol{m}_t) \tag{5.8}$$

where $\boldsymbol{W}_{is} \in \mathbb{R}^{d \times k}$ and $\boldsymbol{W}_{hs} \in \mathbb{R}^{d \times d}$ are matrices of learnable weights. An attentive distribution is then generated over the regions in $\boldsymbol{r}_t$ and the visual sentinel vector $\boldsymbol{s}_t^v$:

$$\boldsymbol{\alpha}_t = \mathrm{softmax}([\boldsymbol{z}_t^r; \boldsymbol{w}_h^T \tanh(\boldsymbol{W}_{ss}\boldsymbol{s}_t^v + \boldsymbol{W}_g\boldsymbol{h}_t)]) \tag{5.9}$$

where $[\cdot]$ indicates concatenation. Based on the attention distribution, we obtain a context vector which can be fed to the LSTM as a representation of what the network is attending:

$$\boldsymbol{c}_t = \sum_{i=1}^{n+1} \boldsymbol{\alpha}_{ti}[\boldsymbol{r}_t; \boldsymbol{s}_t^v] \tag{5.10}$$

Notice that the context vector will be, mostly, an approximation of one of the regions in $\boldsymbol{r}_t$ or the visual sentinel. However, $\boldsymbol{r}_t$ will vary at different timestep according to the chunk-shifting mechanism, thus following the control input. The model can alternate the generation of visual and textual words by means of the visual sentinel.

5.1.3 Objective

The captioning model is trained using a loss function which considers the two output distributions of the model. Given the target ground-truth caption $\boldsymbol{y}^*_{1:T}$, the ground-truth region sets $\boldsymbol{r}^*_{1:T}$ and chunk-shifting gate values corresponding to each timestep $g^*_{1:T}$, we train both distributions by means of a cross-entropy loss. The relationship between target region sets and gate values will be further expanded in the implementation details. The loss function for a sample is defined as:

$$\begin{aligned} L(\theta) = -\sum_{t=1}^{T} \Big(&\log \overbrace{p(y^*_t | \boldsymbol{r}^*_{1:t}, \boldsymbol{y}^*_{1:t-1})}^{\text{Word-level probability}} + \\ &+ g^*_t \log p(g_t = 1 | \boldsymbol{r}^*_{1:t}, \boldsymbol{y}^*_{1:t-1}) + \\ &+ (1 - g^*_t) \log(1 - \underbrace{p(g_t = 1 | \boldsymbol{r}^*_{1:t}, \boldsymbol{y}^*_{1:t-1})}_{\text{Chunk-level probability}}) \Big) \end{aligned} \tag{5.11}$$

Following previous works [152, 155, 4], after a pre-training step using cross-entropy, we further optimize the sequence generation using Reinforcement Learning. Specifically, we use the self-critical sequence training approach [155], which baselines the REINFORCE algorithm with the reward obtained under the inference model at test time.

Given the nature of our model, we extend the approach to work on multiple output distributions. At each timestep, we sample from both $p(y_t|\boldsymbol{R})$ and $p(g_t|\boldsymbol{R})$ to obtain the next word w_{t+1} and region set $\boldsymbol{r}_{t+1}$. Once a EOS tag is reached, we compute the reward of the sampled sentence $\boldsymbol{w}^s$ and backpropagate with respect to both the sampled word sequence $\boldsymbol{w}^s$ and the sequence of chunk-shifting gates $\boldsymbol{g}^s$. The final gradient expression is thus:

$$\nabla_\theta L(\theta) = -(r(\boldsymbol{w}^s) - b)(\nabla_\theta \log p(\boldsymbol{w}^s) + \nabla_\theta \log p(\boldsymbol{g}^s)) \tag{5.12}$$

where $b = r(\hat{\boldsymbol{w}})$ is the reward of the sentence obtained using the inference procedure (*i.e.* by sampling the word and gate value with maximum probability). We then build a reward function which jointly considers the quality of the caption and its alignment with the control signal $\boldsymbol{R}$.

Rewarding caption quality. To reward the overall quality of the generated caption, we use image captioning metrics as a reward. Following previous works [4], we employ the CIDEr metric (specifically, the CIDEr-D score) which has been shown to correlate better with human judgment [193].

Rewarding the alignment. While captioning metrics can reward the semantic quality of the sentence, none of them can evaluate the alignment with respect to the control input[4]. Therefore, we introduce an alignment score based on the Needleman-Wunsch algorithm [139].

Given a predicted caption $\boldsymbol{y}$ and its target counterpart $\boldsymbol{y}^*$, we extract all nouns from both sentences, and evaluate the alignment between them, recalling the relationships between noun chunks and region sets. We use the following scoring system: the reward for matching two nouns is equal to the cosine similarity between their word embeddings; a gap gets a negative reward equal to the minimum similarity value, *i.e.* -1. Once the optimal alignment is computed, we normalize its score, $al(\boldsymbol{y}, \boldsymbol{y}^*)$ with respect to the length of the sequences. The alignment score is thus defined as:

$$\mathrm{NW}(\boldsymbol{y}, \boldsymbol{y}^*) = \frac{al(\boldsymbol{y}, \boldsymbol{y}^*)}{\max(\#\boldsymbol{y}, \#\boldsymbol{y}^*)} \tag{5.13}$$

where $\#\boldsymbol{y}$ and $\#\boldsymbol{y}^*$ represent the number of nouns contained in $\boldsymbol{y}$ and $\boldsymbol{y}^*$, respectively. Notice that $\mathrm{NW}(\cdot, \cdot) \in [-1, 1]$. The final reward that we employ is a weighted version of CIDEr-D and the alignment score.

5.1.4 Controllability through a set of detections

The proposed architecture, so far, can generate a caption controlled by a sequence of region sets $\boldsymbol{R}$. To deal with the case in which the control signal is unsorted, *i.e.* a set of regions sets, we build a *sorting network* which can arrange the control signal in a candidate order, learning from data. The resulting sequence can then be given to the captioning network to produce the output caption (Fig. 5.3).

To this aim, we train a network which can learn a permutation, taking inspiration from Sinkhorn networks [133]. As shown in [133], the non-differentiable parameterization of a permutation can be approximated in terms of a differentiable relaxation, the so-called Sinkhorn operator. While a permutation matrix has exactly one entry of 1 in each row and each column, the Sinkhorn operator iteratively

[4] Although METEOR creates an alignment with respect to the reference caption, this is done for each unigram, thus mixing semantic and alignment errors.

normalizes rows and columns of any matrix to obtain a "soft" permutation matrix, *i.e.* a real-valued matrix close to a permutation one.

Given a set of region sets $\mathcal{R} = \{\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_N\}$, we learn a mapping from $\mathcal{R}$ to its sorted version $\boldsymbol{R}^*$. Firstly, we pass each element in $\mathcal{R}$ through a fully-connected network which processes every item of a region set independently and produces a single output feature vector with length N. By concatenating together the feature vectors obtained for all region sets, we thus get a $N \times N$ matrix, which is then passed to the Sinkhorn operator to obtain the soft permutation matrix $\boldsymbol{P}$. The network is then trained by minimizing the mean square error between the scrambled input and its reconstructed version obtained by applying the soft permutation matrix to the sorted ground-truth, *i.e.* $\boldsymbol{P}^T \boldsymbol{R}^*$.

At test time, we take the soft permutation matrix and apply the Hungarian algorithm [102] to obtain the final permutation matrix, which is then used to get the sorted version of $\mathcal{R}$ for the captioning network.

5.1.5 Implementation details

Language model and image features. We use a language model with two LSTM layers (Fig. 5.3): the input of the bottom layer is the concatenation of the embedding of the current word, the image descriptor, as well as the hidden state of the second layer. This layer predicts the context vector via the visual sentinel as well as the chunk-gate. The second layer, instead, takes as input the context vector and the hidden state of the bottom layer and predicts the next word.

To represent image regions, we use Faster R-CNN [154] with ResNet-101 [64]. In particular, we employ the model finetuned on the Visual Genome dataset [98] provided by [4]. As image descriptor, following the same work [4], we average the feature vectors of all the detections.

The hidden size of the LSTM layers is set to 1000, and that of attention layers to 512, while the input word embedding size is set to 1000.

Ground-truth chunk-shifting gate sequences. Given a sentence where each word of a noun chunk is associated to a region set, we build the chunk-shifting gate sequence $\{g_t^*\}_t$ by setting g_t^* to 1 on the last word of every noun chunk, and 0 otherwise. The region set sequence $\{\boldsymbol{r}_t^*\}_t$ is built accordingly, by replicating the same region set until the end of a noun chunk, and then using the region set of the next chunk. To compute the alignment score and for extracting dependencies, we use the spaCy NLP toolkit[5]. We use GloVe [147] as word vectors.

[5] https://spacy.io/

Sorting network. Given a scrambled sequence of N region sets, each region is encoded through a fully connected network which returns a N-dimensional descriptor. The fully connected network employs visual, textual and geometric features: the Faster R-CNN vector of the detection (2048-d), the GloVe embedding of the region class (300-d) and the normalized position and size of the bounding-box (4-d). The visual vector is processed by two layers (512-d, 128-d), while the textual feature is processed by a single layer (128-d). The outputs of the visual and textual branches are then concatenated with the geometric features and fed through another fully connected layer (256-d). A final layer produces the resulting N-dimensional descriptors. All layers have ReLU activations, except for the last fully-connected which has a $\tanh$ activation. In case the region set contains more than one detection, we average-pool the resulting N-dimensional descriptors to obtain a single feature vector for a region set.

Once the feature vectors of the scrambled sequence are concatenated, we get a $N \times N$ matrix, which is then converted into a "soft" permutation matrix $\boldsymbol{P}$ through the Sinkhorn operator. The operator processes a N-dimensional square matrix $\boldsymbol{X}$ by applying L consecutive row-wise and column-wise normalization, as follows:

$$S^0(\boldsymbol{X}) = \exp(\boldsymbol{X}) \tag{5.14}$$

$$S^l(\boldsymbol{X}) = \mathcal{T}c(\mathcal{T}_r(S^{l-1}(\boldsymbol{X}))) \tag{5.15}$$

$$\boldsymbol{P} := S^L(\boldsymbol{X}) \tag{5.16}$$

where $\mathcal{T}_r(\boldsymbol{X}) = \boldsymbol{X} \oslash (\boldsymbol{X}\mathbf{1}_N\mathbf{1}_N^T)$, and $\mathcal{T}_c(\boldsymbol{X}) = \boldsymbol{X} \oslash (\mathbf{1}_N\mathbf{1}_N^T\boldsymbol{X})$ are the row-wise and column-wise normalization operators, with $\oslash$ denoting element-wise division, $\mathbf{1}_N$ a column vector of N ones. At test time, once L normalizations ($L = 20$ in our experiments) have been performed, the resulting "soft" permutation matrix can be converted into a permutation matrix via the Hungarian algorithm [102]. An overview of the sorting network is shown in Fig. 5.4.

At training time, instead, we measure the mean square error between the scrambled sequence and its reconstructed version obtained by applying the soft permutation matrix to the sorted ground-truth sequence $\boldsymbol{R}^*$, *i.e.* $\boldsymbol{P}^T\boldsymbol{R}^*$. On the implementation side, all tensors are appropriately masked to deal with variable-length sequences and sets. We set the maximum length of input scrambled sequences to 10.

Training details. We use a weight of 0.2 for the word loss and 0.8 for the two chunk-level terms in Eq. 5.11. To train both the captioning model and the sorting

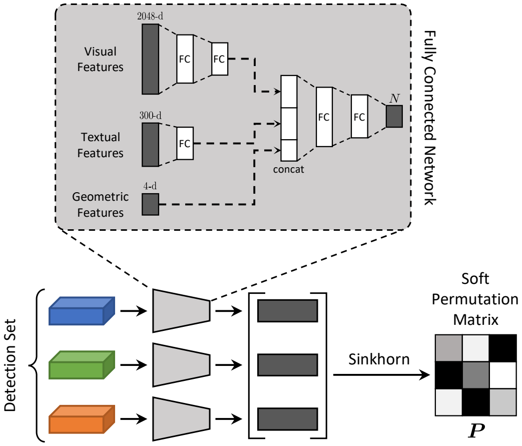


Figure 5.4: Schema of the sorting network.

network, we use the Adam optimizer with an initial learning rate of 5×10^{-4} decreased by a factor of 0.8 every epoch. For the captioning model, we run the reinforcement learning training with a fixed learning rate of 5×10^{-5}. We use a batch size of 100 for all our experiments. During caption decoding, we employ for all experiments the beam search strategy with a beam size of 5: similarly to what has been done when training with Reinforcement Learning, we sample from both output distribution to select the most probable sequence of actions. We use early stopping on validation CIDEr for the captioning network, and validation accuracy of the predicted permutations for the sorting network.

5.1.6 The COCO Entities dataset

We experiment with two datasets: Flickr30k Entities, which already contains the associations between chunks and image regions, and COCO, which we annotate semi-automatically. Table 5.1 summarizes the datasets we use.

Flickr30k Entities [149]. Based on Flickr30k [229], it contains $31,000$ images annotated with five sentences each. Entity mentions in the caption are linked with one or more corresponding bounding boxes in the image. Overall, $276,000$ manually annotated bounding boxes are available. In our experiments, we automatically associate each bounding box with the image region with maximum IoU among

COCO Entities (ours)	Train	Validation	Test
Nb. of captions	545,202	7,818	7,797
Nb. of noun chunks	1,518,667	20,787	20,596
Nb. of noun chunks per caption	2.79	2.66	2.64
Nb. of unique classes	1,330	725	730
Flickr30k Entities	Train	Validation	Test
Nb. of captions	144,256	5,053	4,982
Nb. of noun chunks	416,018	14,626	14,556
Nb. of noun chunks per caption	2.88	2.89	2.92
Nb. of unique classes	1,026	465	458

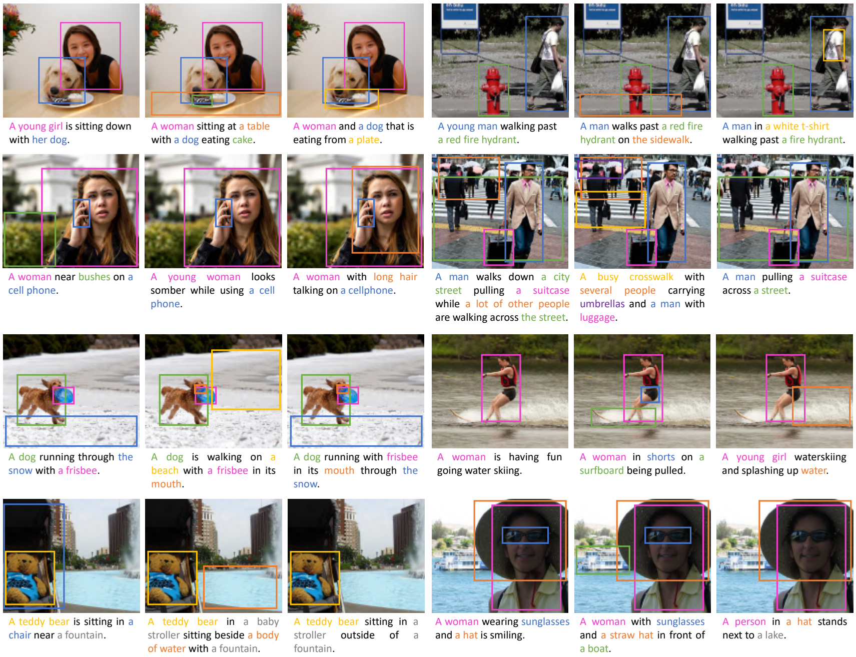
Table 5.1: Statistics on our COCO Entities dataset, in comparison with those of Flick30k Entities.

those detected by the object detector. We use the splits provided by Karpathy *et al.* [90].

COCO Entities. As seen previously, Microsoft COCO [116] contains more than $120,000$ images, each of them annotated with around five crowd-sourced captions. Here, we again follow the splits defined by [90] and automatically associate noun chunks with image regions extracted from the detector [154].

We firstly build an index associating each noun of the dataset with the five most similar class names, using word vectors. Then, each noun chunk in a caption is associated by using either its name or the base form of its name, with the first class found in the index which is available in the image. This association process, as confirmed by an extensive manual verification step, is generally reliable and produces few false positive associations. Naturally, it can result in region sets with more than one element (as in Flickr30k), and noun chunks with an empty region set. In this case, we fill empty training region sets with the most probable detections of the image and let the adaptive attention mechanism learn the corresponding association: we found that this procedure, overall, increases the final accuracy of the network rather than feeding empty region sets. Captions with missing associations are dropped in validation and testing.

Some examples of the additional annotations extracted from COCO are shown in Fig. 5.5. For the ease of visualization, we display a single region for chunk, even though multiple associations are possible. In the last row, we also report samples in which at least one noun chunk could not be assigned to any detection.



A young girl is sitting down with her dog.

A woman sitting at a table with a dog eating cake.

A woman and a dog that is eating from a plate.

A young man walking past a red fire hydrant.

A man walks past a red fire hydrant on the sidewalk.

A man in a white t-shirt walking past a fire hydrant.

A woman near bushes on a cell phone.

A young woman looks somber while using a cell phone.

A woman with long hair talking on a cellphone.

A man walks down a city street pulling a suitcase while a lot of other people are walking across the street.

A busy crosswalk with several people carrying umbrellas and a man with luggage.

A man pulling a suitcase across a street.

A dog running through the snow with a frisbee.

A dog is walking on a beach with a frisbee in its mouth.

A dog running with frisbee in its mouth through the snow.

A woman is having fun going water skiing.

A woman in shorts on a surfboard being pulled.

A young girl waterskiing and splashing up water.

A teddy bear is sitting in a chair near a fountain.

A teddy bear in a baby stroller sitting beside a body of water with a fountain.

A teddy bear sitting in a stroller outside of a fountain.

A woman wearing sunglasses and a hat is smiling.

A woman with sunglasses and a straw hat in front of a boat.

A person in a hat stands next to a lake.

Figure 5.5: Sample captions and corresponding visual groundings from the COCO Entities dataset. Different colors show a correspondence between textual chunks and image regions. Gray color indicates noun chunks for which a visual grounding could not be found, either for missing detections or for errors in the noun-class association.

5.1.7 Experimental settings

The experimental setting we employ is different from that of standard image captioning. In our scenario, indeed, the sequence of set of regions is a second input to the model which shall be consider when selecting the ground-truth sentences to compare against. Also, we employ additional metrics beyond the standard ones described in Sec. 4.1.3, like BLEU-4 [144], METEOR [9], ROUGE [115], CIDEr [193], and SPICE [2].

When evaluating the controllability with respect to a sequence, for each ground-truth regions-image input $(\boldsymbol{R}, \boldsymbol{I})$, we evaluate against all captions in the dataset which share the same pair. Also, we employ the alignment score (NW) to evaluate how the model follows the control input.

Similarly, when evaluating the controllability with respect to a set of regions, given a set-image pair $(\mathcal{R}, \boldsymbol{I})$, we evaluate against all ground-truth captions which have the same input. To assess how the predicted caption covers the control signal, we also define a soft intersection-over-union (IoU) measure between the ground-truth set of nouns and its predicted counterpart, recalling the relationships between region sets and noun chunks. Firstly, we compute the optimal assignment between the two set of nouns, using distances between word vectors and the Hungarian algorithm [102], and define an intersection score between the two sets as the sum of assignment profits. Then, recalling that set union can be expressed in function of an intersection, we define the IoU measure as follows:

$$\mathrm{IoU}(\boldsymbol{y}, \boldsymbol{y}^*) = \frac{\mathrm{I}(\boldsymbol{y}, \boldsymbol{y}^*)}{\#\boldsymbol{y} + \#\boldsymbol{y}^* - \mathrm{I}(\boldsymbol{y}, \boldsymbol{y}^*)} \tag{5.17}$$

where $\mathrm{I}(\cdot, \cdot)$ is the intersection score, and the $\#$ operator represents the cardinality of the two sets of nouns.

5.1.8 Baselines

To assess the performance of our method, we build two controllable baselines which can take as input a control signal in the form of a sequence or a set of image regions. Also, we build two different versions of our model in which we validate the effectiveness of its main components. As further baselines, we also compare against non-controllable captioning approaches, like FC-2K [155], Up-Down [4], and Neural Baby Talk [124].

Controllable LSTM. We start from a model without attention: an LSTM language model with a single visual feature vector. Then, we generate a sequential control input by feeding a flattened version of $\boldsymbol{R}$ to a second LSTM and taking the last hidden state, which is concatenated to the visual feature vector. The structure of the language model resembles that of [4], without attention.

Controllable Up-Down. In this case, we employ the full Up-Down model from [4], which creates an attentive distribution over image regions and make it controllable by feeding only the regions selected in $\boldsymbol{R}$ and ignoring the rest. This baseline is not sequentially controllable.

Ours without visual sentinel. To investigate the role of the visual sentinel and its interaction with the gate sentinel, in this baseline we ablate our model by removing the visual sentinel. The resulting baseline, therefore, lacks a mechanism to distinguish between visual and textual words.

Ours with single sentinel. Again, we ablate our model by merging the visual and chunk sentinel: a single sentinel is used for both roles, in place of $\boldsymbol{s}_t^c$ and $\boldsymbol{s}_t^v$.

5.1.9 Quantitative results

Controllability through a sequence of detections. Firstly, we show the performance of our model when providing the full control signal as a sequence of region sets. Table 5.3 shows results on COCO Entities, in comparison with the aforementioned approaches. We can see that our method achieves state-of-the-art results on all automatic evaluation metrics, outperforming all baselines both in terms of overall caption quality and in terms of alignment with the control signal. Using the cross-entropy pre-training, we outperform the Controllable LSTM and Controllable Up-Down by 32.0 on CIDEr and 0.112 on NW. Optimizing the model with CIDEr and NW further increases the alignment quality while maintaining outperforming results on all metrics, leading to a final 0.649 on NW, which outperforms the Controllable Up-Down baseline by a 0.25. Recalling that NW ranges from -1 to 1, this improvement amounts to a 12.5% of the full metric range.

In Table 5.4, we instead show the results of the same experiments on Flickr30k Entities. Also on this manually annotated dataset, our method outperforms all the compared approaches by a significant margin, both in terms of caption quality and alignment with the control signal.

Controllability through a set of detections. We then assess the performance of our model when controlled with a set of detections. Tables 5.5 and 5.6 show the performance of our method in this setting, respectively on COCO Entities and Flickr30k Entities. We notice that the proposed approach outperforms all baselines and compared approaches in terms of IoU, thus testifying that we are capable of respecting the control signal more effectively. This is also combined with better captioning metrics, which indicate higher semantic quality.

We observe that the CIDEr+NW fine-tuning approach is effective on all settings, and that our model outperforms by a clear margin the baselines both when controlled via a sequence and when controlled by a scrambled set of regions, regardless of the careful choice of the baselines. The performance of the Controllable LSTM baseline is constantly significantly lower than that of the Controllable Up-Down, thus indicating both the importance of an attention mechanism and that of having a good representation of the control signal. The Controllable Up-Down

Method	Samples	B-4	M	R	C	S
AG-CVAE [206]	20	**47.1**	30.9	63.8	130.8	24.4
POS [34]	20	44.9	36.5	67.8	146.8	27.7
Ours	20	44.8	**36.6**	**68.9**	**156.5**	**30.9**

Table 5.2: Diversity performance on the test portion of COCO.

baseline, however, shows lower performance when compared to our approach, in both sequence- and set-controlled scenarios.

Diversity evaluation. We also assess the diversity of the generated captions, comparing with the most recent approaches that focus on diversity. In particular, the variational autoencoder proposed in [206] and the approach of [34], which allows diversity and controllability by feeding PoS sequences. To test our method on a significant number of diverse captions, given an image we take all regions which are found in control region sets, and take the permutations which result in captions with higher log-probability. This approach is fairly similar to the sampling strategy used in [34], even if ours considers region sets. Then, we follow the experimental approach defined in [206, 34]: each ground-truth sentence is evaluated against the generated caption with the maximum score for each metric. Higher scores, thus, indicate that the method is capable of sampling high accuracy captions. Results are reported in Table 5.2, where to guarantee the fairness of the comparison, we run this experiments on the full COCO test split. As it can be seen, our method can generate significantly diverse captions.

5.1.10 Qualitative results

Finally, Fig. 5.6 and 5.7 report qualitative results on COCO Entities. The same image is reported multiple times with different control inputs. As it can be seen, our method generates multiple captions for the same image, and can accurately follow the control input.

Method	Cross-Entropy Loss						CIDEr Optimization						CIDEr + NW Optimization					
	B-4	M	R	C	S	NW	B-4	M	R	C	S	NW	B-4	M	R	C	S	NW
FC-2K† [155]	10.4	17.3	36.8	98.3	25.2	0.257	12.3	18.5	39.6	117.5	26.9	0.273	-	-	-	-	-	-
Up-Down† [4]	12.9	19.3	40.0	119.9	29.3	0.296	14.2	20.0	42.1	133.9	30.0	0.310	-	-	-	-	-	-
Neural Baby Talk† [124]	12.9	19.2	40.4	120.2	29.5	0.305	-	-	-	-	-	-	-	-	-	-	-	-
Controllable LSTM	11.4	18.1	38.5	106.8	27.6	0.275	12.8	18.9	40.9	123.0	28.5	0.290	12.9	19.3	41.3	124.0	28.9	0.341
Controllable Up-Down	17.3	23.0	46.7	161.0	39.1	0.396	17.4	22.9	47.1	168.5	39.0	0.397	17.9	23.6	48.2	171.3	40.7	0.443
Ours *w/* single sentinel	20.0	23.9	51.1	183.3	43.9	0.480	21.7	25.3	54.5	202.6	47.6	0.606	21.3	25.3	54.5	201.1	48.1	0.648
Ours *w/o* visual sentinel	20.8	**24.4**	52.4	191.2	45.1	**0.508**	22.2	25.4	55.0	206.2	47.6	0.607	21.5	25.1	54.7	202.2	48.1	0.639
Ours	**20.9**	**24.4**	**52.5**	**193.0**	**45.3**	**0.508**	**22.5**	**25.6**	**55.1**	**210.1**	**48.1**	**0.615**	**22.3**	**25.6**	**55.3**	**209.7**	**48.5**	**0.649**

Table 5.3: Controllability via a sequence of regions, on test portion of COCO Entities. NW refers to the visual chunk alignment measure defined in Sec. 5.1.3. The † marker indicates non-controllable methods.

Method	Cross-Entropy Loss						CIDEr Optimization						CIDEr + NW Optimization					
	B-4	M	R	C	S	NW	B-4	M	R	C	S	NW	B-4	M	R	C	S	NW
Neural Baby Talk† [124]	8.5	13.5	31.7	53.9	17.9	0.090	-	-	-	-	-	-	-	-	-	-	-	-
Controllable LSTM	6.5	12.0	29.6	40.4	15.7	0.078	6.7	12.1	30.0	45.5	15.8	0.079	6.5	12.6	30.2	43.5	15.8	0.124
Controllable Up-Down	10.1	15.2	34.9	69.2	21.6	**0.158**	10.1	14.8	35.0	69.3	21.2	0.148	10.4	15.2	35.2	69.5	21.7	0.190
Ours *w/* single sentinel	11.0	**15.5**	36.3	71.7	22.6	0.134	11.2	15.8	37.9	77.9	22.9	0.199	10.7	16.1	38.1	76.5	22.8	0.260
Ours *w/o* visual sentinel	10.8	14.9	35.4	69.3	22.2	0.142	11.1	15.5	36.8	75.0	22.2	0.197	11.1	15.5	37.2	74.7	22.4	0.244
Ours	**11.3**	15.4	**36.9**	**74.5**	**23.4**	0.152	**12.4**	**16.6**	**38.8**	**83.7**	**23.5**	**0.221**	**12.5**	**16.8**	**38.9**	**84.0**	**23.5**	**0.263**

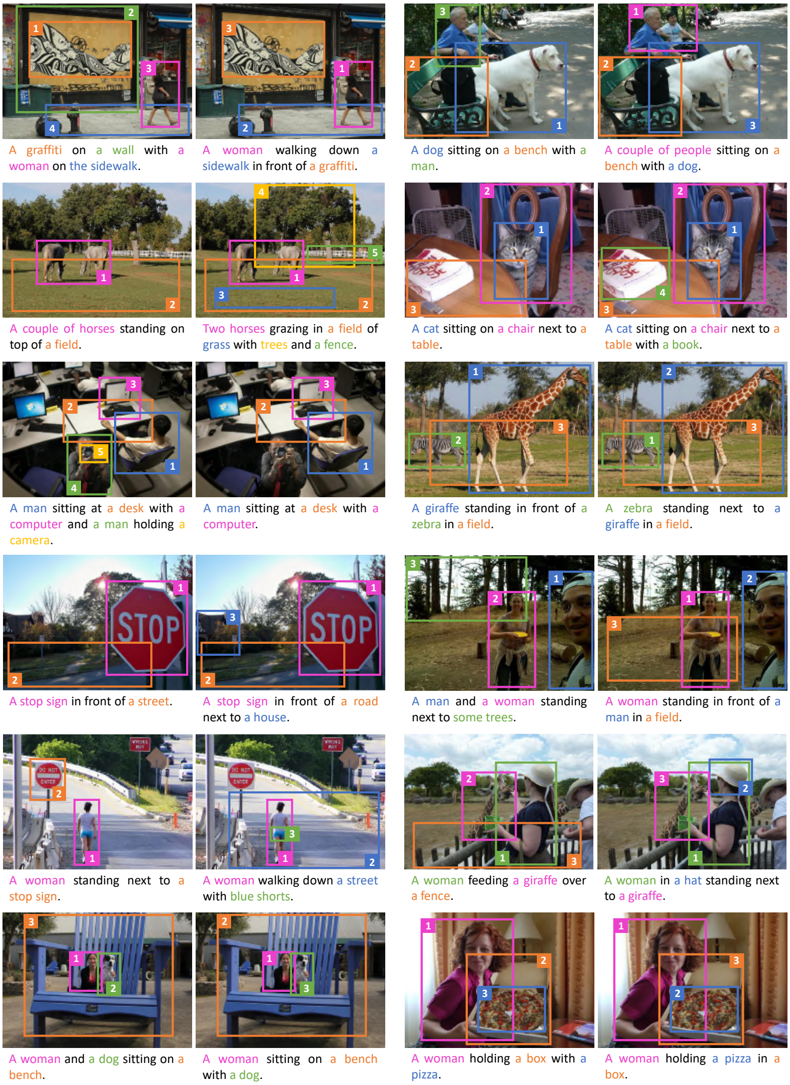
Table 5.4: Controllability via a sequence of regions, on the test portion of Flickr30K Entities. NW refers to the visual chunk alignment measure defined in Sec. 5.1.3. The † marker indicates non-controllable methods.

	Cross-Entropy Loss						CIDEr Optimization						CIDEr + NW Optimization					
Method	B-4	M	R	C	S	IoU	B-4	M	R	C	S	IoU	B-4	M	R	C	S	IoU
FC-2K† [155]	-	-	-	-	-	-	12.5	18.5	39.6	116.5	26.6	61.0	-	-	-	-	-	-
Up-Down† [4]	-	-	-	-	-	-	14.4	20.0	42.2	132.8	29.7	63.2	-	-	-	-	-	-
Neural Baby Talk† [124]	13.1	19.2	40.5	119.1	29.2	62.6	-	-	-	-	-	-	-	-	-	-	-	-
Controllable LSTM	11.5	18.1	38.5	105.8	27.1	60.7	12.9	18.9	40.9	122.0	28.2	62.0	12.9	19.3	41.3	123.4	28.7	0.642
Controllable Up-Down	17.5	23.0	46.9	160.6	38.8	69.2	17.7	22.9	47.3	167.6	38.7	69.4	**18.1**	23.6	48.4	170.5	40.4	71.6
Ours *w/* single sentinel	16.9	22.6	46.9	159.6	40.9	70.2	17.9	23.7	48.7	171.1	43.5	74.4	17.4	23.6	48.4	168.4	43.7	75.4
Ours *w/o* visual sentinel	**17.7**	23.1	47.9	166.6	**42.1**	71.3	18.1	23.7	48.9	172.5	43.3	74.2	17.6	23.4	48.5	168.9	43.6	75.3
Ours	**17.7**	**23.2**	**48.0**	**168.3**	**42.1**	**71.4**	**18.5**	**23.9**	**49.0**	**176.7**	**43.8**	**74.5**	18.0	**23.8**	**48.9**	**173.3**	**44.1**	**75.5**

Table 5.5: Controllability via a set of regions, on the test portion of COCO Entities. IoU refers to the soft intersection-over-union measure defined in Sec. 5.1.7. The † marker indicates non-controllable methods.

	Cross-Entropy Loss						CIDEr Optimization						CIDEr + NW Optimization					
Method	B-4	M	R	C	S	IoU	B-4	M	R	C	S	IoU	B-4	M	R	C	S	IoU
Neural Baby Talk† [124]	8.6	13.5	31.9	53.8	17.8	49.9	-	-	-	-	-	-	-	-	-	-	-	-
Controllable LSTM	6.7	12.0	29.8	41.0	15.6	48.8	6.8	12.1	30.2	45.4	15.6	49.0	6.4	12.5	30.2	42.9	15.6	50.8
Controllable Up-Down	**10.1**	**15.2**	35.1	**68.8**	21.5	**53.6**	10.2	14.8	35.3	69.1	21.1	52.9	10.5	15.2	35.5	69.5	21.6	54.8
Ours *w/* single sentinel	**10.1**	**15.2**	**35.5**	67.5	21.7	52.5	10.1	15.3	36.1	68.9	21.7	53.5	9.5	15.2	35.8	65.6	21.2	**55.0**
Ours *w/o* visual sentinel	9.7	14.5	34.4	63.1	21.0	52.2	9.9	14.7	34.8	65.5	20.8	52.9	9.8	14.8	35.0	64.2	20.9	54.3
Ours	9.9	14.9	35.3	67.3	**22.2**	52.7	**10.8**	**15.7**	**36.4**	**71.3**	**22.0**	**53.9**	**10.9**	**15.8**	**36.2**	**70.4**	**21.8**	**55.0**

Table 5.6: Controllability via a set of regions, on the test portion of Flickr30K Entities. IoU refers to the soft intersection-over-union measure defined in Sec. 5.1.7. The † marker indicates non-controllable methods.



A graffiti on a wall with a woman on the sidewalk.

A woman walking down a sidewalk in front of a graffiti.

A dog sitting on a bench with a man.

A couple of people sitting on a bench with a dog.

A couple of horses standing on top of a field.

Two horses grazing in a field of grass with trees and a fence.

A cat sitting on a chair next to a table.

A cat sitting on a chair next to a table with a book.

A man sitting at a desk with a computer and a man holding a camera.

A man sitting at a desk with a computer.

A giraffe standing in front of a zebra in a field.

A zebra standing next to a giraffe in a field.

A stop sign in front of a street.

A stop sign in front of a road next to a house.

A man and a woman standing next to some trees.

A woman standing in front of a man in a field.

A woman standing next to a stop sign.

A woman walking down a street with blue shorts.

A woman feeding a giraffe over a fence.

A woman in a hat standing next to a giraffe.

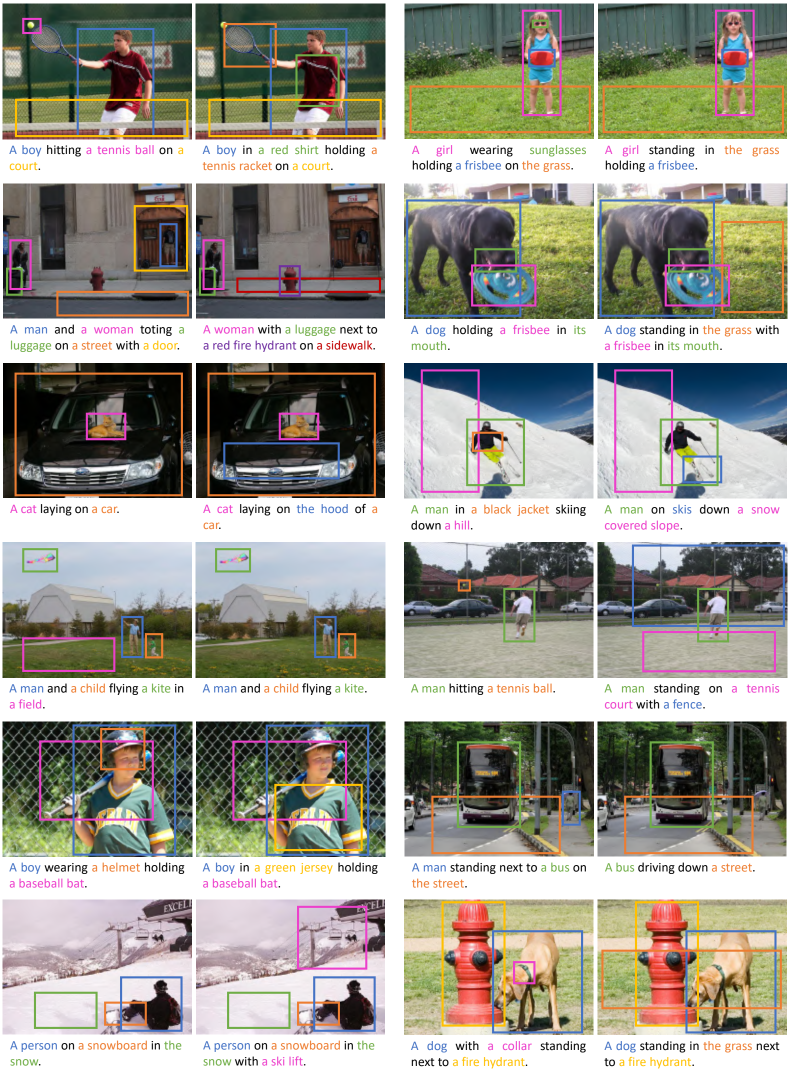
A woman and a dog sitting on a bench.

A woman sitting on a bench with a dog.

A woman holding a box with a pizza.

A woman holding a pizza in a box.

Figure 5.6: Sample results of controllability via a sequence of regions. Different colors and numbers show the control sequence and the associations between chunks and regions.



A boy hitting a tennis ball on a court.

A boy in a red shirt holding a tennis racket on a court.

A girl wearing sunglasses holding a frisbee on the grass.

A girl standing in the grass holding a frisbee.

A man and a woman toting a luggage on a street with a door.

A woman with a luggage next to a red fire hydrant on a sidewalk.

A dog holding a frisbee in its mouth.

A dog standing in the grass with a frisbee in its mouth.

A cat laying on a car.

A cat laying on the hood of a car.

A man in a black jacket skiing down a hill.

A man on skis down a snow covered slope.

A man and a child flying a kite in a field.

A man and a child flying a kite.

A man hitting a tennis ball.

A man standing on a tennis court with a fence.

A boy wearing a helmet holding a baseball bat.

A boy in a green jersey holding a baseball bat.

A man standing next to a bus on the street.

A bus driving down a street.

A person on a snowboard in the snow.

A person on a snowboard in the snow with a ski lift.

A dog with a collar standing next to a fire hydrant.

A dog standing in the grass next to a fire hydrant.

Figure 5.7: Additional sample results of controllability via a set of regions. Different colors show the control set and the associations between chunks and regions.

5.2 Controlling caption generation via people identity

The findings of the previous section have shown that it is possible to control a captioning model through an ordered sequence or an unsorted set of regions by considering the decomposition of a sentence into noun chunks, and grounding chunks to the corresponding visual regions. This approach can be easily applied to images, as seen before, and in practice can be also adapted to the context of videos, given the presence of a large scale video captioning dataset in which parts of the captions are grounded to the corresponding bounding boxes in the video frames [240].

Focusing on bounding boxes containing a person, another level of controllability that may be desired for a captioning model is the ability to link a bounding box of a person to his corresponding identity, thus constraining the captioning model to mention people appearing in the scene with their proper names. In the context of videos, this can be achieved by exploiting large scale movie description datasets in which captions come with character names [186, 160]. However, it is a usual practice in video captioning to replace character names with a generic "someone" tag thus ignoring the naming task during the generation of the textual descriptions. The underlying reason has to be found in the structure of current video captioning models, which are not designed to take into account the visual aspect of each movie character. In these architectures, using captions in which character names are not replaced would simply result in additional dictionary entries, ignoring the fundamental relationship between the character names and their visual appearance, and possibly invalidating the significance of evaluation metrics.

Developing video captioning architectures with naming capabilities requires to deal with several sub-tasks at the time of caption generation. In particular, the architecture has to detect, track, and recognize people within a set of characters. Furthermore, the language model has to be aware of the semantic structure of the caption and has to coordinate itself with the feature extraction part, to detect the presence of a character in the scene.

Unfortunately, current movie description datasets do not contain any kind of supervision that joins the textual mentions and the visual appearances of the characters. Without a supervision between the textual and the visual domain, training video captioning algorithms with naming capabilities is particularly challenging. Indeed, many characters and background actors may appear in the same scene, while only few characters are mentioned in the video descriptions. Therefore, an additional form of supervision that associates the textual and the visual information, by linking characters' visual appearances with their textual mentions, is necessary for the development of novel movie description architectures.

In this section, we introduce the M-VAD Names dataset, specifically designed for supporting the development of video captioning architectures with naming capabilities. The dataset, which is an extension of the well-known Montreal Video Description Dataset [186], consists of visual face tracks and the association between them and the characters' textual mentions.

In addition, we propose a multimodal architecture that addresses the task of replacing generic “someone” tags with proper character names in previously generated captions. The model combines advanced natural language processing tools and state-of-the-art deep neural models for action and face recognition. Experimental results enlighten and quantify many of the challenges associated with the task, and demonstrate the effectiveness of the proposed strategy. Finally, we also show how the proposed model can be applied outside of the M-VAD Names dataset, by extending the evaluation on an additional set of movies.

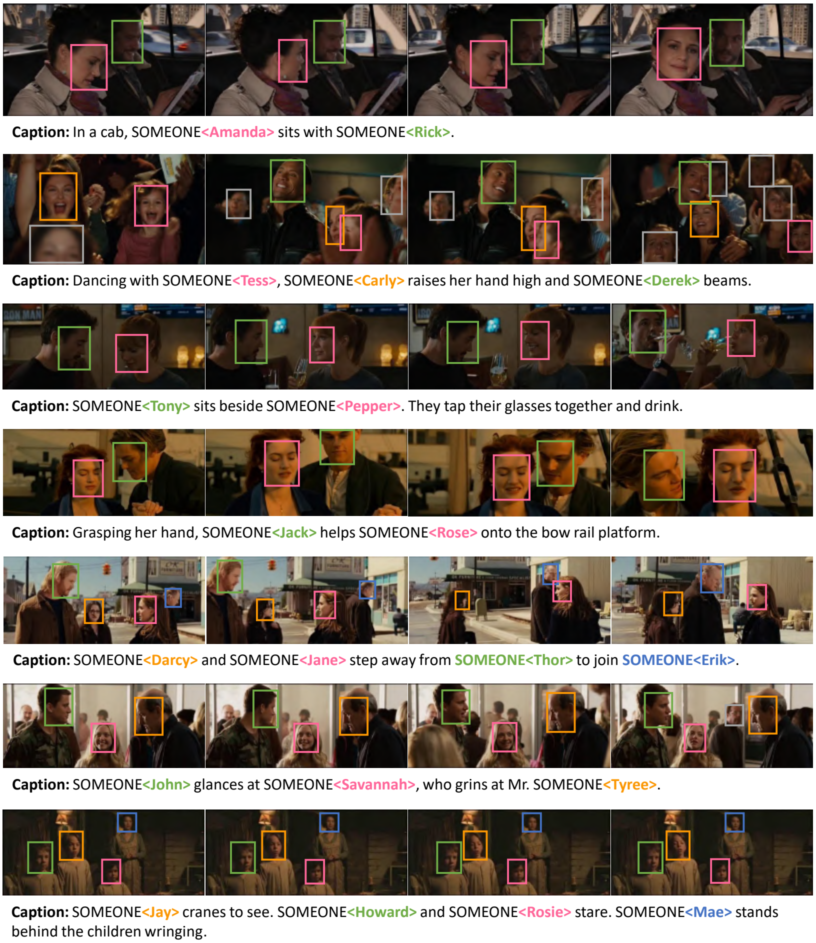
5.2.1 The M-VAD Names dataset

We collect the M-VAD Names dataset, a new set of annotations for the Montreal Video Annotation Dataset (M-VAD) [186] supporting the development of video captioning architectures with naming capabilities. The dataset contains the annotations of the characters' visual appearances, in the form of tracks of face bounding boxes, and the associations with the characters' textual mentions, when available. In particular, we detect and annotate the visual appearances of characters in each video clip of each movie through a semi-automatic approach. Also, we correct some errors in the original M-VAD annotations in order to include more characters in our dataset. Figure 5.8 shows some representative samples of the collected dataset.

In the following, we describe the annotation procedure, from the detection of the face tracks to the semi-automatic annotation process, the generation of the train, validation, and test split, and the method used to extend the original M-VAD captions. Finally, we report statistics and analyses of the proposed dataset.

Face detection and tracking

The first stage of the annotation procedure is the extraction of the face tracks, sequences of consecutive face detections belonging to the same character.



Caption: In a cab, SOMEONE<Amanda> sits with SOMEONE<Rick>.

Caption: Dancing with SOMEONE<Tess>, SOMEONE<Carly> raises her hand high and SOMEONE<Derek> beams.

Caption: SOMEONE<Tony> sits beside SOMEONE<Pepper>. They tap their glasses together and drink.

Caption: Grasping her hand, SOMEONE<Jack> helps SOMEONE<Rose> onto the bow rail platform.

Caption: SOMEONE<Darcy> and SOMEONE<Jane> step away from SOMEONE<Thor> to join SOMEONE<Erik>.

Caption: SOMEONE<John> glances at SOMEONE<Savannah>, who grins at Mr. SOMEONE<Tyree>.

Caption: SOMEONE<Jay> cranes to see. SOMEONE<Howard> and SOMEONE<Rosie> stare. SOMEONE<Mae> stands behind the children wringing.

Figure 5.8: Samples extracted from the M-VAD Names dataset. For each video clip, face tracks are annotated and associated to proper character names mentioned inside the caption. Face tracks that are not associated to a specific movie character (*i.e.* unknown people) are represented in gray color.

To collect them, we sequentially detect faces in each frame of each video clip using the face detector presented in [236]. Then, tracks are formed by grouping consecutive detections belonging to the same character. Specifically, for each detected face, a tracker [7] is initialized with the bounding box corresponding to the detection and a new face track is created. Then, in the following frames, each initialized tracker is updated and each face detection is compared with each tracker prediction using the Intersection over Union (IoU) measure. Applying the Kuhn-Munkres algorithm [102], each face detection is associated with the most overlapping tracker prediction. Then, if the IoU value between a face detection and the associated tracker prediction is over a threshold (called t_{IoU}) and the appearance difference between the detection and the last element of the track is below a threshold (called t_{visual}), the face detection is added to the related track and the tracker is re-initialized on the new detection. Otherwise, a new tracker and a new track are initialized. We empirically set the IoU threshold value to 0.5, while we found that a pixel-wise difference between face detections and last added element of tracks is, when used with a threshold of 10, a sufficient appearance measure to discard most of the errors.

If a tracker prediction is not associated with any face detection (due to occlusions or scene changes, for instance) or the association does not respect the constraints reported above, the tracker prediction is added to the track. If the tracker is not associated again with a face detection for the following 8 frames, or before the end of the video clip, the tracker predictions that were added to the track are removed and the tracker is detached.

At the end of each video clip, face tracks that are composed by less than 8 frames are discarded, as well as tracks that are fully contained in another one. The overall algorithm for the detection and tracking of face tracks is reported in Algorithm 1.

Movie character annotations

After the extraction, each face track has to be labelled with the name of a character from the corresponding movie. To facilitate the annotation procedure, which would require to label every single track with respect to the list of characters of a movie, we firstly cluster similar faces using an embedding space in which similar faces (*i.e.* faces of the same person) lie together, while dissimilar faces (*i.e.* faces of different people) lie far by a clear margin. Then, clusters are manually verified, to guarantee that each cluster contains only tracks from a single character. The annotator is finally asked to match each cluster with the corresponding character.

To obtain the embedding space, we extract face feature vectors using a deep neural model inspired by FaceNet [164] and trained on a sub-set of the MS-Celeb-

Algorithm 1: Track extraction algorithm.

```
Data: M-VAD dataset
Result: Tracks containing characters' visual appearances
foreach video clip in M-VAD dataset do
    foreach frame in video clip do
        foreach initialized tracker do Update the tracker prediction;
        Detect faces in the frame (MTCNN architecture);
        Calc the IoU between each prediction and each detection;
        Solve the detection-tracker association (Kuhn-Munkres algorithm);
        foreach face detection which is not associated do
            Create a new track with the face detection;
            Define a new tracker linked to the new track;
            Initialize the tracker on the face detection;
        end
        foreach initialized tracker do
            if tracker is associated and $IoU > t_{IoU}$ and visual difference $< t_{visual}$ then
                Add the detection to the track associated with the tracker;
                Re-initialize the tracker with the new face detection;
                Reset the tracker counter;
            else
                if $tracker_{counter} < t_{counter}$ then
                    Add the tracker prediction to the associated track;
                    Increment the tracker counter;
                else
                    Remove the last $tracker_{counter}$ items from the track;
                    Detach the tracker;
                end
            end
        end
    end
    foreach initialized tracker with $tracker_{counter} > 0$ do
        Remove the last $tracker_{counter}$ items from the track;
    end
end
```

1M dataset [59]. Then, for each movie, we aggregate tracks containing similar faces by applying a hierarchical clustering algorithm, based on the euclidean distance and on the Ward's minimum variance method [213]. Since each track is composed by a variable number of bounding boxes, we apply the clustering algorithm on the mean of their embeddings. We exclude the smallest tracks (*i.e.* tracks with a side lower than 28 pixels) from the automatic clustering process as we found that their features are not reliable.

Once clusters have been manually verified, so to contain one single character, each cluster is either assigned to a character of the movie, or rejected as *wrong* (if it contains false positive detections by the face detector or by the tracker), or as *unknown* (if the character is a background actor or if human annotators are not able to recognize it). To get the list of characters of each movie, we use IMDb. Finally, each annotation is checked by at least three different people in order to prevent as many errors as possible. At the end of the process, every track, corresponding to a character appearance in the movie, is associated to his textual mention in the M-VAD captions, if present. To this end, we manually build a dictionary which maps every character to the set of his names in a movie, and use it for matching textual mentions with tracks. For instance, in the *Robin Hood* movie, Friar Tuck is sometimes referred to as *Tuck* and sometimes as *Friar*; similarly, in *Snow Flower and the Secret Fan*, Nina is sometimes called *Lily*, but also *Flower* and *Sophia*. Once these ambiguities have been solved through the dictionary, each track is mapped to the correct character identity.

M-VAD captions

Along with the M-VAD Names dataset, we release an extended version of the original M-VAD movie descriptions. In particular, during the annotation process, we found that several annotated characters were not tagged as “someone” in the original M-VAD captions but were mentioned with their proper names. The corresponding captions could be thus considered as errors of the original M-VAD dataset since, as mentioned, existing video captioning architectures are not able to mention a character with its proper name.

To fix this problem, we add new annotations (*i.e.* new “someone” tags) in every movie caption for each mentioned character that is not annotated in the original M-VAD, but that we have correctly annotated in the previous stage of the process. Overall, we fix $1,253$ M-VAD descriptions by adding 116 unique characters that appeared in the original captions but that were not tagged as “someone”.

Training, validation and test splits

Original M-VAD training, validation, and test set are obtained by splitting the 92 movies in three disjoint parts, in order to be able to train video captioning algorithms on a sub-set of movies and to validate and test them on different movies, effectively testing the generalization capabilities of the models. However, when considering the naming task, video clips of the same movie have to be in every

	Overall	Avg. per movie	Avg. per character
Train videos	19,023	207	-
Validation videos	2,976	32	-
Test videos	2,836	31	-
Mentioned characters	1,566	17	-
Annotated characters	1,093	12	-
Mentions	34,388	374	23
Tracks	63,442	690	62
Bounding boxes	2,636,595	28,658	2,587

Table 5.7: Overall statistics of the M-VAD Names dataset. Along with the number of videos in the train, validation, and test splits, we report the number of mentioned characters, annotated characters, textual mentions, face tracks, and annotated bounding boxes.

split, so that the captioning algorithms can learn the visual appearance of the characters on the training set and apply it on the validation and test set. Therefore, we release the official training, validation, and test set for the M-VAD Names dataset.

In particular, we generate the splits applying the following constraints. Firstly, we forced every movie to have 80% of the video clips into the training set, 10% into the validation set and 10% into the test set. Secondly, we split the video clips with only one mention, and the video clips with two or more mentions using the same proportions. Finally, we enforced, when possible, to have at least one video clip for every character in each sub-set of the dataset, giving priority to the training set. Applying this set of soft constraints, training, validation, and test set tend to respectively have 80%, 10%, and 10% of video clips of each movie, of video clips of each character, of video clips with one mention, and of video clips with two or more mentions.

Statistics

In Table 5.7, we report the main statistics of the proposed dataset. Overall, the dataset contains $24,835$ annotated video clips, $1,566$ unique mentioned characters, and $1,093$ unique annotated characters ($1,093$ instead of 908). With respect to the $34,388$ mentions in the screenplays, the movie characters appear in $63,442$ different face tracks resulting in more than 2 millions annotated bounding boxes.

These statistics refer to the tracks associated with a “someone” tag in the caption, while the dataset contains every annotated track, regardless of the existence of the association with a caption tag. Considering every annotated track, the dataset is composed by more than 100k face tracks and 4M annotated bounding boxes. This approach has three main advantages. First, additional annotated tracks can be used for learning the characters' visual appearances since they do not depend on the captions. Then, additional annotated tracks can be linked to caption nouns and pronouns (despite the missing "someone" tags) by applying fine NLP or co-reference resolution algorithms. Finally, thanks to the high number of face bounding boxes and their association to specific characters/actors, the dataset can be used for other tasks as well, like action recognition and training of visual-semantic spaces on videos.

5.2.2 Replacing the "someone"

With the M-VAD Names dataset, we aim to provide sufficient labelled data to allow the development of video captioning architectures with naming capabilities, *i.e.* architectures able to correctly generate captions mentioning proper character names. In this section, we address a strictly related problem that shares many of the challenges of the video captioning with the naming task, yet without considering the generation of movie descriptions. In particular, we investigate the task of replacing the "someone" tags in existing captions with proper character names. Therefore, we need to analyze both video clips and textual descriptions to find the correct association between visual and textual actions computed by the characters.

A summary of the proposed method is shown in Figure 5.9. Firstly, we parse ground-truth captions, in which each character name is replaced with a "someone" tag, with an NLP parser in order to extract each verb associated to a "someone" tag. Then, by using the M-VAD Names dataset, in which characters' visual appearances are associated to their textual mentions, we train a neural network that projects visual actions and textual verbs into a joint multimodal embedding space in which the distance between a verb and a track is inversely proportional to their similarity. After assigning each verb to a visual track, a face recognition algorithm is applied to identify the character and to replace the "someone" tag with the correct character name, by concluding the replacement task.

Textual-visual embedding space

We project input tracks and verbs into a shared multimodal embedding space in which the distance between a verb and a track is inversely proportional to their

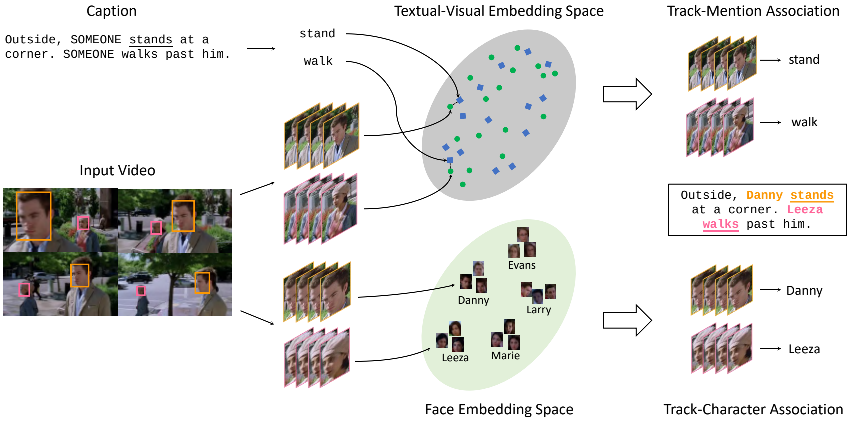

Caption
Outside, SOMEONE stands at a corner. SOMEONE walks past him.
Input Video
stand
walk
Textual-Visual Embedding Space
Track-Mention Association
stand
walk
Outside, Danny stands at a corner. Leeza walks past him.
Evans
Danny
Larry
Leeza
Marie
Face Embedding Space
Danny
Leeza
Track-Character Association

Figure 5.9: Summary of our approach for replacing "someone" tags with proper character names. We project verbs and body tracks in a unified embedding space to find the best verb-track association. Similarly, we project face tracks in an embedding space to recognize the corresponding characters, completing the replacement task.

similarity.

Regarding the visual data representation, we use the 4096-dimensional output of the last but one fully connected layer of the C3D network [187], pre-trained on the Sports-1M dataset [91], as the visual features for the tracks. In particular, we expand the spatial area of the face tracks to include the upper-body of the subject, as done in [12], and we split the tracks in 16-frame long sub-sequences with a stride of 8 frames. Moreover, for each sub-sequence, we fixed the dimension and the position of the track bounding box as the smallest area containing every body bounding box of the sub-sequence. We therefore obtain spatially and temporarily continuous sub-sequences of 16 frames for each original face track. We compute the C3D visual features for each of them. At training time, in order to increase the generalization capabilities of the network, we select a random 16-frame long sub-sequence each time the track is selected, while we average feature vectors of each track, obtaining a single 4096-dimensional vector, at validation and test time.

Regarding the textual data representation, we convert every verb to a 300-dimensional semantic feature vector by using the GloVe embeddings [147] provided by spaCy[6], an open-source software library for natural language processing.

Then, we project the visual and the textual features by passing through a fully connected neural network with two branches. The network is trained forcing every track and its corresponding verb to have close projections and forcing every track and every non-corresponding verb to be far by at least a margin α. We can express the cost functions of this formulation as:

$$p(\mathbf{a}, \mathbf{b}) = \|\phi_v(\mathbf{a}) - \phi_t(\mathbf{b})\|_2^2, \tag{5.18a}$$

$$n(\mathbf{a}, \mathbf{b}) = \max(\alpha - \|\phi_v(\mathbf{a}) - \phi_t(\mathbf{b})\|_2^2, 0) \tag{5.18b}$$

or, using a triplet formulation, as:

$$t(\mathbf{a}, \mathbf{b}, \mathbf{b}^-) = \max(\|\phi_v(\mathbf{a}) - \phi_t(\mathbf{b})\|_2^2 - \|\phi_v(\mathbf{a}) - \phi_t(\mathbf{b}^-)\|_2^2 + \alpha, 0) \tag{5.19a}$$

$$v(\mathbf{a}, \mathbf{b}, \mathbf{a}^-) = \max(\|\phi_v(\mathbf{a}) - \phi_t(\mathbf{b})\|_2^2 - \|\phi_v(\mathbf{a}^-) - \phi_t(\mathbf{b})\|_2^2 + \alpha, 0) \tag{5.19b}$$

where $\phi_v(\cdot)$ and $\phi_t(\cdot)$ are respectively the visual-branch and the textual-branch projection function, while $\mathbf{a}$ and $\mathbf{b}$ are the features of a track and of a verb. We denote with $\mathbf{b}^-$ the features of a verb that does not correspond with $\mathbf{a}$ (*i.e.* a verb that is different from $\mathbf{b}$) and with $\mathbf{a}^-$ the features of a track that does not correspond with $\mathbf{b}$ (*i.e.* a track that is not associated to the verb $\mathbf{b}$).

When using the first formulation, a commonly used loss function is the so-called siamese loss, defined as:

$$L = \sum_{i=1}^{N} p(\mathbf{a}_i, \mathbf{b}_i) + n(\mathbf{a}_i, \mathbf{b}_i^-) \tag{5.20}$$

where N is the number of valid verb-track pairs. When using the latter formulation, instead, the so-called triplet loss, or one of its variants, is usually used. The one-term formulation is:

$$L = \sum_{i=1}^{N} t(\mathbf{a}_i, \mathbf{b}_i, \mathbf{b}_i^-) \tag{5.21}$$

Recently, a two-term variation has been proposed and successfully employed in [171, 94, 90, 242]:

$$L = \sum_{i=1}^{N} t(\mathbf{a}_i, \mathbf{b}_i, \mathbf{b}_i^-) + v(\mathbf{a}_i, \mathbf{b}_i, \mathbf{a}_i^-) \tag{5.22}$$

[6] https://spacy.io

Addressing our particular task, however, we do not only want to force the proximity of the corresponding tracks and verbs and a minimum distance between the non-corresponding ones, but we also want to force valid verbs and wrong tracks to be far. In particular, we formulate the following four-terms loss function:

$$L = \sum_{i=1}^{N} p(\mathbf{a}_i, \mathbf{b}_i) + n(\mathbf{a}_i, \mathbf{b}_i^-) + p(\mathbf{a}_i^+, \mathbf{b}_i) + n(\mathbf{a}_i^w, \mathbf{b}_i) \tag{5.23}$$

where N is the number of valid verb-track pairs, $\mathbf{b}_i^-$ are the features of a verb that does not correspond to the visual track, $\mathbf{a}_i^+$ are the features of a track (different from $\mathbf{a}_i$) associated to the same verb $\mathbf{b}_i$, and $\mathbf{a}_i^w$ are the features of a "wrong" track.

Furthermore, since the goal of the whole architecture is to distinguish verb-track pairs extracted from the same video clip, we introduce the following sampling procedure. Given that we have to select a wrong verb $\mathbf{b}_i^-$ (*i.e.* a verb that is different from the verb $\mathbf{b}_i$) and a positive track $\mathbf{a}_i^+$ (*i.e.* a track, different from $\mathbf{a}_i$, related to the same verb $\mathbf{b}_i$) for each valid verb-track pair, we pick them out from the same video clip of the track $\mathbf{a}_i$ and the verb $\mathbf{b}_i$. If the video clip does not contain $\mathbf{b}_i^-$ or $\mathbf{a}_i^+$, the missing elements are chosen in video clips of the same movie, if possible, otherwise they are randomly chosen between any video clip of the dataset.

Finally, at validation and test time, we compute the distances between verbs and tracks of each video clip and we find the best verb-track association by applying the Kuhn-Munkres algorithm on the distance matrix.

Face recognition

In order to fulfill the replacement of the "someone" tags, every track that has been joined to a verb has to be associated to a movie character. Therefore, we convert each track to a 128-dimensional embedded representation by using a deep neural model inspired by FaceNet [164] and trained on a sub-set of the MS-Celeb-1M dataset [59]. Then, we classify each embedding using a kd-tree, an optimized version of the K-Nearest Neighbours classifier, fitted on the character embedded representations of the training set. The K-NN classifier has the advantage of being particularly flexible when considering characters with different visual aspects within the same movie (*i.e.* classes with many clusters lying in different areas of the 128-d embedding space). On the contrary, other classifiers, such as linear models and other types of clustering, are not always capable of correctly classify these cases.

5.2.3 Experimental settings

Along with the proposed loss function, defined in Eq. 5.23, we evaluate the performances of different loss functions. In particular, we test the binary loss function, which is defined as the binary cross entropy on a single label, and the siamese loss function, which is defined in Eq. 5.20. Moreover, we test the two-terms version of the triplet loss function (Eq. 5.22) and a four-terms variation, which is defined as the two-terms version with the addition of the terms $t(\mathbf{a}_i^+, \mathbf{b}_i, \mathbf{b}_i^-)$ and $v(\mathbf{a}_i, \mathbf{b}_i, \mathbf{a}_i^w)$, where $\mathbf{a}_i^+$, $\mathbf{b}_i^-$, and $\mathbf{a}_i^w$ are defined as in Eq. 5.23. In addition, we evaluate the siamese, the triplet, and the proposed loss function by using both the euclidean distance and the cosine similarity.

Implementation details

The multimodal neural network that projects textual and visual features into the same embedding space is composed by two branches, formed by one 128-dimensional fully connected layer (with the ReLU activation function) each. The first branch projects the C3D visual features into the embedding space, while the second one projects the GloVe textual features into the same multimodal space. When evaluating the binary loss function, an additional 128-dimensional fully connected layer, which takes as input the concatenation of the two branches with the ReLU activation function, and a one-dimensional fully connected layer with the sigmoid activation function, which predicts the correspondence or non-correspondence of the verb-track pair, are added to the network.

During the training process, we minimize the loss function by applying the Stochastic Gradient Descent using an initial learning rate set to 0.002 with Nesterov momentum 0.9 and weight decay 0.0005. We use batches composed by 128 random samples. The loss margin α is fixed to 0.2. We ignore the smallest tracks (*i.e.* face tracks with a side lower than 28 pixels) in order to prevent the addition of noise during the training phase.

5.2.4 Experimental results

Table 5.8 shows experimental results in terms of the final accuracy of replacing “someone” tags in existing captions with proper character names. We report the results of the proposed model trained with all the aforementioned loss functions. For the triplet loss (with two and four terms), the siamese version, and the proposed loss function, results are reported using both the euclidean distance and the cosine

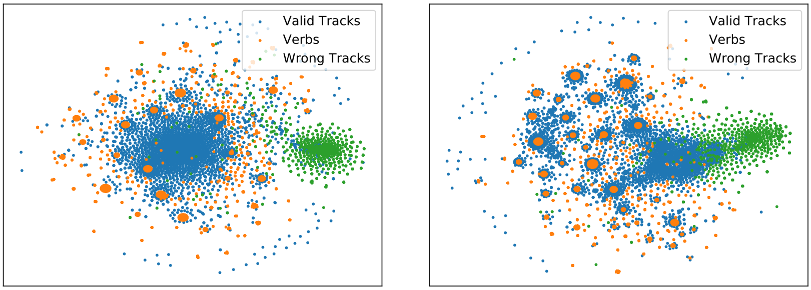
	Val. Acc. (%)	Test Acc. (%)
Random assignment	11.9	11.6
Binary with two terms	48.2	49.9
Triplet Loss with two terms (cosine similarity)	54.3	53.6
Triplet Loss with two terms (euclidean distance)	56.4	58.5
Siamese (cosine similarity)	54.4	54.0
Siamese (euclidean distance)	56.1	59.0
Binary with four terms	49.8	51.0
Triplet Loss with four terms (cosine similarity)	58.7	58.2
Triplet Loss with four terms (euclidean distance)	57.8	**59.1**
Proposed Loss (cosine similarity)	58.7	58.0
Proposed Loss (euclidean distance)	**60.1**	59.0

Table 5.8: Experimental results on the “replacing the someone” task, with different loss functions. Results are reported, in terms of accuracy, on both validation and test splits of the M-VAD Names dataset.

similarity. For reference, we also test the results of a random replacement of any “someone” tag with a character name randomly extracted from the character list of each movie.

As it can be seen, the proposed strategy of considering positive and negative pairs of verbs and tracks as well as the wrong detections is beneficial for the final accuracy. In particular, on the validation set, the model trained with the proposed loss obtains the best performances. On the testing set, instead, the model trained with the triplet loss with four terms is the best performing one, even though by a slight margin.

Figure 5.10 shows a representation of the textual-visual embedding space obtained by training the model with the proposed loss function using both the euclidean distance and the cosine similarity. In particular, we report each verb-track pair of the M-VAD Names validation set along with all wrong visual tracks of the corresponding video clips. To get a suitable two-dimensional representation out of a 128-dimensional space, we run the t-SNE algorithm [126, 191], which iteratively finds a non-linear projection which preserves pairwise distances from the original space. As it can be noticed, the represented embedding spaces are composed by clusters of verb representations that probably correspond to verbs with a similar meaning. Wrong tracks are discriminated quite well in both spaces, while valid tracks are better divided and assigned to a specific verb cluster when using the euclidean distance thus confirming the quantitative results reported in



(a) Proposed Loss (cosine similarity)

(b) Proposed Loss (euclidean distance)

Figure 5.10: Comparison between textual-visual embedding spaces obtained by training the model with the proposed loss function using both euclidean distance and cosine similarity. Visualization is obtained by running the t-SNE algorithm on top of the verb-track embedded representations. Best seen in color.

Table 5.8.

In Figure 5.11, we also report the results in terms of validation accuracy obtained on single movies. In particular, the graph shows the results obtained on the 10 movies with the best accuracy results and those obtained on the 10 movies with the worst ones. These results highlight that correct verb-track matches are more difficult for a specific sub-set of movies. This is probably due to different number of characters or different number of unknown and wrong tracks that could cause greater difficulty in associating a verb with its corresponding visual track.

To validate the face recognition method, in Table 5.9 we report a comparison between different classifiers. In details, we compare the K-Nearest Neighbours (K-NN) classifier (with a k value of 5) with the SVM (applying the Radial Basis Function kernel) and the Adaboost (with 30 Decision Trees) classifier. For each of them, we only use the face tracks of the training set to classify all validation and test samples. As it can be seen, the K-NN performs better on both the validation and the test set.

In order to assess the effectiveness of the proposed sampling strategy, we compare it by sampling the verb-track pairs within the whole dataset. Table 5.10 shows the validation accuracies of the different sampling strategies. Results confirm that the proposed procedure of sampling verb-track pairs within the same video clip, if possible, allows to better discriminate samples from the same movie.

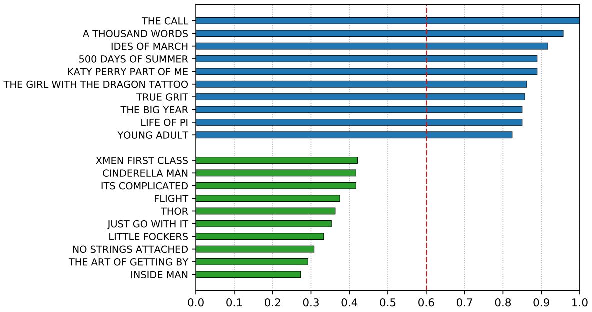


Figure 5.11: Accuracy results on single movies of the proposed approach on the M-VAD Names validation set. We report the 10 best results and 10 worst ones, respectively represented in blue and green. The vertical line represents the averaged accuracy on all movies.

	Val. Acc. (%)	Test Acc. (%)
K-NN	**85.2**	**86.3**
SVM	84.2	85.2
Adaboost	64.7	65.8

Table 5.9: Accuracy of different character face classifiers. Accuracy is calculated as the number of correct predictions on the known-character tracks of the validation and the test set.

5.2.5 Testing on different movies

After assessing the performance of the proposed architecture on the M-VAD Names dataset, in which the training, the validation, and the test split share the same character names, we address a more challenging and realistic evaluation scenario. In this case, we evaluate on movies outside of the M-VAD Names, while using the multimodal embedding space trained on the proposed dataset. To link characters' appearances with their identities, we initialize the face embedding space by randomly sampling 10% of the face tracks. Beyond this limited supervision signal, which is mandatory when new characters are added, we do not exploit any other training data related to the new set of movies.

The set of external videos contains three movies belonging to the MPII-MD dataset for video captioning [160], namely *Harry Potter and the Philosopher's*

	Val. Acc. (%)
Sampling within a video clip	**60.1**
Sampling within the whole dataset	58.8

Table 5.10: Comparison of different sampling strategies on the M-VAD Names validation set using the proposed loss function to train the model.

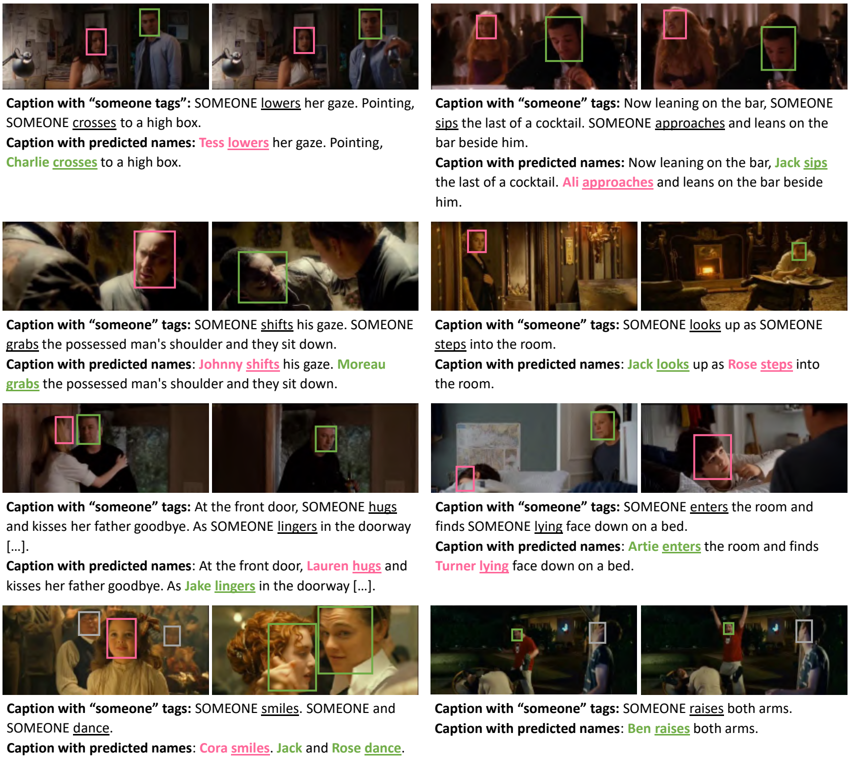
	Face Class. (%)	Replacement (%)
Harry Potter and the Philosopher's Stone	78.6	59.0
Pulp Fiction	81.4	54.6
Sherlock Holmes: A Game of Shadows	82.0	65.6
Overall accuracy	**80.7**	**60.1**

Table 5.11: Face classification and "someone" replacement results on an external set of movies from the MPII-MD dataset [160], using 10% of the tracks for training the face embedding space, and the multimodal embedding space model pre-trained on the M-VAD Names dataset.

Stone, *Pulp Fiction* and *Sherlock Holmes: A Game of Shadows*. Table 5.11 shows the accuracy results for both the face classification and the "someone" replacement task. Numbers are reported on the portion of tracks which were not used for the initialization of the face embedding space. The overall accuracy is reported by averaging on every considered track (notice that each movie has a different number of tracks). As it can be noticed, the accuracy of the replacement task is similar to the one obtained when testing on the M-VAD Names, thus confirming the generalization capabilities of the proposed model.

5.2.6 Qualitative results

Figure 5.12 shows some qualitative results on sample clips from the M-VAD Names validation set. For each movie clip, we report the original textual description with "someone" tags and that with the corresponding character names predicted by our approach. As it can be seen, our model is able to discriminate tracks containing different actions and to associate them with the corresponding verb in the captions. Also, visual tracks of unknown characters or character tracks that are not associated to a verb in the caption are correctly not paired to any verb, as for example in the fourth row of the figure.



Caption with "someone tags": SOMEONE lowers her gaze. Pointing, SOMEONE crosses to a high box.
Caption with predicted names: Tess lowers her gaze. Pointing, Charlie crosses to a high box.

Caption with "someone" tags: Now leaning on the bar, SOMEONE sips the last of a cocktail. SOMEONE approaches and leans on the bar beside him.
Caption with predicted names: Now leaning on the bar, Jack sips the last of a cocktail. Ali approaches and leans on the bar beside him.

Caption with "someone" tags: SOMEONE shifts his gaze. SOMEONE grabs the possessed man's shoulder and they sit down.
Caption with predicted names: Johnny shifts his gaze. Moreau grabs the possessed man's shoulder and they sit down.

Caption with "someone" tags: SOMEONE looks up as SOMEONE steps into the room.
Caption with predicted names: Jack looks up as Rose steps into the room.

Caption with "someone" tags: At the front door, SOMEONE hugs and kisses her father goodbye. As SOMEONE lingers in the doorway [...].
Caption with predicted names: At the front door, Lauren hugs and kisses her father goodbye. As Jake lingers in the doorway [...].

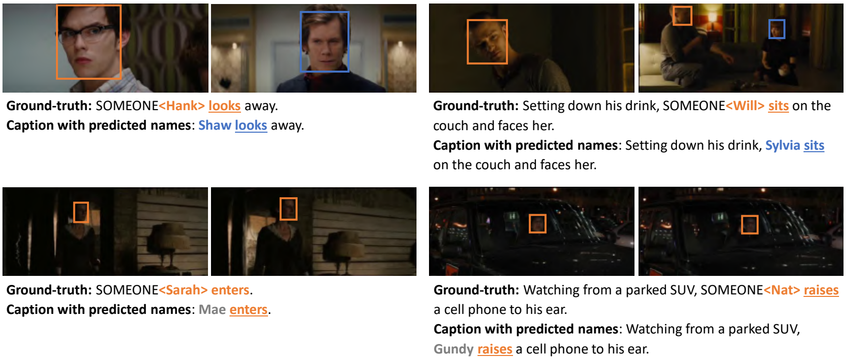
Caption with "someone" tags: SOMEONE enters the room and finds SOMEONE lying face down on a bed.
Caption with predicted names: Artie enters the room and finds Turner lying face down on a bed.

Caption with "someone" tags: SOMEONE smiles. SOMEONE and SOMEONE dance.
Caption with predicted names: Cora smiles. Jack and Rose dance.

Caption with "someone" tags: SOMEONE raises both arms.
Caption with predicted names: Ben raises both arms.

Figure 5.12: Sample results of the proposed method for replacing "someone" tags with character proper names. For each sample, tracks associated with the same verb are represented with the same color, while tracks that are not associated with any verb are reported in gray.

Finally, we report some failure cases in Figure 5.13. In particular, the figure shows two verb-track association errors (first row), and two cases in which the error is due to the face recognition phase (second row). In the first case, the verb in the caption is associated with a different visual track of the considered movie clip that is then correctly classified with the corresponding character name. In the other one, instead, the visual track is correctly associated with the corresponding verb in the caption, but the face recognition algorithm fails to identify the correct



Ground-truth: SOMEONE<Hank> looks away.
Caption with predicted names: Shaw looks away.

Ground-truth: Setting down his drink, SOMEONE<Will> sits on the couch and faces her.
Caption with predicted names: Setting down his drink, Sylvia sits on the couch and faces her.

Ground-truth: SOMEONE<Sarah> enters.
Caption with predicted names: Mae enters.

Ground-truth: Watching from a parked SUV, SOMEONE<Nat> raises a cell phone to his ear.
Caption with predicted names: Watching from a parked SUV, Gundy raises a cell phone to his ear.

Figure 5.13: Failure cases from the M-VAD validation set. In the first row, two track-verb association errors, while in the second row, two face recognition errors.

character appearing in the movie clip.

Chapter 6

Cross-modal retrieval

As seen in the previous chapters, recent advancements in computer vision and natural language processing have made it possible the development of neural networks capable of bridging the gap between vision and language resulting in new solutions not only for image and video captioning, but also for cross-modal retrieval [94, 45], visual question answering [215, 4, 166], and vision-and-language navigation [5, 211, 46]. In this chapter, we focus on cross-modal retrieval and on the development of deeply-learnable architectures which can retrieve visual items given textual queries and vice versa.

Contributions

The key idea of many cross-modal retrieval approaches has been that of learning a joint multimodal embedding space in which text and images could be projected and compared. In the first part of this chapter, we take a different approach and reformulate the problem of text-image retrieval as that of learning a translation between the textual and visual domain. Our proposal leverages an end-to-end trainable architecture that can translate text into image features and vice versa, and regularizes this mapping with a cycle-consistency criterion. Experimental evaluations for text-to-image and image-to-text retrieval, conducted on small, medium and large-scale datasets show consistent improvements over the baselines,

This chapter is related to publications [12, 13, 14, 21, 23] reported in Appendix B, by the author of the thesis. See Appendix B for details.

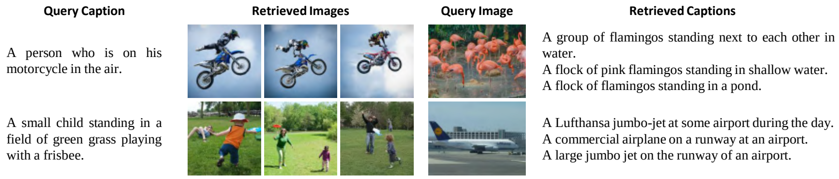
thus confirming the appropriateness of using a cycle-consistent constrain for the text-image matching task.

While these solutions has shown impressive performances on fully-supervised settings in which a large amount of training data is available, their application to more challenging scenarios has been rarely investigated. In the second part of this chapter, we go beyond these limitations and tackle the design of visual-semantic algorithms in the domain of the digital humanities and cultural heritage. This setting not only advertises more complex visual and semantic structures but also features a significant lack of training data which makes the use of fully-supervised approaches infeasible. With this aim, we propose cross-modal retrieval solutions that can automatically align illustrations and textual elements without paired supervision transferring the knowledge learned on ordinary visual-semantic datasets to the artistic domain. Experiments, performed on two datasets specifically designed for this domain, validate the proposed strategies and quantify the domain shift between natural images and artworks.

6.1 Toward cycle-consistent approaches

As mentioned, text-image cross retrieval is one of the core challenges in computer vision and multimedia. The task concerns the retrieval of visual items given textual queries and vice versa, and can be casted as a ranking problem, for which the correct item should be closer to the query than to any other element in the dataset (Fig. 6.1). Since visual and textual data belong to two distinct modalities, previous methods have often relied on the construction of a common multimodal embedding space [205, 45, 94, 138], with learnable functions to project data from the two modalities in the joint embedding. Retrieval, in this case, is then carried out by measuring distances in the joint space, which should be low for matching text-image pairs and large for non-matching pairs.

Despite approaches based on a common visual-semantic embedding have led to state-of-the-art results, in this section we investigate a different approach. Specifically, we take the problem of retrieving images and captions as a *translation* from the image domain to the textual domain and vice versa. In the first direction, an image, represented with a feature vector, is converted to a textual representation of its content; in the latter direction, a sentence is converted into an image feature which reflects its meaning. The concept is visually described in Fig. 6.2. In (a), we show a traditional visual semantic model in which textual items (squares) and visual features (circles) are projected in a common embedding space, by requiring



A person who is on his motorcycle in the air.

A small child standing in a field of green grass playing with a frisbee.

A group of flamingos standing next to each other in water.
A flock of pink flamingos standing in shallow water.
A flock of flamingos standing in a pond.

A Lufthansa jumbo-jet at some airport during the day.
A commercial airplane on a runway at an airport.
A large jumbo jet on the runway of an airport.

(a) Text-to-image retrieval

(b) Image-to-text retrieval

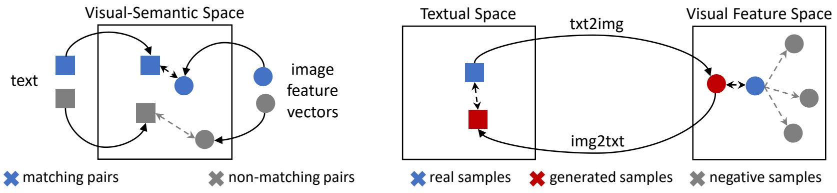
Figure 6.1: Text-image retrieval examples.

matching pairs (depicted in blue) to lie closer than non-matching pairs (depicted in gray). In (b) we instead outline our approach: textual items (squares) can be translated to visual features (circles) by means of the “txt2img” translation function, while the “img2txt” function translates visual features back into textual items. The overall architecture is trained end-to-end by combining two objectives: generated visual features (depicted in red) are required to be realistic with respect to positive and negative image samples (depicted, respectively, in blue and gray); further, a cyclic constraint is imposed to guarantee that the forward and backward translations are consistent.

In the remainder of the section, we show how this cycling approach acts as a good regularizer for the retrieval task in both image-to-text and text-to-image scenarios. Furthermore, we demonstrate how our proposed strategy is particularly beneficial in the case of small and medium-scale datasets.

6.1.1 A cycle-consistent text and image retrieval network

We propose a Cycle-consistent Text and Image Retrieval network, `CyTIR-Net`, which works in an end-to-end manner. `CyTIR-Net` is a combination of recurrent and convolutional architectures that operates a translation between the textual and the visual domains, where the latter is parametrized as the space of the features extracted from a Convolutional Neural Network. Under the model, input captions can be translated to proper image features, and image vectors can be translated back to the textual domain. Exploiting this translation capability, a reconstruction constraint makes sure that the reconstructed text is similar to the original text, *i.e.* that the generated image vectors are meaningful when compared to real image vectors, and contain sufficient information to reconstruct their captions. Experimentally, this will be shown to be a good regularizer to enhance



(a) Ordinary visual-semantic models (b) Our approach

Figure 6.2: Overview of the proposed approach in comparison with ordinary visual-semantic models. Instead of relying on a joint embedding space, we cast the problem of text-image retrieval as that of learning a translation between the textual and visual domain, with a reconstruction objective that keeps the overall process cycle-consistent. Best seen in colors.

the discriminative power of the model in a cross-modal retrieval scenario.

Fig. 6.3 provides a high-level description of the model. We discuss the implementation details in the following subsections.

From text to images

The first part of the architecture consists of a visual-semantic model that transforms a sentence s into a meaningful vector in the image feature space, $\tilde{x}$. Words are represented with one-hot vectors that are embedded with a linear embedding E, which can be either learned end-to-end together with the model or pre-trained using another word embedding model, like Word2Vec [135], GloVe [147] or FastText [13]. The beginning of each sentence is marked with a `BOS` token and the end with a `EOS` token. Under the model, words are consumed by a Gated Recurrent Unit (GRU) layer [29]. The following updates of the hidden units and cells of a GRU define the model:

$$\begin{aligned}
\mathbf{r}_\tau &= \sigma(\mathbf{W}_{ir}\mathbf{e}_\tau + \mathbf{W}_{hr}\mathbf{h}_{\tau-1} + \mathbf{b}_h) \\
\mathbf{z}_\tau &= \sigma(\mathbf{W}_{iz}\mathbf{e}_\tau + \mathbf{W}_{hz}\mathbf{h}_{\tau-1} + \mathbf{b}_z) \\
\mathbf{n}_\tau &= \tanh(\mathbf{W}_{in}\mathbf{e}_\tau + \mathbf{r}_\tau(\mathbf{W}_{hn}\mathbf{h}_{\tau-1} + \mathbf{b}_{hn}) + \mathbf{b}_{in}) \\
\mathbf{h}_\tau &= (1 - \mathbf{z}_\tau)\mathbf{n}_\tau + \mathbf{z}_\tau\mathbf{h}_{\tau-1}, \qquad (6.1)
\end{aligned}$$

where $\mathbf{e}_\tau$ represents embedded words, *i.e.* $\mathbf{e}_\tau = E\mathbf{w}_\tau$ being $\mathbf{w}_\tau$ the one-hot representation of a word, and σ is the sigmoid function. Once the model has

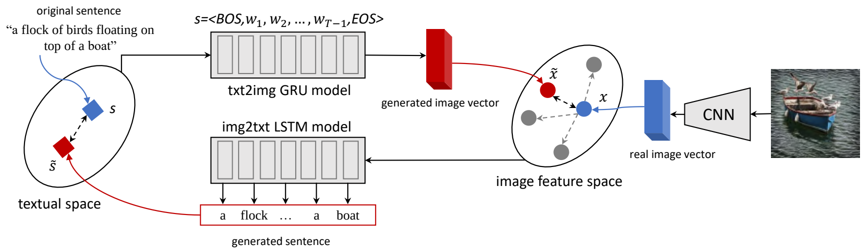


Figure 6.3: Architecture of the `CyTIR-Net`. Two learnable models translate between the textual and the visual domains, by generating image vectors from sentences and sentences from image vectors. A cycle-consistent reconstruction term regularizes the model by imposing the feasibility of the backward translation.

consumed the `EOS` token, we take the hidden state as the projection of the sentence into the image space.

$$\tilde{x} = \texttt{txt2img}(s) = \mathbf{h}_T. \tag{6.2}$$

We call this the *generated image vector*. We learn the set of parameters of this model, θ, with a cost function which encourages the generated image vector to be close to that of an image that has been described by the same caption. This, in general, forces the generated image vectors to be consistent with the structure of the space of real images. To this aim, we define a similarity function inside the image feature space, ξ (*e.g.* the cosine similarity) and apply a hinge-based triplet ranking loss with maximum violation [214, 45]. For a matching image-sentence pair, the loss is defined as follows:

$$\begin{aligned} L_{t2i}(\theta) = \max_{\hat{x}} & \left[\alpha - \xi(x, \texttt{txt2img}(s)) + \xi(\hat{x}, \texttt{txt2img}(s))\right]_+ \\ & + \max_{\hat{s}} \left[\alpha - \xi(x, \texttt{txt2img}(s)) + \xi(x, \texttt{txt2img}(\hat{s}))\right]_+ . \end{aligned} \tag{6.3}$$

This hinge loss comprises two symmetric terms. Given a real image feature vector x which matches with sentence s, the first maximum is taken over all real negative image vectors $\hat{x}$ which do not match with s. The second, instead, is computed over all negative sentences $\hat{s}$ which do not match with x. If the generated image vector $\texttt{text2img}(s)$ is closer to the real image vector x than to any negative image vector by a margin α, then the first hinge loss is zero. Conversely, if the real image vector x is closer to the generated image vector

$\texttt{txt2img}(s)$ than to any vector generated from a negative image by a margin α, then the second hinge loss is zero.

In practice, the maximums in Eq. 6.3 are computed only inside the mini-batch, to avoid the costly operation of computing the `txt2img` vector for each caption in the dataset. When samples in the mini-batch are chosen randomly, if the dataset shuffling is repeated at each epoch and the mini-batch size is sufficiently large, this procedure leads to a good approximation of Eq. 6.3 and good learning behavior.

We notice, further, that this loss can be implemented quite efficiently. Once real and generated image vectors are ℓ_2-normalized, the pairwise distance matrix can be computed with a matrix product, recalling that the cosine similarity is equivalent to a dot product for ℓ_2-normalized vectors. If the two tensors are sorted so that corresponding real and generated image vectors are in the same position, the main diagonal of the matrix contains $\xi(x, \texttt{txt2img}(s))$, so we can obtain the first (second) term of Eq. 6.3 by repeating the diagonal vector horizontally (vertically), subtracting it from the pairwise similarity matrix and adding the margin. Once the diagonal of the resulting matrix has been cleared[1], we can take the maximum over columns (rows). The final loss, which is fed to the optimizer, is the mean of $L_{t2i}(\theta)$ over the mini-batch.

The `txt2img` model, alone, is equivalent to a visual-semantic embedding model in which the image projection function is the identity, *i.e.* a model in which the common embedding space is collapsed into the image space. By showing that our complete architecture overcomes the performance of models with a multimodal embedding space, we implicitly show that a common space is not always needed to address text-image retrieval.

From images to text

While sentences can be projected into an image feature space, the second component of the model translates image vectors x into the textual space by generating a textual description $\tilde{s}$. This roughly corresponds to an image captioning model: the image is treated as the first input of an LSTM-based recurrent model [68], which generates a sentence one word at a time by updating his hidden units and cells

[1] Each element of the diagonal would contain $\alpha - \xi(x, \texttt{txt2img}(s)) + \xi(x, \texttt{txt2img}(s)) = \alpha$, thus potentially invalidating the result of the maximum.

under the following equations.

$$
\begin{aligned}
\mathbf{i}_\tau &= \sigma(\mathbf{W}_{ii}\mathbf{e}_\tau + \mathbf{W}_{hi}\mathbf{h}_{\tau-1} + \mathbf{b}_i)\\
\mathbf{f}_\tau &= \sigma(\mathbf{W}_{if}\mathbf{e}_\tau + \mathbf{W}_{hf}\mathbf{h}_{\tau-1} + \mathbf{b}_f)\\
\mathbf{g}_\tau &= \tanh(\mathbf{W}_{ig}\mathbf{e}_\tau + \mathbf{W}_{hg}\mathbf{h}_{\tau-1} + \mathbf{b}_g)\\
\mathbf{o}_\tau &= \sigma(\mathbf{W}_{io}\mathbf{e}_\tau + \mathbf{W}_{ho}\mathbf{h}_{\tau-1} + \mathbf{b}_o)\\
\mathbf{c}_\tau &= \mathbf{f}_\tau\mathbf{c}_{\tau-1} + \mathbf{i}_\tau\mathbf{g}_\tau\\
\mathbf{h}_\tau &= \mathbf{o}_\tau \tanh(\mathbf{c}_\tau)\\
w_\tau &\sim softmax(\mathbf{W}_d\mathbf{h}_\tau),
\end{aligned}
\tag{6.4}
$$

where $\mathbf{e}_\tau$ corresponds to the input image feature vector for $\tau = 0$, then the output of the network, w_τ is fed back to the input in the next timesteps as a one-hot vector.

To deal with the fixed-size weights of the LSTM model, we project both the image feature vector and words to a common dimensionality, using a linear projection. Following recent models for captioning [155], we also apply a dropout regularization on the hidden state $\mathbf{h}_\tau$ at each iteration. We noticed that an LSTM cell performed better than a GRU cell in this part of the model.

At each iteration, the hidden state is linearly projected to the dimensionality of the vocabulary through $\mathbf{W}_d$, and a softmax activation is then used to produce a probability distribution over the vocabulary, from which w_τ is sampled. For each input image vector, the model is run until an `EOS` token is produced: the final textual representation of an image i is therefore defined as

$$\tilde{s} = \texttt{img2txt}(x) = [w_0, w_1, ..., w_T]\,, \tag{6.5}$$

which we call the *generated sentence*.

Closing the loop

The `txt2img` and `img2txt` models defined above realize the forward and backward translations between the image and the textual domain. The mapping between the two spaces is regularized with a cycle-consistency criterion, in which we require the feasibility of the forward and backward translation at the same time. In practice, we require that the projection of a generated image vector into the textual space should be similar to the text from which the vector originated, *i.e.*

$$\texttt{img2txt}(\texttt{txt2img}(s)) \approx s. \tag{6.6}$$

We realize Eq. 6.6 by computing the negative log-likelihood of generated words with respect to the words in s.

Formally, given the set of words in s, $\{w_0^*, w_1^*, ..., w_T^*\}$, the learning objective at each step is defined by feeding the previous words in s to the model, and computing the negative log of the probability of generating the correct word:

$$L_{i2t}(\theta) = -\sum_{\tau=0}^{T} \log(p(w_\tau | w_0^*, w_1^*, ..., w_{\tau-1}^*)). \tag{6.7}$$

The final loss function of the model for a training sample is a weighted sum of the L_{t2i} and the L_{i2t} losses. We average the losses computed over a mini-batch before updating the weights of the model. Notice, further, that the L_{t2i} loss backpropagates only on the `txt2img` model, while the reconstruction loss backpropagates on the overall architecture, thus potentially changing the representation provided by the `txt2img` model as well. When fine-tuning image feature vectors, moreover, both models contribute to the modification of the CNN weights, building an image representation that is both discriminative in the image feature space, and suitable for text reconstruction.

At evaluation time, captions are projected into the image feature space by means of the `txt2img` model and similarities are computed by means of the cosine distance. The image feature space is thus employed as a replacement of the traditional multimodal embedding space, often used by competitor models [205, 138].

6.1.2 Experimental settings

In this section, we provide all implementation details and describe the datasets used in our experiments.

Datasets. To test the behavior of our architecture on both medium and large-scale datasets, we employ Flickr8k [69], Flickr30k [229] and Microsoft COCO [116]. As already mentioned in the previous chapters, Flickr8k and Flickr30k are respectively composed of $8,000$ and $31,000$ images, where each image comes with 5 captions. Following the splits defined in [90], for Flickr8k and Flickr30k we use $1,000$ images for validation, $1,000$ images for testing and the rest for training. COCO instead contains more than 120,000 images, each of them annotated with 5 human-collected captions. Also for this dataset, we use the splits defined in [90], where the training set is composed of $82,783$ images, while both validation and test set contain $5,000$ images. However, there are also $30,504$ images that were originally in the validation set of this dataset but have been left out in this split. As done by other text-image retrieval methods [45], we add these images in the training set thus obtaining a total of $113,287$ images to train the model. Following a common practice, retrieval results on COCO are reported by averaging over 5 folds of $1,000$ test images each.

Metrics. For evaluating the results of our `CyTIR-Net` model and for comparing it with different baselines and state-of-the-art methods, we report rank-based performance metrics $R@K$ ($K = 1, 5, 10$) and median rank ($Med\,r$). In particular, $R@K$ computes the percentage of test images or test sentences for which at least one correct result is found among the top-K retrieved sentences, in the case of text retrieval, or the top-K retrieved images, in the case of image retrieval. On the contrary, $Med\,r$ is the median rank of the first correct result in the ranking.

Implementation details. To encode input images and define the image feature space, we extract global feature vectors from the activations of CNNs pre-trained on large-scale image classification, which is a dominant approach employed by many previous works [45, 41, 58, 77]. Image classification is indeed a good proxy task for retrieval, and CNNs trained for classification tend to learn global features which are general and discriminative enough to build a suitable feature space for our task. Specifically, we use and compare four networks, namely VGG-16 [168], VGG-19 [168], ResNet-50 [64], and ResNet-152 [64] pre-trained on ImageNet [163]. In all cases, we extract features from the last but one layer before the final classification stage, so that features can be general enough to adapt to unseen classes. On VGG-16 and VGG-19, we extract the activation from the last but one fully connected layer (which is usually denoted as `fc7`), while on ResNet-50 and ResNet-152 we take the output of the final average pooling layer. This leads to a dimensionality of the final feature vector of 4096 for VGG networks and 2048 for ResNet models. Images are pre-processed to adapt to the training conditions of the CNNs. Specifically, are firstly resized to 256 pixels for the smaller edge, and then randomly cropped to a size of 224 × 224 during training, and center cropped during test.

For encoding image captions, instead, we use a GRU network, as described in Sec. 6.1.1. Since we do not project images and corresponding captions in a joint embedding space, we need to set the output dimensionality of the GRU to the size of image embeddings (*i.e.* 4096 or 2048, depending on the CNN at hand). The dimensionality of the word embeddings that are input to the GRU is set to 300, even when they are learned end-to-end.

As mentioned, for the img2txt model, we instead use an LSTM network. We set the dimensionality of both LSTM hidden state and input word embeddings to 300, and we apply a dropout regularization on the hidden state with a probability of 0.5 at each iteration.

All experiments have been performed using the Adam optimizer [93]. Due to different learning capabilities of the two components of our model, we use two different learning rates. For the parameters of the txt2img model, we use an initial learning rate of 2×10^{-4}, which is then decreased by a factor of 10 every 50 epochs. For the img2txt model, instead, we set the initial learning to 4×10^{-4}, and multiply it by a factor of 0.8 every 5 epochs. The fine-tuned models are trained with a learning rate of 2×10^{-5} by taking a model trained for 25 epochs with a fixed image encoder. For all experiments, we use a mini-batch of 128 samples and we set the margin α to 0.2.

6.1.3 Comparison with baselines

As a baseline for CyTIR-Net, we consider the txt2img model, which removes the cycle-consistency regularizer and is therefore well suited to evaluate the claims of the proposal regarding the role of the cycle-consistent constraint. This, also, is practically equivalent to a visual-semantic embedding model in which the visual projector is the identity function. In the following, we evaluate the model with respect to this baseline and using different image feature vectors and word embeddings.

Comparison between different image feature vectors. We first evaluate the performance of our model by comparing image feature vectors extracted from different CNNs, namely VGG-16 [168], VGG-19 [168], ResNet-50 [64], and ResNet-152 [64]. Table 6.1 reports the performance of the CyTIR-Net model in comparison to that of the txt2img model alone. For the sake of conducting a good variety of experiments, here we rely on the Flickr8k and Flickr30k datasets and do not fine-tune the CNN model. All results are obtained by using learnable word embeddings.

As it can be seen, CyTIR-Net outperforms the baseline in both text and image retrieval and also by using different image feature vectors, thus demonstrating the effectiveness of our proposal. Moreover, the performance of the ResNet-152 model is superior to that of other CNNs on both considered datasets. For this reason, all following experiments are performed by using the ResNet-152 as image feature vector.

Model	**CNN**	Text Retrieval				Image Retrieval			
		R@1	**R@10**	**R@10**	**Med r**	**R@1**	**R@5**	**R@10**	**Med r**
				Flickr8k					
`txt2img`	VGG-16	22.6	51.5	64.0	5.0	16.7	42.0	54.2	9.0
`CyTIR-Net`	VGG-16	**24.9**	**54.2**	**65.6**	**4.0**	**17.5**	**42.7**	**55.9**	**8.0**
`txt2img`	VGG-19	23.5	**53.3**	**65.6**	**5.0**	16.4	41.2	53.9	9.0
`CyTIR-Net`	VGG-19	**24.7**	52.2	65.2	**5.0**	**17.9**	**42.9**	**55.9**	**8.0**
`txt2img`	ResNet-50	27.3	53.5	66.9	5.0	15.1	41.1	55.5	8.0
`CyTIR-Net`	ResNet-50	**28.7**	**56.9**	**69.5**	**4.0**	**16.0**	**42.8**	**58.0**	**7.0**
`txt2img`	ResNet-152	25.7	54.8	69.0	**4.0**	15.8	41.6	56.0	8.0
`CyTIR-Net`	ResNet-152	**28.2**	**57.4**	**71.1**	**4.0**	**17.5**	**44.6**	**59.0**	**7.0**
				Flickr30k					
`txt2img`	VGG-16	29.9	59.1	72.5	4.0	21.5	47.0	58.9	**6.0**
`CyTIR-Net`	VGG-16	**30.9**	**61.0**	**72.7**	**3.0**	**21.9**	**47.9**	**60.1**	**6.0**
`txt2img`	VGG-19	31.0	58.3	69.2	4.0	21.3	46.8	59.0	7.0
`CyTIR-Net`	VGG-19	**32.9**	**61.6**	**71.4**	**3.0**	**22.2**	**47.9**	**59.8**	**6.0**
`txt2img`	ResNet-50	34.0	64.4	75.8	**3.0**	**21.1**	46.9	59.2	**6.0**
`CyTIR-Net`	ResNet-50	**36.6**	**67.1**	**77.3**	**3.0**	20.9	**47.1**	**60.9**	**6.0**
`txt2img`	ResNet-152	36.9	67.0	78.2	3.0	22.8	50.0	63.3	**5.0**
`CyTIR-Net`	ResNet-152	**41.7**	**68.9**	**78.9**	**2.0**	**23.8**	**51.3**	**64.0**	**5.0**

Table 6.1: Comparison between image feature vectors extracted from different CNNs on Flickr8k and Flickr30k validation set.

Comparison between different word embeddings. We learn linear word embeddings from scratch while training the model, and test with a variety of state-of-the-art pre-computed embeddings, namely Word2Vec [135], GloVe [147] or FastText [13]. In the latter case, the embedding matrix E is initialized using pre-computed embeddings and freezed during learning.

Table 6.2 reports the performance of the `CyTIR-Net` model on validation sets with different word embedding strategies, together with that of the `txt2img` model alone. Also in this case, we report experiments on the Flickr8k and Flickr30k datasets and we do not fine-tune the CNN.

Firstly, we observe that the performance of the complete model is always superior to that of the baseline, thus confirming the importance of translating backwards to the textual space. Using learnable word embeddings, this accounts for a $\frac{28.2-25.7}{25.7} = 9.73\%$ relative improvement on R@1 on Flickr8k and a 13.01%

Model	Word Emb.	Text Retrieval				Image Retrieval			
		R@1	**R@10**	**R@10**	**Med r**	**R@1**	**R@5**	**R@10**	**Med r**
Flickr8k									
txt2img	-	25.7	54.8	69.0	**4.0**	15.8	41.6	56.0	8.0
CyTIR-Net	-	**28.2**	**57.4**	**71.1**	**4.0**	**17.5**	**44.6**	**59.0**	**7.0**
txt2img	GloVe	29.2	60.2	74.5	**3.0**	19.2	46.7	61.7	**6.0**
CyTIR-Net	GloVe	**32.2**	**62.7**	**76.2**	**3.0**	**19.9**	**48.8**	**62.8**	**6.0**
txt2img	FastText	29.8	58.7	73.4	4.0	17.9	45.8	60.3	7.0
CyTIR-Net	FastText	**32.2**	**61.4**	**74.1**	**3.0**	**19.2**	**47.5**	**62.0**	**6.0**
txt2img	Word2Vec	28.1	58.0	71.3	**4.0**	17.1	44.1	58.7	7.0
CyTIR-Net	Word2Vec	**30.9**	**59.4**	**72.7**	**4.0**	**18.9**	**46.8**	**61.2**	**6.0**
Flickr30k									
txt2img	-	36.9	67.0	78.2	3.0	22.8	50.0	63.3	**5.0**
CyTIR-Net	-	**41.7**	**68.9**	**78.9**	**2.0**	**23.8**	**51.3**	**64.0**	**5.0**
txt2img	GloVe	36.4	67.4	78.4	**2.0**	22.8	50.7	64.2	**5.0**
CyTIR-Net	GloVe	**41.1**	**68.9**	**79.0**	**2.0**	**23.0**	**51.3**	**64.6**	**5.0**
txt2img	FastText	37.7	66.0	77.8	3.0	22.1	49.8	63.4	6.0
CyTIR-Net	FastText	**40.8**	**68.5**	**79.1**	**2.0**	**23.5**	**51.3**	**63.8**	**5.0**
txt2img	Word2Vec	35.9	66.4	76.9	3.0	**22.3**	49.7	62.9	6.0
CyTIR-Net	Word2Vec	**41.2**	**68.2**	**79.3**	**2.0**	**22.3**	**50.7**	**63.7**	**5.0**

Table 6.2: Comparison between different word embeddings on Flickr8k and Flickr30k validation set.

relative improvement on Flickr30k, when doing text retrieval. When doing caption retrieval, the relative improvement is 10.76% on Flickr8k and 4.39% on Flickr30k, thus suggesting that the cycle consistency constraint helps on both retrieval directions, being however slightly better when retrieving text from images.

Comparing the results obtained with different word embeddings, then, it can be seen that GloVe vectors outperform all other solutions by a considerable margin on Flickr8k. The same behavior, however, is not observable in the case of Flickr30k, where GloVe still outperform other pre-computed solutions, but learning the embeddings end-to-end provides the best performance. We assume that this is caused by the different size of the two datasets: on relatively small datasets like Flickr8k the benefit of a pre-trained and generic word space is more significant than on larger datasets, where a specific and more suited word embedding can be learned. Following this claim, in the rest of the experiments we will employ GloVe

Model	Text Retrieval				Image Retrieval			
	R@1	**R@5**	**R@10**	**Med r**	**R@1**	**R@5**	**R@10**	**Med r**
CCA [95]	31.0	59.3	73.7	4.0	21.3	50.0	64.6	5.0
Word2VisualVec [39]	33.6	-	75.3	3.0	-	-	-	-
2WayNet [40]	43.4	63.2	-	-	29.3	49.7	-	-
`txt2img`	28.8	**59.5**	71.1	**4.0**	17.1	44.5	59.3	7.0
`CyTIR-Net`	**30.0**	58.6	**71.3**	**4.0**	**18.6**	**46.2**	**61.2**	**6.0**
`txt2img` with HN	35.5	64.2	77.0	**3.0**	21.9	51.9	66.2	**5.0**
`CyTIR-Net` with HN	**36.1**	**66.2**	**78.3**	**3.0**	**22.6**	**52.9**	**66.4**	**5.0**
`txt2img` (CNN fine-tuned)	34.4	63.7	77.7	3.0	24.0	54.9	69.5	5.0
`CyTIR-Net` (CNN fine-tuned)	**37.9**	**68.3**	**78.8**	**2.0**	**27.4**	**57.8**	**71.5**	**4.0**
`txt2img` with HN (CNN fine-tuned)	43.7	**74.4**	83.2	**2.0**	30.5	**62.0**	74.4	**3.0**
`CyTIR-Net` with HN (CNN fine-tuned)	**44.9**	74.0	**83.9**	**2.0**	**31.2**	**62.0**	**75.2**	**3.0**

Table 6.3: Results on Flickr8k test set. Best results are underlined for each metric.

vectors on Flickr8k, and train our own word embeddings on Flickr30k and COCO.

6.1.4 Comparison with state of the art

Here we report experimental results on the test sets of the aforementioned datasets comparing our model with the baseline and related works.

For a comprehensive analysis, we report the numbers obtained by the `txt2img` baseline, and those obtained with and without the maximum violation in Eq. 6.3. The suffix `HN` indicates experiments using the maximum violation, while the absence of the same suffix indicates that the experiment has been run by replacing the maximums of Eq. 6.3 with sums (*i.e.*, using the traditional hinge-based ranking loss). We also indicate, in parenthesis, whether the CNN has been fine-tuned.

Tables 6.3 and 6.4 show the results on the Flickr8k and Flickr30k, respectively. Results demonstrate that using the cycle-consistency constraint helps to increase the retrieval performance in almost all combinations, confirming our intuition of translating back to the textual domain to improve the text-image retrieval performance. In Table 6.5, we instead report the same experiments when training our model on COCO. Also in this case, the use of the cycle-consistency criterion has been demonstrated to be beneficial for the final performance of the model either by using maximum violation or by fine-tuning the CNN. Furthermore, by comparing the performances shown on Flickr8k, Flickr30k and COCO, it can easily be seen that the regularization power of the proposed cycle-consistent approach is more evident in the case of small and medium-sized datasets. We

Model	Text Retrieval				Image Retrieval			
	R@1	**R@5**	**R@10**	**Med r**	**R@1**	**R@5**	**R@10**	**Med r**
CCA [95]	33.3	62.0	74.7	3.0	25.6	53.2	66.8	5.0
Word2VisualVec [39]	39.7	-	76.7	2.0	-	-	-	-
2WayNet [40]	49.8	67.5	-	-	36.0	55.6	-	-
sm-LSTM [76]	42.5	71.9	81.5	2.0	30.2	60.4	72.3	3.0
Embedding Network [205]	43.2	71.6	79.8	-	31.7	61.3	72.4	-
VSE++ [45]	52.9	-	87.2	1.0	39.6	-	79.5	2.0
DAN [138]	55.0	81.8	89.0	1.0	39.4	69.2	79.1	2.0
`txt2img`	33.9	64.9	76.6	3.0	21.0	50.4	65.0	**5.0**
`CyTIR-Net`	**36.9**	**67.8**	**79.2**	**2.0**	**21.6**	**51.8**	**65.5**	**5.0**
`txt2img` with HN	44.2	73.7	81.5	**2.0**	**29.2**	58.5	71.2	**4.0**
`CyTIR-Net` with HN	**45.0**	**73.8**	**82.7**	**2.0**	28.9	**59.3**	**71.9**	**4.0**
`txt2img` (CNN fine-tuned)	45.5	74.9	83.5	**2.0**	33.3	63.4	74.6	**3.0**
`CyTIR-Net` (CNN fine-tuned)	**48.1**	**76.3**	**83.9**	**2.0**	**35.3**	**65.7**	**76.5**	**3.0**
`txt2img` with HN (CNN fine-tuned)	54.9	80.8	87.6	**1.0**	40.3	68.2	78.1	**2.0**
`CyTIR-Net` with HN (CNN fine-tuned)	**56.6**	**82.2**	**88.7**	**1.0**	**41.5**	**69.0**	**79.1**	**2.0**

Table 6.4: Results on Flickr30k test set. Best results are underlined for each metric.

underline that this is an important feature of the proposed approach, in that it can enhance the final retrieval performance when training data is scarce, at no additional annotation cost.

6.1.5 Cross-dataset experiments

To investigate the generalization capabilities of our model, we also perform cross-dataset experiments, in which we test the model trained on one dataset on the test set of another dataset. The objective, in this case, is to demonstrate how a model trained on one dataset can generalize to other settings, and to further validate the regularization capabilities of our cycle-consistent constraint on this setting. Table 6.6, in particular, reports the performance obtained by `CyTIR-Net` and `txt2img` trained on Flickr8k, Flickr30k and COCO, and tested on a different domain. As it can be observed, our full model shows better generalization capabilities when compared to the `txt2img` baseline on both image and text retrieval, and when transferring from all datasets to any dataset.

Also, we compare our results with the recent Word2VisualVec approach [39], where authors ran the same kind of experiments using only Flickr8k and Flick30k. We notice that our model is able to achieve a better generalization performance,

Model	Text Retrieval				Image Retrieval			
	R@1	**R@5**	**R@10**	**Med r**	**R@1**	**R@5**	**R@10**	**Med r**
CCA [95]	39.4	67.9	80.9	2.0	25.1	59.8	76.6	4.0
2WayNet [40]	55.8	75.2	-	-	39.7	63.3	-	-
sm-LSTM [76]	53.2	83.1	91.5	1.0	40.7	75.8	87.4	2.0
Embedding Network [205]	54.9	84.0	92.2	-	43.3	76.4	87.5	-
VSE++ [45]	64.6	-	95.7	1.0	52.0	-	92.0	1.0
`txt2img`	44.3	77.0	**88.1**	**2.0**	**29.2**	**64.2**	**79.4**	**3.0**
`CyTIR-Net`	**44.7**	**78.0**	**88.1**	**2.0**	28.9	63.8	**79.4**	**3.0**
`txt2img` with HN	52.1	81.7	**91.2**	1.2	34.5	70.4	83.7	2.6
`CyTIR-Net` with HN	**53.2**	**82.4**	91.1	**1.0**	**34.9**	**71.1**	**84.0**	**2.4**
`txt2img` (CNN fine-tuned)	48.7	81.4	90.7	1.8	40.8	77.6	88.9	**2.0**
`CyTIR-Net` (CNN fine-tuned)	**53.8**	**84.1**	**92.4**	**1.0**	**44.1**	**80.5**	**91.1**	**2.0**
`txt2img` with HN (fine-tuned)	57.6	**87.3**	**94.6**	**1.0**	**48.0**	81.9	**91.4**	**1.8**
`CyTIR-Net` with HN (CNN fine-tuned)	**57.9**	**87.3**	94.2	**1.0**	47.9	**82.2**	91.3	**1.8**

Table 6.5: Results on the MSCOCO 1000-image test set. Best results are underlined for each metric.

with a relative improvement of $\frac{52.5-40.3}{40.3} = 30.27\%$ in terms of R1 when transferring from Flickr30k to Flickr8k, and $\frac{29.6-26.7}{26.7} = 10.86\%$ when transferring in the opposite direction. Most importantly, transferring from Flickr datasets to COCO and vice versa, we still notice that the performance of `CyTIR-Net` surpasses that of `txt2img`, further confirming the effectiveness of our solution.

6.1.6 Text reconstruction results

A key feature of our model is that it can reconstruct an input caption once it has been translated to a visual feature vector. This ability, which is modeled in Eq. 6.6, improves the performances of the model by imposing that the generated visual features should be sufficiently informative for a second model to reconstruct the caption describing the image.

In the previous sections, we have shown how this constraint helps to improve retrieval results. Here, instead, we quantify the reconstruction capability of the model, by computing machine translation metrics between original and generated sentences. In particular, we employ four popular metrics for evaluation: BLEU [144], ROUGE$_L$ [115], METEOR [9], and CIDEr [193].

Table 6.7 reports the text reconstruction results, matching each input caption with the corresponding output of the `img2txt` model. Experiments have been

Model	Text Retrieval			Image Retrieval			Text Retrieval			Image Retrieval		
	R@1	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**
	Flickr8k → Flickr30k						**Flickr30k → Flickr8k**					
Word2VisualVec [39]	26.7	50.7	60.7	-	-	-	40.3	69.3	81.0	-	-	-
txt2img	26.9	50.7	61.5	18.3	39.7	51.3	48.1	76.7	88.5	28.3	60.6	74.7
CyTIR-Net	**29.6**	**52.4**	**63.4**	**20.6**	**42.9**	**53.8**	**52.5**	**81.4**	**89.4**	**30.4**	**64.6**	**77.5**
	Flickr8k → COCO						**COCO → Flickr8k**					
txt2img	13.9	34.0	46.0	8.5	24.9	36.6	26.0	56.2	68.9	**17.5**	**42.4**	55.0
CyTIR-Net	**14.2**	**36.1**	**47.8**	**9.0**	**26.1**	**38.4**	**29.7**	**58.9**	**70.2**	**17.5**	42.3	**55.3**
	Flickr30k → COCO						**COCO → Flickr30k**					
txt2img	23.4	49.5	63.6	12.8	36.0	51.6	28.8	**57.0**	67.4	18.3	42.2	54.6
CyTIR-Net	**24.5**	**51.1**	**64.7**	**13.5**	**37.1**	**52.6**	**29.0**	56.6	**67.6**	**19.2**	**43.3**	**54.8**

Table 6.6: Cross-dataset experiments.

carried out on the test sets of Flickr8k, Flickr30k, and COCO. As can be seen, the CyTIR-Net model is able to reconstruct original captions with high quality, achieving a BLEU score higher than 0.5 in most of the cases. This indicates that the model has learned to exploit the textual meaning of the caption while encoding it into a proper visual feature vector. Further, we recall that our model is still different from a pure autoencoder, as it is trained to produce visual feature vectors that are consistent with real ones, as testified by the retrieval performances discusses in the previous sections.

6.1.7 Complexity and run-time analysis

Compared to ordinary visual-semantic models, our approach introduces an additional component (the LSTM for text reconstruction) and an additional loss function. Therefore, the gain in performance at test time is obtained at the expensive of a slight increase in training times. However, we shall note that our model has the same run-time complexity at test time of any ordinary visual-semantic model. Once the model has been trained, indeed, both visual and textual queries can be projected in the visual feature space, where distances are computed, while the img2txt model is only used at training time as a regularizer. This is equivalent, in terms of computational complexity, to projecting visual and textual elements in a shared multimodal embedding space.

To further analyze the performances of our model, in the following we report mean execution times on a workstation equipped with a NVIDIA GeForce GTX

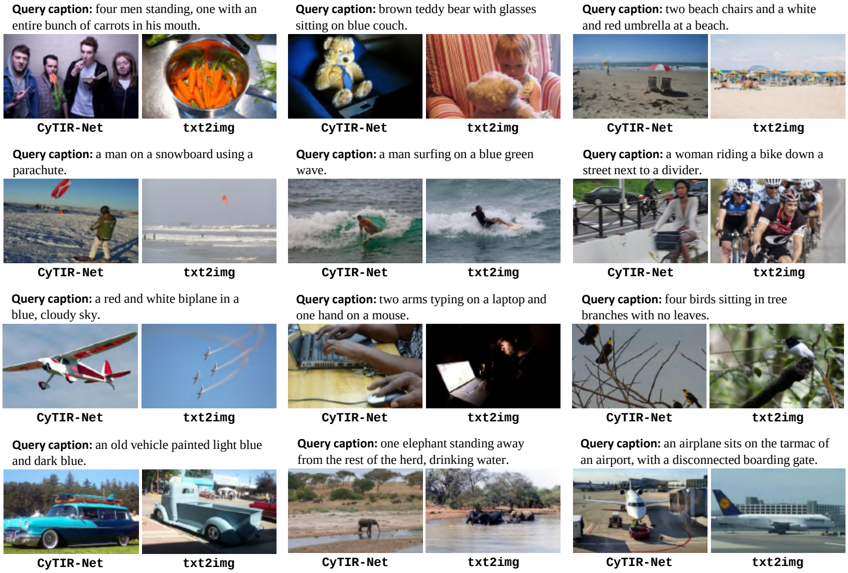
Dataset	B-1	B-2	B-3	B-4	M	R	C
Flickr8k	71.5	56.8	44.7	35.6	29.3	61.8	84.3
Flickr30k	78.2	63.8	53.2	45.4	35.5	70.0	113.3
COCO	95.7	92.0	89.4	87.3	59.4	94.4	237.8

Table 6.7: Text reconstruction results.

1080Ti and Intel Core i7-6850K CPU @ 3.60GHz. Computing the projection of an image query requires on average 0.97 ms when using ResNet-152 as the CNN backbone; computing the projection of a textual query, instead, requires on average 0.21 ms. The computation of pairwise distances, and the ranking of the results demands around 0.52 ms per query. This roughly corresponds to a total execution time of 1.49 ms for a visual query and 0.73 ms for a textual query.

6.1.8 Qualitative results

We also present some qualitative results for text-to-image and image-to-text retrieval, which are shown in Fig. 6.4 and 6.5. Here, we compare against the `txt2img` baseline. As it can be seen, our model is able to encode details of images and text, while the baseline often fails to do the same, ignoring relevant parts of the query caption (such as in the top row of Fig. 6.4) and significant details (second and third rows of Fig. 6.4), or hallucinating the number of people or objects (as in Fig. 6.5). Being the `txt2img` baseline practically equivalent to our model without the cycle-consistent regularization, this further confirms the appropriateness of the `CyTIR-Net` model.



Query caption: four men standing, one with an entire bunch of carrots in his mouth.

Query caption: brown teddy bear with glasses sitting on blue couch.

Query caption: two beach chairs and a white and red umbrella at a beach.

Query caption: a man on a snowboard using a parachute.

Query caption: a man surfing on a blue green wave.

Query caption: a woman riding a bike down a street next to a divider.

Query caption: a red and white biplane in a blue, cloudy sky.

Query caption: two arms typing on a laptop and one hand on a mouse.

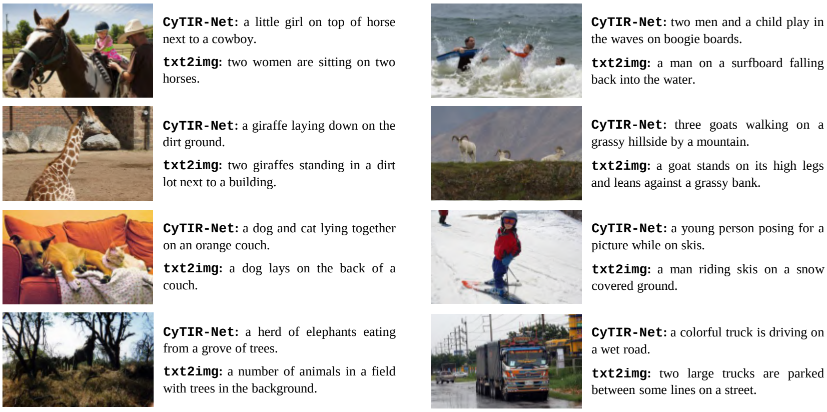
Query caption: four birds sitting in tree branches with no leaves.

Query caption: an old vehicle painted light blue and dark blue.

Query caption: one elephant standing away from the rest of the herd, drinking water.

Query caption: an airplane sits on the tarmac of an airport, with a disconnected boarding gate.

Figure 6.4: Text-to-image retrieval results.



`CyTIR-Net`: a little girl on top of horse next to a cowboy.

`txt2img`: two women are sitting on two horses.

`CyTIR-Net`: two men and a child play in the waves on boogie boards.

`txt2img`: a man on a surfboard falling back into the water.

`CyTIR-Net`: a giraffe laying down on the dirt ground.

`txt2img`: two giraffes standing in a dirt lot next to a building.

`CyTIR-Net`: three goats walking on a grassy hillside by a mountain.

`txt2img`: a goat stands on its high legs and leans against a grassy bank.

`CyTIR-Net`: a dog and cat lying together on an orange couch.

`txt2img`: a dog lays on the back of a couch.

`CyTIR-Net`: a young person posing for a picture while on skis.

`txt2img`: a man riding skis on a snow covered ground.

`CyTIR-Net`: a herd of elephants eating from a grove of trees.

`txt2img`: a number of animals in a field with trees in the background.

`CyTIR-Net`: a colorful truck is driving on a wet road.

`txt2img`: two large trucks are parked between some lines on a street.

Figure 6.5: Image-to-text retrieval results.

6.2 Retrieving images and textual sentences in the artistic domain

In the previous section, we have addressed the problem of cross-modal retrieval on a fully supervised setting in which paired training samples are available and come from general-purpose datasets where the state of the art of concept recognition methods is useful and well assessed. In the domain of arts and culture, however, both visual and textual elements are far from those of ordinary datasets. On one side, textual descriptions often contain technical language with symbolic reminds, metaphors and artistic or historical connections; on the other side, artworks and illustrations are characterized by visual features different from those of natural images. Beyond this domain-shift issue, the supervised training of a common visual-semantic embedding requires sufficiently large datasets. Instead, the artistic domain is often characterized by small scale datasets in which the pairing between visual and textual elements is not available or expensive to obtain.

Tackling the aforementioned setting, we propose a semi-supervised visual-semantic embedding model (`SS-VSE`) for cross-modal retrieval in the artistic domain. Our approach relies on the construction of a common semantic embedding, in which the knowledge learned on a supervised and ordinary visual-semantic dataset is transferred to an artistic dataset in which the pairing between images and sentences is not available. After using global feature vectors, we also investigate the use of auto-encoders (`SS-VSE-AE`) to obtain more compact representations of input images and sentences. Experiments are conducted on two datasets specifically designed for the artistic domain. In particular, we collect the BibleVSA dataset which contains illustrations and textual sentences extracted from the commentaries of a historical manuscript, and exploit the SemArt dataset [48] that is instead composed of artwork images and textual comments. In this section, extensive experiments are presented to validate the proposed solution and to visualize the effect of the knowledge transfer between source and target datasets.

6.2.1 Semi-supervised cross-modal retrieval

In the following, we describe our strategy for cross-modal retrieval in the artistic domain. Our model has a two-fold role: retrieving relevant images given textual sentences as queries, and retrieve relevant sentences when given images as queries. Parameters of the model are learned with the objective of maximizing recall at K – *i.e.* the fraction of queries for which the most relevant item is ranked among the

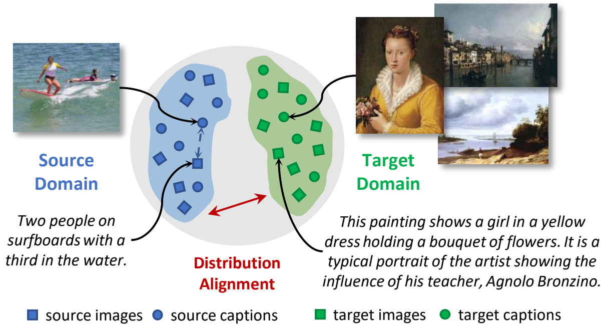

Source Domain
Two people on surfboards with a third in the water.
Distribution Alignment
Target Domain
This painting shows a girl in a yellow dress holding a bouquet of flowers. It is a typical portrait of the artist showing the influence of his teacher, Agnolo Bronzino.
■ source images
● source captions
■ target images
● target captions

Figure 6.6: Visual and textual data from the artistic domain are different from those addressed by ordinary visual-semantic datasets, posing significant challenges in the automatic understanding of arts and culture. Our approach can align illustrations and textual elements by transferring the knowledge learned on standard datasets to match images and captions coming from a target domain.

top K retrieved ones. As training data in the artistic domain is often scarce, we build a proposal that does not need a paired training set in which the associations between images and sentences are known in advance. Rather, our model transfers the knowledge learned on a source annotated dataset to a target dataset in which the pairing between the two modalities is unknown at training time.

In a nutshell, the paradigm of the common embedding space is exploited to learn similarities between images and sentences. In addition to using global feature vectors to encode data from both modalities, we also investigate the use of auto-encoders to learn more compact representations of images and sentences. To transfer knowledge to the artistic domain without leveraging annotated pairs, we devise a distribution alignment strategy based on the Maximum Mean Discrepancy measure, which aims at uncovering suitable cross-modal representation of cultural heritage data without supervision.

Visual-semantic embeddings

Aligning works of arts and their corresponding textual descriptions requires the ability to compare visual and textual data in this particular domain. Differently from the approach described in the previous section, we adopt the strategy of creating a shared multimodal embedding space, in which both textual and visual elements can be projected and compared using a similarity function.

Formally, we denote $\phi(I, \mathbf{w}_\phi) \in \mathbb{R}^{D_\phi}$ as the feature representation computed from an image I of the dataset (such as the representation coming from a CNN), and $\psi(T, \mathbf{w}_\psi) \in \mathbb{R}^{D_\psi}$ as the representation of a textual element T, computed, for example, using a text encoder on one-hot vectors, or as a function of pre-trained word embeddings. Here, $\mathbf{w}_\phi$ and $\mathbf{w}_\psi$ indicate, respectively, the learnable weights of the visual and textual encoders.

To project those representations into a common semantic space, we perform a linear projection followed by a ℓ_2-normalization step, so that the resulting embedding space lies on the ℓ_2 unit ball:

$$f(I, \mathbf{w}_f, \mathbf{w}_\phi) = \ell_{2,norm}(\mathbf{w}_i^\intercal \phi(I, \mathbf{w}_\phi)) \tag{6.8}$$

$$g(T, \mathbf{w}_g, \mathbf{w}_\psi) = \ell_{2,norm}(\mathbf{w}_c^\intercal \psi(T, \mathbf{w}_\psi)), \tag{6.9}$$

where $\ell_{2,norm}$ is the ℓ_2 normalization function. Being D the dimensionality of the joint embedding space, $\mathbf{w}_f$ is a $D_\phi \times D$ matrix, and $\mathbf{w}_g$ is a $D_\psi \times D$ matrix.

Visual and textual elements can be compared in the joint multimodal embedding space by computing the cosine similarity (equivalent, in this case, to a dot product) between their projections, so that the similarity between an image I and a caption T becomes

$$s(I, T) = f(I, \mathbf{w}_f, \mathbf{w}_\phi) \cdot g(T, \mathbf{w}_g, \mathbf{w}_\psi). \tag{6.10}$$

Clearly, the utility of the joint embedding space is maximized when it exhibits suitable cross-modality matching properties, *i.e.* when similarities in the embedding space correspond to meaningful similarities in both modalities. In this case, the embedding space acts as a bridge between the two modalities and makes it possible to retrieve textual pieces describing a query image, and images described by a query caption by identifying the closest neighbors in both modalities.

Given a dataset annotated with matching visual-semantic pairs, a good proxy of this property is to verify that corresponding pairs are neighbours in the embedding space. As a matter of fact, classical approaches have relied on the availability of paired datasets, and have learned the joint embedding for a specific domain in a completely supervised way, *e.g.* training the parameters of the model according to a Hinge triplet ranking loss with margin, which imposes suitable similarities between matching and non-matching elements. Formally, it is defined as:

$$\ell(I,T) = \sum_{\hat{T}} \left[\alpha - s(I,T) + s(I,\hat{T})\right]_+ + \\ + \sum_{\hat{I}} \left[\alpha - s(I,T) + s(\hat{I},T)\right]_+ \tag{6.11}$$

where $[x]_+ = \max(0,x)$ and α is a margin. In the equation above, (I,T) is a matching image-text pair (*i.e.*, such that T describes the content of I, and I represents the content of T), while $\hat{T}$ is a negative text with respect to I (such that $\hat{T}$ does not describe I), and $\hat{I}$ is a negative image with respect to T (such that T does not describe $\hat{I}$). The terms contained in both sums require that the difference in similarity between the matching and the non-matching pair is higher than a margin α: in the first sum, this is done by considering an image anchor and matching or non-matching captions; in the latter, instead, a caption is used as anchor.

As reported by a recent work by [45] and as seen in the previous section, in a completely supervised setting it is often beneficial to replace the sums in Eq. 6.11 with maximum operations, so to consider only the most violating non-matching pair.

Auto-encoding images and sentences

In addition to the use of plain global feature vectors, we also investigate an alternative projection strategy in which images and sentences are fed to an auto-encoder to learn a more compact yet powerful representation of the input, which can in turn be used as the input of the projection function defined in Eq. 6.8.

To this end, we design a textual auto-encoder which can convert variable-length captions to fixed-length representations from which input sentences can be reconstructed. In particular, our model exploits Gated Recurrent Networks (GRUs) [29] for both encoding and decoding. Formally, given a sentence $T = (w_1, w_2, ..., w_N)$ with length N, we firstly encode it word by word through a single-layer GRU and take the last hidden state of the Recurrent layer as the encoding of the sentence. Given the recurrent relation defined by the GRU cell and the t-h word, *i.e.*

$$\mathbf{h}_t = \text{GRU}_e(w_t, \mathbf{h}_{t-1}), \tag{6.12}$$

the encoding of the input sentence is defined as:

$$\mathbf{h}_N = \text{GRU}_e(w_N, \mathbf{h}_{N-1}). \tag{6.13}$$

In the decoding stage, the input sentence is reconstructed by feeding $\mathbf{h}_N$ to a second GRU layer which is in charge of generating the reconstructed sentence. During training, at the t-th iteration the Recurrent layer is fed with $\mathbf{h}_N$ and the previous ground-truth words, and it is trained to predict the t-h word. Formally, the training objective is thus:

$$\max_{\mathbf{w}} \sum_{t=1}^{T} \log \Pr(w_t | w_{t-1}, w_{t-2}, ..., w_1, \mathbf{h}_N). \tag{6.14}$$

The probability of a word is modeled via a softmax layer applied to the output of the decoder. To reduce the dimensionality of the decoder, a linear embedding transformation is used to project one-hot word vectors into the input space of the decoder and, vice-versa, to project the output of the decoder to the dictionary space.

Given the auto-encoder for the textual part, we build an encoder-decoder model that can take an image feature vector as input and reconstruct it starting from an intermediate and more compact representation. In practice, the encoder model is composed of a single fully connected layer. We indeed notice that a single layer leads to have a fairly informative representation of the image feature vector. Formally, we define the output of the encoder model $\mathbf{z}$ (*i.e.* the intermediate representation of the input image) as

$$\mathbf{z} = \tanh(\mathbf{W_e}\phi(I) + b_e), \tag{6.15}$$

where $\mathbf{W_e}$ and b_e are, respectively, the weight matrix and the bias vector of the encoder. Notice that the output of the encoder layer is fed through a $\tanh$ non-linearity activation function.

The decoder model has a symmetric structure. Therefore, starting from the intermediate vector $\mathbf{z}$, the decoder applies a single fully connected layer that transforms $\mathbf{z}$ to the size of the input image feature vector. Formally, the reconstructed image feature vector $\hat{\phi}(I)$ is defined as

$$\hat{\phi}(I) = \mathbf{W_d}\mathbf{z}_i + b_d, \tag{6.16}$$

where $\mathbf{W_d}$ and b_d are the weight matrix and the bias vector of the decoder. Overall, the image auto-encoder is trained to minimize the reconstruction error for each input image. We define the decoder loss function as the mean square error between the original image feature vector $\phi(I)$ and the corresponding reconstruction $\hat{\phi}(I)$.

Aligning distributions

While the knowledge of matching and non-matching pairs on a source dataset can be exploited to train the embedding space, the two reconstruction losses can be applied to both the source and the target dataset, thus building encoded representations which are suitable for both datasets. However, this is not enough to transfer knowledge from the source domain to the target domain, as there is no guarantee that encoded words and sentences from the target dataset will lie together in the embedding space.

To this end, we match the distributions of textual and visual data in the target domain, while learning from pairs sampled from the source domain. Following recent works in the field [78, 189, 219], we use the Maximum Mean Discrepancy (MMD) to compare distributions. This, basically, computes the distance between the expectations of the two distributions in a reproducing kernel Hilbert space $\mathcal{H}_\kappa$ endowed with a kernel κ, and can be used as an additional loss term:

$$\mathcal{L}_{mmd} = \|\mathbf{E}_{I\sim\mathcal{I}}\left[f(I)\right] - \mathbf{E}_{T\sim\mathcal{T}}\left[g(T)\right]\|^2_{\mathcal{H}_\kappa}, \tag{6.17}$$

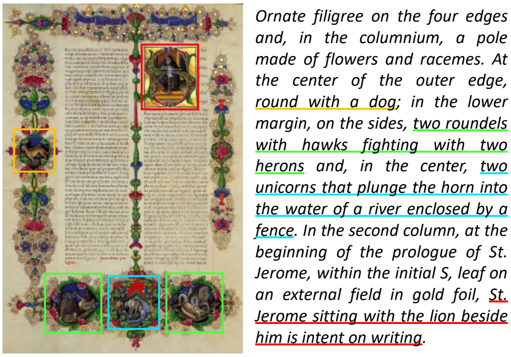
where $\mathcal{I}$ is the distribution of the illustrations, and $\mathcal{T}$ is the distribution of captions. The kernel in the MMD criterion must be a universal kernel, and thus we empirically choose a Gaussian kernel:

$$\kappa(\mathbf{x}, \mathbf{y}) = \exp\left(-\sigma\|\mathbf{x} - \mathbf{y}\|^2\right). \tag{6.18}$$

At training time, we sample two mini-batches of samples, one from the supervised set and a second one from the unsupervised dataset. The back-propagated loss is then the sum of the supervised loss (Eq. 6.11) on the supervised set, plus the MMD loss $\mathcal{L}_{mmd}$ approximated over the batch from the unsupervised set. Additionally, the two loss terms of the auto-encoders are evaluated over both the supervised and the unsupervised batches.

6.2.2 The BibleVSA dataset

In the experimental section, we evaluate the semi-supervised visual-semantic alignment strategy previously described in the context of historical manuscripts. Specifically, we consider the problem of understanding if a commentary of a digital artistic document has some parts referring to specific illustrations. In this context, we propose a new visual-semantic alignment dataset starting from the digitized version of the Borso d'Este Holy Bible, one of the most significant illustrated



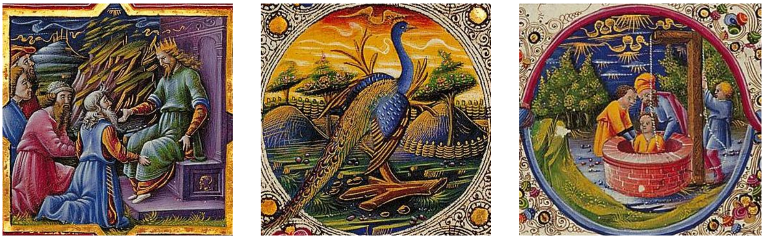
Ornate filigree on the four edges and, in the columnium, a pole made of flowers and racemes. At the center of the outer edge, round with a dog; in the lower margin, on the sides, two roundels with hawks fighting with two herons and, in the center, two unicorns that plunge the horn into the water of a river enclosed by a fence. In the second column, at the beginning of the prologue of St. Jerome, within the initial S, leaf on an external field in gold foil, St. Jerome sitting with the lion beside him is intent on writing.

Figure 6.7: A sample page from the Borso d'Este Holy Bible with the corresponding commentary and detected illustrations.

manuscripts of Renaissance. The dataset, which we name BibleVSA, provides the alignments between miniature illustrations and parts of text in the commentary, and can be used both to evaluate visual semantic embeddings, and to evaluate the alignment task.

The entire manuscript of the Borso d'Este Holy Bible consists of 320 high resolution digitized images $(3,894 \times 2,792)$, for a total of 640 pages. To extract illustrations from each page, we employ the technique proposed in [55], which has been specifically tested on the same manuscript. Results have then been manually refined in order to have a highly accurate segmentation. In a nutshell, the method of [55] begins by removing textual regions from the input image, then a block-based analysis is performed on the remaining regions. Sliding a square block over the image, they extract RGB and HSV color histograms and a Gradient Spatial Dependency Matrix as texture descriptor. Those blocks are then classified by a linear SVM as being either illustrations or decorations.

Having a reliable annotation of the bounding box of each illustration, we then exploit an Italian commentary of the Borso d'Este Holy Bible. For each page, the commentary provides a set of paragraphs describing the visual content of each of the illustrations, the decorations of the page, and of the textual content itself. We must firstly notice that, on the one hand, the commentary provides descriptions of the Bible on a per-page basis, and it is therefore well suited as a weakly-supervised form of annotation. On the other hand, the commentary contains information which are unrelated to the task at hand, so the task of aligning each illustration



Vignette depicting Solomon receiving homage from the princes.

A round with a peacock in a fenced area.

Joseph dropped by the brothers in the well.

Figure 6.8: Samples illustration-caption pairs extracted from the commentaries of the Borso d'Este Holy Bible.

with the commentary is more than just a partitioning of the text.

As an example, Fig. 6.7 reports the digitized version of one page, with its corresponding commentary and detected illustrations. The alignment is visualized by using the same color code for bounding boxes and textual strings. It can be noticed that the part of the text referring to illustrations is just a portion of the paragraph, while the remaining parts describe either the decorations (*ornate filigree on the four edges*) or the textual content (*at the beginning of the prologue of St. Jerome, within the initial S*). Also, descriptions jointly refer to the external frame and the content of the miniature, and often include names of people, saints and lineages.

We build a manual annotation of the alignments between each illustration and the commentary. Here we again employ a semi-automatic procedure: the original commentary is first automatically translated into English by using an off-the-shelf translator, and it is then manually checked. Each annotator is then asked to align each illustration with a piece of the commentary. The overall task is assisted by the fact that the commentary reports the position of each illustration inside the page (*e.g.*, *at the center of the outer edge* in Fig. 6.7); these parts are then removed from the final alignment, as they do not describe the content of the illustration.

The annotation process results in (a) a natural language caption of each illustration, which can be used for training visual-semantic embeddings, or caption generation architectures; and (b) the knowledge of which part of the commentary describes an illustration, which can be exploited for evaluating the alignment task. Fig. 6.8 reports three sample miniature-description pairs from the dataset. As the reader can witness, the gap between the visual and the textual elements is significantly higher than in usual visual-semantic datasets, both for the complexity of the illustrations, and for the high-level semantics of the captions.

Overall, the datasets consists of 2,282 annotated illustrations. Considering its twofold application (for training visual-semantic embeddings and for solving the alignment task inside a single page), we build train, validation and test splits. Firstly, all the illustrations found in pages with a single miniature are placed in the training set, to avoid trivial validation and testing cases in the alignment scenario, and enriching the training set with useful samples for training embeddings. Then, we split the remaining pages placing them in the three sets according to a 60-20-20 ratio. This results in 1,671 training, 293 validation and 307 test image-caption pairs.

6.2.3 Experimental settings

Datasets

We perform experiments on two different visual-semantic datasets containing artistic images and corresponding textual descriptions: BibleVSA, described in Sec. 6.2.2, and SemArt [48]. This dataset is composed of $21,384$ paintings extracted from the Web Gallery of Art, which contains European fine-art reproductions between the 8th and the 19th century. Each image is associated to an artistic comment and to a set of 7 different attributes comprising the title, the author, and the type of the painting. Overall, the dataset is divided in training, validation and test split with $19,244$, $1,069$ and $1,069$ elements, respectively. The average length of each artistic comment is more than 80, with a maximum number of words equal to 830. This highlights the difference between SemArt and ordinary visual-semantic datasets (*i.e.* COCO has an average caption length lower than 11) and accentuates the challenges of this set of data. To first validate our solution in a less complex scenario, we limit the validation and test set to 300 randomly selected image-text pairs. Then, we evaluate our model using a different number of retrievable items.

As source domains, we use Flickr30k and COCO which are composed of natural images and are commonly used to train cross-modal retrieval methods. For these two datasets, we use the splits provided by [90].

Implementation details

To encode input images, we use two different convolutional networks: the VGG-19 [168] and ResNet-152 [64]. We extract image features from the *fc7* layer of the

VGG-19 and from the average pooling layer of the ResNet-152 thus obtaining an input image embedding dimensionality D_ϕ of 4096 and 2048, respectively.

For encoding image descriptions, we use a GRU network [29]. We set the dimensionality of the GRU and of the joint embedding space D to 512, while the input size of word embeddings D_ψ is set to 300. We use either a text encoder on one-hot vectors or different pre-trained word embeddings (such as GloVe [147] and FastText [13]) as input of the GRU.

The model with textual and visual auto-encoders is trained using the same input and output sizes. For the training with pre-trained word embeddings, instead of using the loss function defined in Eq. 6.14, we compute the cosine distance between original and reconstructed embeddings of each word.

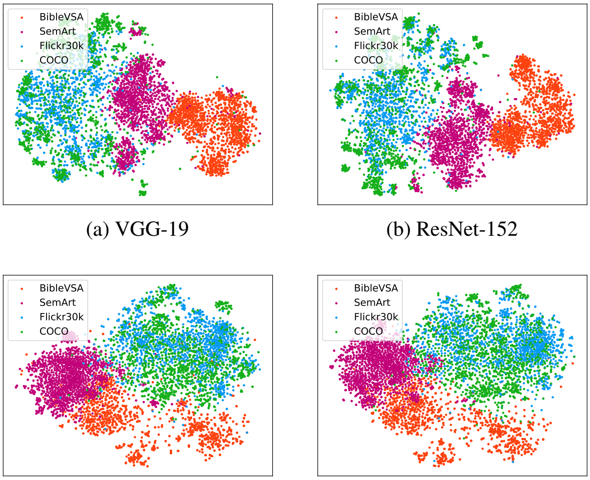
All experiments are performed by using Adam optimizer [93] with a learning rate of 0.0002 for 15 epochs and then decreased by a factor of 10. We set the margin α to 0.2, the σ parameter of the Gaussian kernel to 1 and the size of the mini-batch to 128.

6.2.4 Analysis of artistic visual-semantic data

To get an insight of characteristics of the BibleVSA and SemArt datasets, we analyze the distribution of image and textual features respectively obtained from CNNs and sentence embeddings and compare them with those extracted from classical visual-semantic datasets.

For the visual part, we extract the activation from the VGG-19 and ResNet-152 networks, while, for textual elements, we embed each word of a caption with a word embedding strategy (either GloVe or FastText). To get a feature vector for a sentence, we sum the ℓ_2 normalized embeddings of the words, and we apply the ℓ_2-norm also to the results. This strategy is largely used in image and video retrieval literature and is known for preserving the information of the original vectors into a compact representation with fixed dimensionality [185] .

Fig. 6.9 shows the distributions of visual and textual features of both datasets. To get a suitable two-dimensional representation, we run the t-SNE algorithm [126], which iteratively finds a non-linear projection that preserves the statistical distribution of the pairwise distances from the original space. As it can be observed, the features of ordinary visual-semantic datasets share almost the same visual and textual distributions. BibleVSA and SemArt, on the contrary, feature a completely different distribution, according to both modalities and all feature extractors. This underlines, on the one hand, that artistic datasets define a completely new domain. On the other hand, instead, this motivates the low



(a) VGG-19

(b) ResNet-152

(c) GloVe

(d) FastText

Figure 6.9: Comparison between the visual and textual features of ordinary visual-semantic datasets (Flickr30k, COCO) and those of BibleVSA and SemArt dataset. Visualization is obtained by running the t-SNE algorithm on top of the features. Best seen in color.

performance of existing models when tested on these datasets.

6.2.5 Cross-modal retrieval results

Firstly, we assess the performance of our full model when using different CNN features or different word embeddings, to get an insight of the role of different global feature vectors. To evaluate the effectiveness of the visual-semantic embeddings, we report rank-based performance metrics $R@K$ $(K = 1, 5, 10)$ for image and caption retrieval. In Table 6.8, we show the performance of the proposed approach on the test sets of BibleVSA and SemArt when using image features extracted, respectively, from VGG-19 and ResNet-152. Table 6.9 compares the use of FastText and GloVe embeddings versus a learned word embedding matrix. In this case, the results on SemArt test set are obtained by using 300 randomly selected retrievable items.

We limit this analysis to a single source dataset (*i.e.* COCO), as we have

Method	CNN Feat.	Text Retrieval			Image Retrieval		
		R@1	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**
		COCO → BibleVSA					
`SS-VSE`	VGG-19	**13.1**	29.5	36.1	3.9	16.7	27.5
`SS-VSE`	ResNet-152	9.8	**31.1**	**50.8**	**6.2**	**22.3**	**30.8**
`SS-VSE-AE`	VGG-19	**9.8**	**27.9**	34.4	**3.6**	15.7	25.9
`SS-VSE-AE`	ResNet-152	6.6	23.0	**36.1**	**3.6**	**19.7**	**29.8**
		COCO → SemArt					
`SS-VSE`	VGG-19	3.7	11.7	19.0	2.3	10.0	19.3
`SS-VSE`	ResNet-152	**6.7**	**19.3**	**27.0**	**5.0**	**17.3**	**29.3**
`SS-VSE-AE`	VGG-19	**5.0**	**14.3**	**22.7**	1.7	9.0	15.3
`SS-VSE-AE`	ResNet-152	4.7	12.7	21.0	**3.7**	**11.0**	**18.0**

Table 6.8: Semi-supervised cross-modal retrieval results using different visual features. Results are reported on BibleVSA and SemArt test set.

observed similar behaviours on Flickr30k. The two variants of our approach are denoted as `SS-VSE` and `SS-VSE-AE`, where the first refers to the model with global feature vectors and linear projection, and the latter refers to the model with the visual and textual auto-encoder. As it can be observed, the global descriptor extracted from ResNet-152 outperforms the one extracted from VGG-19 in almost all settings. Noticeably, learned word embeddings outperform pre-trained solutions. We speculate that this performance drop is due to the the highly specialized nature of the target datasets. In this regards, word embeddings seem to offer a poor initialization point with respect to a from-scratch learning of the word embedding matrix.

Another interesting consideration is that the use of hard negatives in the triples loss function is typically beneficial in a supervised setting [45]. Instead, in our semi-supervised setting, we do not report the same advantages in improving the alignment of the target domain.

6.2.6 Evaluation of semi-supervised embeddings

In Tables 6.10 and 6.11, we compare the performances of the two proposed semi-supervised approaches (`SS-VSE` and `SS-VSE-AE`) on BibleVSA and SemArt test set with respect to the two models trained without the distribution alignment (`VSE` and `VSE-AE`). For these experiments, we use global feature vectors extracted

Method	Word Emb.	Text Retrieval			Image Retrieval		
		R@1	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**
		COCO → BibleVSA					
`SS-VSE`	FastText	8.2	19.7	34.4	2.6	16.7	26.6
`SS-VSE`	GloVe	6.6	23.0	39.3	3.6	16.7	27.2
`SS-VSE`	-	**9.8**	**31.1**	**50.8**	**6.2**	**22.3**	**30.8**
`SS-VSE-AE`	FastText	**6.6**	**27.9**	34.4	3.3	14.4	25.2
`SS-VSE-AE`	GloVe	4.9	19.7	**41.0**	**3.9**	13.8	27.5
`SS-VSE-AE`	-	**6.6**	23.0	36.1	3.6	**19.7**	**29.8**
		COCO → SemArt					
`SS-VSE`	FastText	1.7	5.0	7.7	0.7	2.3	7.3
`SS-VSE`	GloVe	3.3	11.3	16.0	2.0	11.0	17.7
`SS-VSE`	-	**6.7**	**19.3**	**27.0**	**5.0**	**17.3**	**29.3**
`SS-VSE-AE`	FastText	3.7	10.0	17.0	3.0	9.3	11.7
`SS-VSE-AE`	GloVe	2.7	12.0	17.0	1.7	7.0	12.3
`SS-VSE-AE`	-	**4.7**	**12.7**	**21.0**	**3.7**	**11.0**	**18.0**

Table 6.9: Semi-supervised cross-modal retrieval results using different word embeddings. Results are reported on BibleVSA and SemArt test set.

from ResNet-152 and learned word embeddings. Given the significant size of SemArt dataset, we report retrieval results when using different sets of database items (*i.e.* 100, 300, 500, 1000). We notice that, when using a medium-scale source dataset like Flickr30k, the use of the auto-encoder is competitive with the use of a linear projection of the global feature vector. Instead, when transferring from a large-scale dataset like COCO, the reconstruction term is not needed and the reduced size of the representation degrades the performance. In all settings, the MMD loss gives a significant contribution to the final performance thus confirming the effectiveness of our distribution alignment strategy.

To get a better understanding of the role of the MMD loss, we also show the learned multimodal embedding space by using t-SNE visualizations. Figure 6.10 shows the embedding spaces when transferring from COCO to SemArt, with and without the MMD loss. As it can be noticed, without the MMD loss the distribution of textual and visual elements on the target domain remains almost separate, as the learning signal from the source domain is not general enough on the target domain. On the contrary, when applying the MMD loss the distribution of the learned image embeddings matches that of the textual counterpart on the

Method	Text Retrieval			Image Retrieval		
	R@1	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**
	Flickr30k → BibleVSA					
`VSE`	3.3	8.2	16.4	1.6	12.1	19.7
`SS-VSE`	**9.8**	**23.0**	**39.3**	**4.6**	**16.1**	**26.6**
`VSE-AE`	1.6	4.9	13.1	3.0	9.8	17.0
`SS-VSE-AE`	**3.3**	**23.0**	**29.5**	**3.3**	**13.1**	**23.0**
	COCO → BibleVSA					
`VSE`	1.6	9.8	16.4	2.6	10.5	20.0
`SS-VSE`	**9.8**	**31.1**	**50.8**	**6.2**	**22.3**	**30.8**
`VSE-AE`	3.3	6.6	14.8	1.6	9.8	19.7
`SS-VSE-AE`	**6.6**	**23.0**	**36.1**	**3.6**	**19.7**	**29.8**

Table 6.10: Semi-supervised retrieval results on BibleVSA test set.

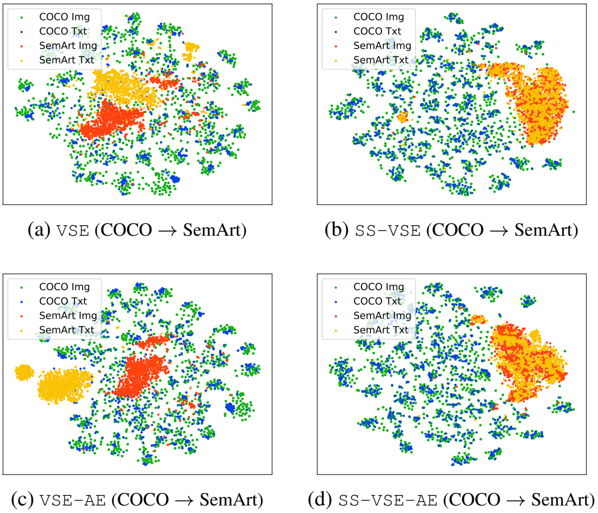
target domain, thus confirming the effectiveness of the proposed semi-supervised strategy. Noticeably, the distributions of the source and target domain still remain separate in the embedding space, thus underlying the diverse nature of the two sets.

Finally, Fig. 6.11 reports sample qualitative results on BibleVSA and SemArt dataset. As it can be noticed, our method can retrieve significant elements without employing any paired supervision from the artistic dataset.

	$N = 100$						$N = 300$					
Method	Text Retrieval			Image Retrieval			Text Retrieval			Image Retrieval		
	R@1	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**
	Flickr30k → SemArt											
VSE	2.0	10.0	14.0	3.0	11.0	17.0	1.7	5.7	8.7	1.0	5.3	7.3
SS-VSE	**7.0**	**23.0**	**40.0**	**10.0**	**23.0**	**37.0**	**5.0**	**15.3**	**22.0**	**3.7**	**13.3**	**17.7**
VSE-AE	3.0	9.0	15.0	5.0	12.0	19.0	2.3	6.0	7.3	0.7	6.0	9.0
SS-VSE-AE	**6.0**	**28.0**	**42.0**	**6.0**	**18.0**	**30.0**	**4.0**	**12.7**	**20.0**	**2.3**	**10.0**	**16.3**
	COCO → SemArt											
VSE	5.0	13.0	21.0	3.0	8.0	19.0	1.7	8.7	15.3	1.0	8.0	12.3
SS-VSE	**16.0**	**34.0**	**52.0**	**12.0**	**32.0**	**48.0**	**6.7**	**19.3**	**27.0**	**5.0**	**17.3**	**29.3**
VSE-AE	6.0	15.0	20.0	3.0	11.0	22.0	3.0	7.3	11.7	0.3	3.7	6.7
SS-VSE-AE	**7.0**	**24.0**	**39.0**	**6.0**	**17.0**	**26.0**	**4.7**	**12.7**	**21.0**	**3.7**	**11.0**	**18.0**

	$N = 500$						$N = 1000$					
Method	Text Retrieval			Image Retrieval			Text Retrieval			Image Retrieval		
	R@1	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**
	Flickr30k → SemArt											
VSE	0.8	2.6	6.0	0.4	3.6	5.6	0.5	1.6	2.8	0.1	1.2	2.8
SS-VSE	**3.6**	**9.6**	**14.6**	**1.8**	**7.6**	**12.0**	**1.5**	**6.2**	**10.0**	**1.2**	**3.5**	**7.4**
VSE-AE	1.2	4.2	6.4	0.8	3.0	5.4	0.5	2.2	4.1	0.5	1.6	3.1
SS-VSE-AE	**1.8**	**9.2**	**14.8**	**1.6**	**6.0**	**11.4**	**1.0**	**5.6**	**9.4**	**0.6**	**3.4**	**6.8**
VSE-AE	3.0	9.0	15.0	5.0	12.0	19.0	2.3	6.0	7.3	0.7	6.0	9.0
SS-VSE-AE	**6.0**	**28.0**	**42.0**	**6.0**	**18.0**	**30.0**	**4.0**	**12.7**	**20.0**	**2.3**	**10.0**	**16.3**
VSE	1.2	3.6	6.4	1.6	3.4	6.0	1.0	2.7	3.6	0.5	2.3	3.6
SS-VSE	**3.8**	**12.2**	**19.8**	**3.4**	**11.6**	**19.4**	**2.7**	**8.9**	**14.0**	**2.3**	**6.9**	**12.9**
VSE-AE	1.6	4.0	6.2	1.2	2.8	4.0	0.8	2.6	4.0	0.8	1.6	2.3
SS-VSE-AE	**2.0**	**10.0**	**15.8**	**2.2**	**5.0**	**10.8**	**0.9**	**6.1**	**10.0**	**1.0**	**3.8**	**5.8**

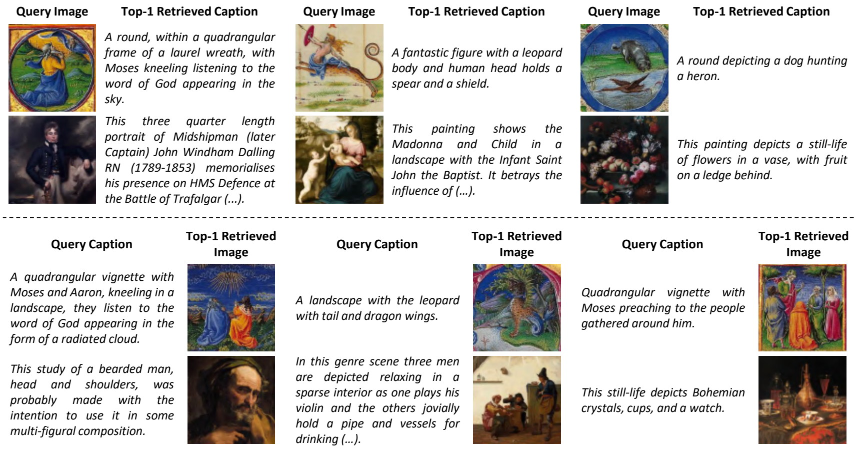
Table 6.11: Semi-supervised cross-modal retrieval results on SemArt test set using a different number N of retrievable items.



(a) VSE (COCO → SemArt) (b) SS-VSE (COCO → SemArt)

(c) VSE-AE (COCO → SemArt) (d) SS-VSE-AE (COCO → SemArt)

Figure 6.10: Comparison between t-SNE projections of the embedding spaces learned with (b-d) and without (a-c) the MMD loss. Best seen in color.



A round, within a quadrangular frame of a laurel wreath, with Moses kneeling listening to the word of God appearing in the sky.

A fantastic figure with a leopard body and human head holds a spear and a shield.

A round depicting a dog hunting a heron.

This three quarter length portrait of Midshipman (later Captain) John Windham Dalling RN (1789-1853) memorialises his presence on HMS Defence at the Battle of Trafalgar (...).

This painting shows the Madonna and Child in a landscape with the Infant Saint John the Baptist. It betrays the influence of (...).

This painting depicts a still-life of flowers in a vase, with fruit on a ledge behind.

A quadrangular vignette with Moses and Aaron, kneeling in a landscape, they listen to the word of God appearing in the form of a radiated cloud.

A landscape with the leopard with tail and dragon wings.

Quadrangular vignette with Moses preaching to the people gathered around him.

This study of a bearded man, head and shoulders, was probably made with the intention to use it in some multi-figural composition.

In this genre scene three men are depicted relaxing in a sparse interior as one plays his violin and the others jovially hold a pipe and vessels for drinking (...).

This still-life depicts Bohemian crystals, cups, and a watch.

Figure 6.11: Qualitative image-to-text (upper) and text-to-image (lower) results on BibleVSA (first and third rows) and SemArt (second and fourth rows) dataset.

Chapter 7

Conclusions

The goal of the research activities I carried out during my Ph.D. period was to develop new deep learning architectures that are able to replicate the human ability of automatically describing in natural language a given visual stimuli, mainly focusing on the salient objects present in the scene. This goal has been tackled by following two main research directions which motivate all the works presented in the previous chapters.

The first one is that of identifying which regions of an image attract human gazes at the first glance. This task, called saliency prediction, has been addressed by introducing two different models based on deep neural networks which have been demonstrated to effectively replicate human eye fixations.

The second research direction is instead that of creating a bridge between vision and language which has motivated a big part of my studies and the majority of the recent works I did. In this regard, we have first tackled the problem by exploiting saliency information to generate a textual sentence that describes a given image. This has also motivated the importance of automatically estimating the human focus of attention (*i.e.* the saliency of an image) for a wide variety of computer vision tasks, just like image captioning. Then, we have addressed the captioning task by using either fully-attentive models or more complex architectures that enable the controllability of a captioning framework from the exterior. In the last part of the thesis, we have instead focused on cross-modal retrieval, which is another important task that effectively combines vision and language.

In the following, we summarize the contributions made by this thesis and draw the final conclusions on the results achieved so far.

Saliency prediction

Motivated by the importance of automatically estimating the human focus of attention on images, we introduced two different saliency prediction models based on deep neural networks. The first model combines medium and high level features extracted from a CNN to effectively predict the salient regions of an image. While this model is based on feed-forward networks, the second network leverages on recurrent architectures. In particular, the main novelty of our second proposal is an Attentive Convolutional LSTM specifically designed to sequentially enhance saliency predictions. The same idea could potentially be employed in other tasks in which an image refinement is profitable. Furthermore, we captured an important property of human gazes by introducing in both networks a learned prior component that takes into account the center bias present in human eye fixations. The effectiveness of each component of both models has been validated through extensive evaluation on different saliency benchmarks. In particular, we showed that our second solution based on attentive mechanisms achieves state-of-the-art results and overcomes several other saliency methods by a big margin in different experimental settings.

Image captioning

Bringing together vision and language by creating deep learning architectures capable of automatically describing images in natural language is one of the core challenges in computer vision and artificial intelligence. In this context, we addressed the image captioning task from two different perspectives.

Firstly, we claimed that saliency information could be useful to condition an image captioning architecture, by providing an indication of what is salient and what is not. To this end, we proposed a novel image captioning architecture which extends the machine attention paradigm by creating two attentive paths conditioned on the output of a saliency prediction model. The first one is focused on salient regions, and the second on contextual regions: the overall model exploits the two paths during the generation of the caption, by giving more importance to salient or contextual regions as needed. The role of saliency with respect to context has been investigated by collecting statistics on semantic segmentation datasets, while the captioning model has been evaluated on large scale captioning datasets, using standard automatic metrics and by evaluating the diversity and the dictionary size of the generated corpora. Although in this work our focus has been that of demonstrating the effectiveness of saliency on captioning, rather than that of beating captioning approaches which rely on different cues, we point out that our method can be easily incorporated also in more complex architectures.

Secondly, inspired by the use of fully-attentive mechanisms in sequence modeling tasks, we presented one of the first Transformer-based architectures for image captioning. In particular, our model encapsulates a multi-layer encoder for image regions and a multi-layer decoder which generates the output sentence. To exploit both low-level and high-level contributions, encoding and decoding layers are connected in a mesh-like structure, weighted through a learnable gating mechanism. Noticeably, this connectivity pattern is unprecedented for other fully-attentive architectures. In our visual encoder, relationships between image regions are encoded in a multi-level fashion exploiting learned a priori knowledge, which is modeled via persistent memory vectors. We showed that our solution surpasses all previous proposals for image captioning, achieving a new state of the art on the online COCO evaluation server and ranking first among all other published methods. As a complementary contribution, we conducted experiments to compare different fully-attentive architectures on image captioning and validated the performance of our model when describing novel objects which are not in the training set.

Controllable captioning

Despite standard image captioning models have demonstrated a huge improvement in the last few years, these architectures still lack controllability thus limiting their application to complex scenarios. To address these issues, we presented a novel framework for image captioning which is controllable from the exterior, and which can produce natural language captions explicitly grounded on a sequence or a set of image regions. The approach explicitly considers the hierarchical structure of a sentence by predicting a sequence of noun chunks. Also, it takes into account the distinction between visual and textual words, thus providing an additional grounding at the word level. We evaluated the model with respect to a set of carefully designed baselines, on Flickr30k Entities and on COCO, which we semi-automatically augmented with grounding image regions for training and evaluation purposes. Our proposed method achieves state-of-the-art results for controllable image captioning on Flick30k and COCO both in terms of diversity and caption quality, even when compared with methods which focus on diversity.

On a different note, we also explored the possibility of controlling caption generation through people identity. In particular, we found that another ability which is still missing in modern captioning architectures is that of naming people appearing in the scene with their proper names. This can be achieved in the context of videos in which large movie description datasets are available with textual sentences annotated with character names. In this setting, we introduced a novel dataset specifically designed for supporting the development of video captioning architectures with naming capabilities. The dataset consists of visual face tracks and their association with characters' textual mentions, thus explicitly linking characters' visual appearances with their names in the textual sentences. Moreover, we presented a multimodal architecture that addresses the task of replacing generic "someone" tags with proper character names in previously generated captions. The model combines advanced natural language processing tools and state-of-the-art deep neural models for action and face recognition. Experimental results demonstrated, through extensive analyses on the proposed dataset, the effectiveness of the devised solutions and highlighted the challenges of the considered task.

Cross-modal retrieval

While image captioning models combine vision and language in a generative way, cross-modal retrieval architectures build common representations to integrate the two domains and retrieve textual elements given visual queries, and vice versa. In this context, we addressed the problem either by using supervised solutions or semi-supervised approaches exploiting the knowledge learned on large-scale dataset to retrieve items on a different domain.

Regarding the supervised setting, we presented a novel architecture that builds upon recent advances in representation learning and natural language processing and deals with text-image retrieval by casting it as a translation problem between the visual and the textual domain, further regularizing it with a cycle-consistency constraint. Experiments carried out on different datasets, demonstrated the effectiveness of our method, showing significant improvements, especially in the case of small and medium-sized datasets.

On a different setting, we tackled the task of building visual-semantic retrieval approaches for the cultural heritage and digital humanities domain. To this aim, we proposed a semi-supervised approach which does not rely on labelled data on the artistic domain and translates the knowledge learned on ordinary visual-semantic datasets to the more challenging case of artistic data. After introducing a novel dataset for the task composed of illustrations and textual commentaries coming from historical manuscripts, we validated the proposed strategy through extensive experiments and analyses.

Future works and open problems

Several research efforts can be done to further improve the results presented in this thesis, both in terms of employed techniques and novel architectures. In particular, the majority of the works presented in the previous chapters combine vision and language by using recurrent neural networks. However, as seen in Sec. 4.2, the use of fully-attentive mechanisms has recently increased the effectiveness and the performance of almost all vision and language methods, ranging from image captioning to visual question answering and cross-modal retrieval. Following this research path, some of the previously presented methods can be rethought and redesigned by using Transformer-based solutions as a starting point to better combine visual and textual features. As an example, we are currently working on the extension of the work presented in Sec. 5.1 on controllable captioning in which the recurrent neural network of the language model is replaced by a Transformer-based architecture.

On a related line, one of the lacks of current captioning models is the ability to mention objects which are not included in the training set. Despite some recent advancements in the field, the performances of these automatic models are still far from that of humans, thus making it difficult to apply captioning in challenging and complex scenarios. This setting, which is also called novel object captioning, is one of the open challenges for vision-and-language applications and needs to be widely explored by future researches in this field.

Publications and achievements

The efforts presented in this thesis have resulted in publications in international conferences and journals. Among all the others, the work on grounded and controllable captioning, presented in Sec. 5.1, has been accepted at the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2019. Also, the research activities on fully-attentive models for image captioning, like the work on the $\mathcal{M}^2$ Transformer reported in Sec. 4.2, have been accepted at the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020 and the IEEE International Conference on Robotics and Automation 2020.

The works on saliency prediction, as well, have resulted in several publications which have been widely appreciated by the community. In particular, the work on the Multi-Level Network, presented in Sec. 3.1, is currently the most cited paper of the International Conference on Pattern Recognition 2016, while the Saliency Attentive Model, described in Sec. 3.2, has won the LSUN Saliency Prediction

Challenge 2017 and has resulted in a journal paper on IEEE Transactions on Image Processing. Some of the other presented results, instead, are currently under revision in major conferences or journals.

As complementary result of this thesis, I have developed together with other colleagues from my lab a PyTorch library, called Speaksee[1], specifically designed for visual-semantic tasks. It contains utility functions and re-implementations of state-of-the-art models for different tasks that combine vision and language, such as image captioning, cross-modal retrieval, and visual-question answering. It is worth noting that almost all our recent works that integrate visual-semantic techniques are based on this library.

[1] https://github.com/aimagelab/speaksee

Appendix A

Other research activities

In the following pages, I briefly report other research activities I carried out during my Ph.D. period that did not fall under the line of this thesis and were therefore not discussed in the previous chapters. Many of these works have been done in collaboration with other Ph.D. students and researchers from my lab, whom I must thank for their cooperation and for the work we did together. The three main topics that fall under this category are:

- image-to-image translation (Sec. A.1);
- face retrieval (Sec. A.2);
- vision-and-language navigation (Sec. A.3).

A.1 Image-to-image translation

Our society has inherited a huge legacy of cultural artifacts from past generations: buildings, monuments, books, and exceptional works of art. While this heritage would benefit from algorithms which can automatically understand its content, computer vision techniques have been rarely adapted to work in this domain.

One of the reasons is that applying state-of-the-art techniques to artworks is rather difficult, and often brings poor performance. This can be motivated by the fact that the visual appearance of artworks is different from that of photo-realistic images, due to the presence of brush strokes, the creativity of the artist and the specific artistic style at hand. As current vision pipelines exploit large datasets consisting of natural images, learned models are largely biased towards them. The result is a gap between high-level convolutional features of the two domains, which leads to a decrease in performance in the target tasks, such as classification, detection or segmentation.

We present a solution to the aforementioned problem that avoids the need for re-training neural architectures on large-scale datasets containing artistic images. In particular, we propose an architecture which can reduce the shift between the feature distributions from the two domains, by translating artworks to photo-realistic images which preserve the original content. As paired training data is not available for this task, we revert to an unpaired image-to-image translation setting [241], in which images can be translated between different domains while preserving some underlying characteristics. In our *art-to-real* scenario, the first domain is that of paintings while the second one is that of natural images. The shared characteristic is that they are two different visualizations of the same class of objects, for example, they both represent landscapes.

In the translation architecture that we propose, new photo-realistic images are obtained by retrieving and learning from existing details of natural images and exploiting a weakly-supervised semantic understanding of the artwork. To this aim, a number of memory banks of realistic patches is built from the set of photos, each containing patches from a single semantic class in a memory-efficient representation. By comparing generated and real images at the patch level, in a multi-scale manner, we can then drive the training of a generator network which learns to generate photo-realistic details, while preserving the semantics of the original painting. As performing a semantic understanding of the painting would create a chicken-egg problem, in which unreliable data is used to drive the training

This section is related to publications [15, 19, 20] reported in Appendix B, by the author of the thesis. See Appendix B for details.

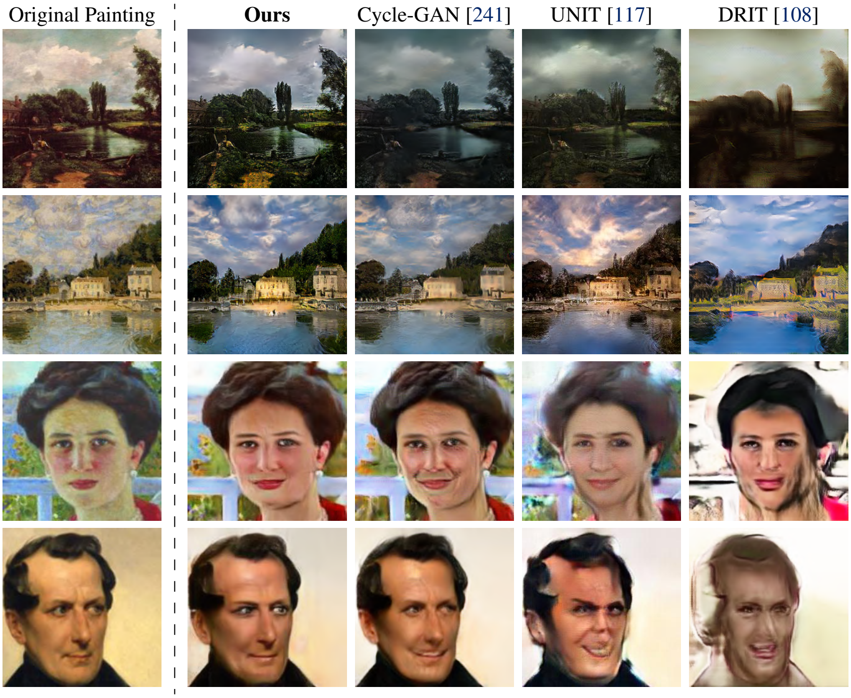


Figure A.1: Qualitative results on landscape paintings and portraits.

and the generation, we propose a strategy to update the semantic masks during training, leveraging the partial convergence of a cycle-consistent framework.

We apply our model to a wide range of artworks which include paintings from different artists and styles, landscapes and portraits. Through experimental evaluation, we show that our architecture can improve the realism of translated images when compared to state-of-the-art unpaired translation techniques. Fig. A.1 shows some qualitative results of our approach, compared to those from other methods. Overall, we observe increased realism in our generated images, due to more detailed elements and fewer blurred areas, especially in the landscape results. Portrait samples reveal that brush strokes disappear completely, leading to a photo-realistic visualization. Our results contain fewer artifacts and are more faithful to the paintings, more often preserving the original facial expression.

A.2 Face retrieval

Even though there isn't an artwork depicting yourself, there are probably some which contains faces that look just like you [32]. Inspired by this consideration, we present an interactive framework which can retrieve similar faces from a database of paintings, given a query photo from a visitor. Once the visitor has arrived to the museum, we imagine a situation where he can enter a photo boot, take a photo of himself, and get back the name and the location of the painting where his doppelgänger lies.

The problem is that of finding faces that look similar between the real domain (that of the visitor) and the artistic one (that of paintings). As images from these two domains can look very different in terms of low-level statistics, the task demands for a domain adaptation stage, in which different features coming from the two domains can be merged together. Additionally, there is no supervision for the task, as no dataset contains photo of real people annotated with the most similar paintings in a given collection.

To tackle these issues, we firstly define a deep learning-based domain translation strategy, which let us recover artistic proxies of real faces, so to generate synthetic artworks while still exploiting potentially useful annotations coming from real datasets. Secondly, we employ the supervision given by datasets for tasks which are similar to the one we are addressing. We concentrate on face re-

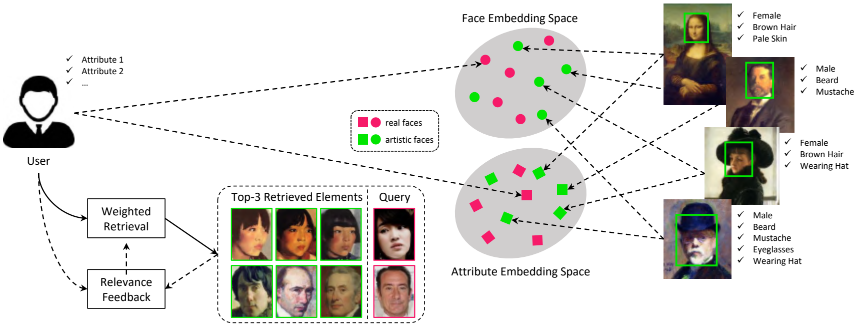


Figure A.2: Overview of our approach: given a real face as query, we retrieve similar faces from paintings, jointly taking into account the aesthetic similarity and the matching between semantic attributes. The user can further refine the retrieved set by providing feedbacks and imposing constraints on attributes.

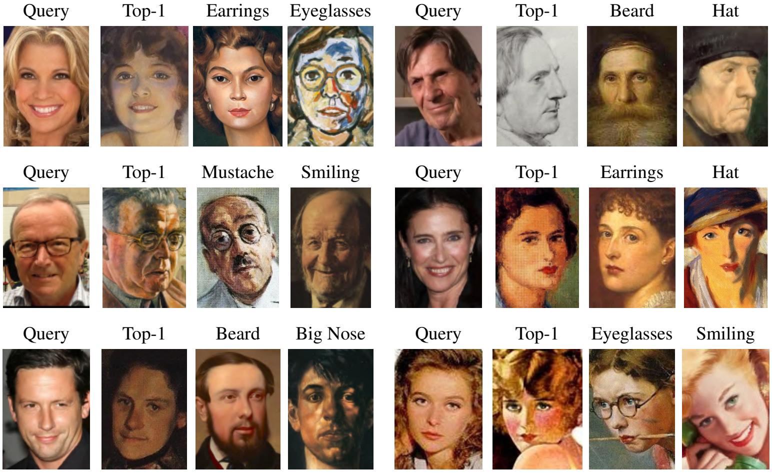


Figure A.3: Top-1 retrieved results from WikiArt with attribute constraints.

cognition (*i.e.* recognizing images of the same person) and face attribute detection (*i.e.* recognizing whether a face contains a given set of attributes). While both these tasks are different from the one we are considering, their supervision is still useful to learn on our task.

Finally, we also let the user interact with the application. By a first mean, the user can provide feedbacks on retrieved results by telling the application which faces do look similar to him, and which are not satisfactory. This, in combination with a relevance feedback strategy, let us recover better and more subjective results at each iteration. Fig. A.2 shows the overview of our approach.

The proposal is evaluated on both synthetic and more realistic settings, on a variety of different datasets. In particular, we employ a style-transferred version of Celeb-A [121], as it contains annotations for face and attribute recognition. Additionally, we evaluate on WebCaricature [79], which contains caricatures of famous people, and collect an additional test set containing real and unpaired artworks. Fig. A.3 shows some qualitative results using query images from Celeb-A and a set of artworks from WikiArt[1] as retrievable items. For each sample, we report the top-1 retrieved result without constraints and the top-1 retrieved results by imposing the presence of some specific facial attributes.

[1] https://www.wikiart.org/

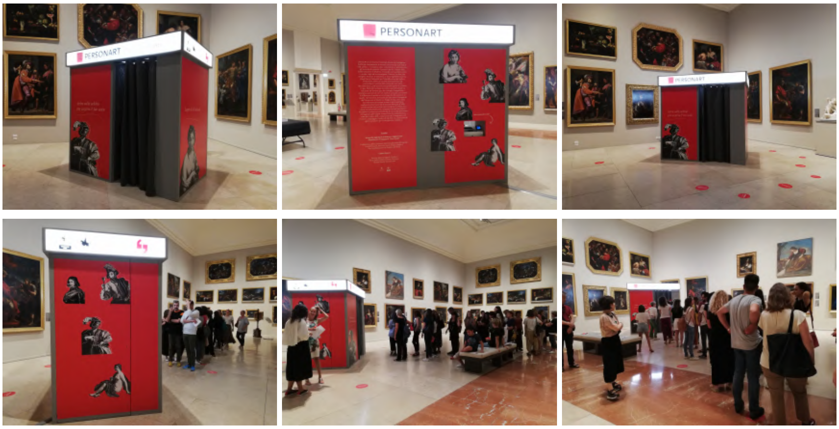


Figure A.4: Images of *PersonArt*, an interactive exhibition at the Estense Gallery of Modena in which a visitor can discover his/her own doppelgänger among the portraits of the museum.

PersonArt

This work is resulted in an interactive demo, called *PersonArt*, at the Estense Gallery of Modena in which a photo boot has been installed during a local event collecting more than $1,100$ visitors (approximately 400 males and 750 females) in just three days. In this event, each visitor had the possibility of interacting with our application and finding the top-3 most similar paintings, returned jointly taking into account the aesthetic similarity and the matching between facial attributes. After choosing a painting among the top-3 results, the visitor could discover his/her own doppelgänger inside the museum following the detailed instructions printed by the system. Figure A.4 reports some images of the exhibition.

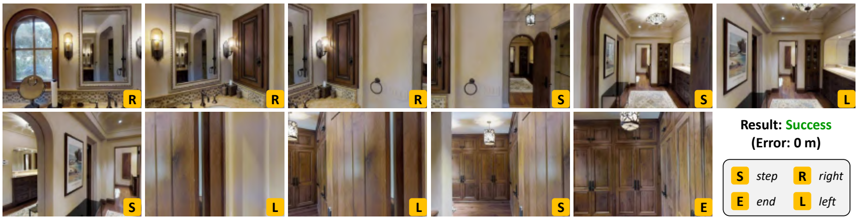
A.3 Vision-and-language navigation

Effective instruction-following and contextual decision-making can open the door to a new world for researchers in embodied AI. Deep neural networks have the potential to build complex reasoning rules that enable the creation of intelligent agents, and research on this subject could also help to empower the next generation of collaborative robots. In this scenario, Vision-and-Language Navigation (VLN) [5] plays a significant part in current research. This task requires to follow natural language instructions through unknown environments, discovering the correspondences between lingual and visual perception step by step. Additionally, the agent needs to progressively adjust navigation in light of the history of past actions and explored areas. Even a small error while planning the next move can lead to failure because perception and actions are unavoidably entangled; indeed, *we must perceive in order to move, but we must also move in order to perceive* [49]. For this reason, the agent can succeed in this task only by efficiently combining the three modalities – language, vision, and actions.

We propose to exploit fully-attentive networks to merge the knowledge coming from different domains. Encouraged by recent work on fully-attentive networks [192], we devise *Perceive, Transform, and Act* (PTA), a novel architecture for VLN in which the different modalities are free to be conditioned on the full history of previous actions. While previous approaches rely on a recurrent policy to track the agent's internal status through time [5, 210, 125], we directly infer the state from the observations via attention and avoid the critical back-propagation through time. For this reason, our agent can model the dependencies tied to navigation more efficiently and generalize to longer episodes better than other models. To the best of our knowledge, our model is the first Transformer-like architecture to merge three different modalities.

Another challenge is represented by the agent adaptability to real-world applications. Recent literature identifies two main operating settings for VLN, called *low-level action space* and *high-level action space* [107]. Low-level methods make predictions over an output space of known dimension, which corresponds to the agent locomotor system – *rotate* $X°$, *tilt up/down*, and *step forward* are examples of low-level actions. The concept of a high-level, *panoramic* action space was first proposed by Fried *et al.* [46]: differently from the low-level output space, it aims to predict the path to the goal without decoding the sequence of atomic actions explicitly. In this setting, the agent can move inside the environment using

This section is related to the publication [22] reported in Appendix B, by the author of the thesis. See Appendix B for details.



Instruction: *Exit the bathroom and walk down the hall to the second doorway on your left. Turn left and enter the room through that doorway.*

Figure A.5: Navigation episode from the R2R dataset [5]. For each step, we report the agent first-person point of view and the next predicted action (from left to right, top to bottom).

a teleporting system. We believe that this aspect limits adaptability to real-world applications, and for this reason we design our model for low-level use.

Our goal is to navigate unseen indoor environments with the only help of natural language instructions and egocentric visual observations. To merge multimodal knowledge coming from the environment, we devise a two-stage encoder which exploits both temporal and spatial attention. At each time step, the agent selects a move to progress towards the goal. To that end, we fuse contextual information with the history of actions via attention and build a multimodal decoder which merges the three modalities: actions, images, and text. We then decode a probability distribution over a low-level output space in which possible actions are atomic moves like *turn* or *step ahead*. After a first phase in which we train the agent with classical imitation learning, we implement an extrinsic reward function to promote coherence between ground-truth and predicted trajectories.

Experimental results show that PTA achieves state-of-the-art performance on low-level VLN which is close to real-world applications and requires to decode fine-grained atomic actions. Fig. A.5 shows a qualitative result from the R2R dataset [5]. Notably, PTA is able to ground concepts such as "*the second doorway on your left*" and terminates the navigation episode successfully. Since our agent operates in a low-level setup, it needs to orientate towards the next viewpoint before stepping ahead, making the decoding phase more challenging.

Appendix B

List of publications

The following list of publications includes all conference papers, journal articles, and book chapters published during my Ph.D. period, as well as recent pre-prints. Content and experimental results published in some of these papers have been included in the previous chapters, with explicit permission given by the other authors.

[1] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. Multi-Level Net: A Visual Saliency Prediction Model. In *Proceedings of the European Conference on Computer Vision Workshops*, 2016.

[2] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. A Deep Multi-Level Network for Saliency Prediction. In *Proceedings of the International Conference on Pattern Recognition*. IEEE, 2016.

[3] Stefano Pini, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Towards video captioning with naming: A novel dataset and a multi-modal approach. In *Proceedings of the International Conference on Image Analysis and Processing*, 2017.

[4] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. Visual saliency for image captioning in new multimedia services. In *Proceedings of the IEEE International Conference on Multimedia & Expo Workshops*, 2017.

[5] Stefano Pini, Olfa Ben Ahmed, Marcella Cornia, Lorenzo Baraldi, Rita Cucchiara, and Benoit Huet. Modeling multimodal cues in a deep learning-based framework for emotion recognition in the wild. In *Proceedings of the ACM International Conference on Multimodal Interaction*, 2017.

[6] Marcella Cornia, Davide Abati, Lorenzo Baraldi, Andrea Palazzi, Simone Calderara, and Rita Cucchiara. Attentive models in vision: Computing saliency maps in the deep learning era. In *Proceedings of the Conference of the Italian Association for Artificial Intelligence*, 2017.

[7] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. Predicting Human Eye Fixations via an LSTM-based Saliency Attentive Model. *IEEE Transactions on Image Processing*, 27(10):5142–5154, 2018.

[8] Marcella Cornia, Davide Abati, Lorenzo Baraldi, Andrea Palazzi, Simone Calderara, and Rita Cucchiara. Attentive models in vision: Computing saliency maps in the deep learning era. *Intelligenza Artificiale*, 12(2):161–175, 2018.

[9] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. SAM: Pushing the Limits of Saliency Prediction Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018.

[10] Marcella Cornia, Stefano Pini, Lorenzo Baraldi, and Rita Cucchiara. Automatic image cropping and selection using saliency: An application to historical manuscripts. In *Proceedings of the Italian Research Conference on Digital Libraries*, 2018.

[11] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. Paying more attention to saliency: Image captioning with saliency and context attention. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 14(2):48, 2018.

[12] Lorenzo Baraldi, Marcella Cornia, Costantino Grana, and Rita Cucchiara. Aligning text and document illustrations: Towards visually explainable digital humanities. In *Proceedings of the International Conference on Pattern Recognition*, 2018.

[13] Marcella Cornia, Lorenzo Baraldi, Hamed R Tavakoli, and Rita Cucchiara. Towards Cycle-Consistent Models for Text and Image Retrieval. In *Proceedings of the European Conference on Computer Vision Workshops*, 2018.

[14] Angelo Carraggi, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Visual-Semantic Alignment Across Domains Using a Semi-Supervised Approach. In *Proceedings of the European Conference on Computer Vision Workshops*, 2018.

[15] Matteo Tomei, Lorenzo Baraldi, Marcella Cornia, and Rita Cucchiara. What was Monet seeing while painting? Translating artworks to photo-realistic images. In *Proceedings of the European Conference on Computer Vision Workshops*, 2018.

[16] Stefano Alletto, Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. Recognizing social relationships from an egocentric vision perspective. In *Multimodal Behavior Analysis in the Wild*, pages 199–224. Elsevier, 2019.

[17] Stefano Pini, Marcella Cornia, Federico Bolelli, Lorenzo Baraldi, and Rita Cucchiara. M-VAD Names: A Dataset for Video Captioning with Naming. *Multimedia Tools and Applications*, 78(10):14007–14027, 2019.

[18] Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Show, Control and Tell: A Framework for Generating Controllable and Grounded Captions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[19] Matteo Tomei, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Art2Real: Unfolding the Reality of Artworks via Semantically-Aware Image-to-Image Translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[20] Matteo Tomei, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Image-to-Image Translation to Unfold the Reality of Artworks: an Empirical Analysis. In *Proceedings of the International Conference on Image Analysis and Processing*, 2019.

[21] Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Massimiliano Corsini, and Rita Cucchiara. Artpedia: A New Visual-Semantic Dataset with Visual and Contextual Sentences in the Artistic Domain. In *Proceedings of the International Conference on Image Analysis and Processing*, 2019.

[22] Federico Landi, Lorenzo Baraldi, Marcella Cornia, Massimiliano Corsini, and Rita Cucchiara. Perceive, Transform, and Act: Multi-Modal Attention Networks for Vision-and-Language Navigation. *arXiv preprint arXiv:1911.12377*, 2019.

[23] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, Massimiliano Corsini, and Rita Cucchiara. Explaining digital humanities by aligning images and textual descriptions. *Pattern Recognition Letters*, 129:166–172, 2020.

[24] Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. SMArT: Training Shallow Memory-aware Transformers for Robotic Explainability. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2019.

[25] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. $\mathcal{M}^2$: Meshed-Memory Transformer for Image Captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Bibliography

[1] Harsh Agrawal, Karan Desai, Xinlei Chen, Rishabh Jain, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. nocaps: novel object captioning at scale. In *Proceedings of the International Conference on Computer Vision*, 2019. 75, 77, 86

[2] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. SPICE: Semantic Propositional Image Caption Evaluation. In *Proceedings of the European Conference on Computer Vision*, 2016. 75, 102

[3] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2017. 11, 47, 86

[4] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 10, 14, 68, 74, 75, 77, 80, 83, 86, 91, 96, 97, 98, 103, 106, 107, 129

[5] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 129, 175, 176

[6] Jyoti Aneja, Aditya Deshpande, and Alexander G Schwing. Convolutional image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 10, 68

[7] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Visual tracking with online multiple instance learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 113

[8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*, 2015. 32, 58

[9] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the Annual Meeting on Association for Computational Linguistics Workshops*, 2005. 57, 75, 102, 143

[10] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. Hierarchical Boundary-Aware Neural Encoder for Video Captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 12, 22

[11] Loris Bazzani, Hugo Larochelle, and Lorenzo Torresani. Recurrent Mixture Density Network for Spatiotemporal Visual Attention. In *Proceedings of the International Conference on Learning Representations*, 2017. 9

[12] Piotr Bojanowski, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Finding actors and actions in movies. In *Proceedings of the International Conference on Computer Vision*, 2013. 12, 118

[13] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. 132, 139, 156

[14] Ali Borji. Boosting bottom-up and top-down visual features for saliency estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 8

[15] Ali Borji. Saliency prediction in the deep learning era: Successes and limitations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 7

[16] Ali Borji and Laurent Itti. State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):185–207, 2012. 7

[17] Ali Borji and Laurent Itti. CAT2000: A Large Scale Fixation Dataset for Boosting Saliency Research. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015. 20, 29, 37, 38, 42

[18] Ali Borji, Hamed R Tavakoli, Dicky N Sihite, and Laurent Itti. Analysis of Scores, Datasets, and Models in Visual Saliency Prediction. In *Proceedings of the International Conference on Computer Vision*, 2013. 40

[19] Neil Bruce and John Tsotsos. Saliency based on information maximization. In *Advances in Neural Information Processing Systems*, 2006. 7, 29, 41, 43

[20] Guy Thomas Buswell. How People Look at Pictures: A Study of The Psychology and Perception in Art. University of Chicago Press, 1935. 15

[21] Zoya Bylinskii, Tilke Judd, Ali Borji, Laurent Itti, Frédo Durand, Aude Oliva, and Antonio Torralba. MIT Saliency Benchmark. http://saliency.mit.edu/. 29, 39

[22] Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand. What do different evaluation metrics tell us about saliency models? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(3):740–757, 2018. 29, 39

[23] Zoya Bylinskii, Adrià Recasens, Ali Borji, Aude Oliva, Antonio Torralba, and Frédo Durand. Where should saliency models look next? In *Proceedings of the European Conference on Computer Vision*, 2016. 43, 50

[24] Moran Cerf, Jonathan Harel, Wolfgang Einhäuser, and Christof Koch. Predicting human gaze using low-level saliency combined with face detection. In *Advances in Neural Information Processing Systems*, 2008. 8

[25] Danqi Chen and Christopher Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2014. 92

[26] Tseng-Hung Chen, Yuan-Hong Liao, Ching-Yao Chuang, Wan-Ting Hsu, Jianlong Fu, and Min Sun. Show, adapt and tell: Adversarial training of crossdomain image captioner. In *Proceedings of the International Conference on Computer Vision*, 2017. 14

[27] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive Faster R-CNN for object detection in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 14

[28] Yuhua Chen, Wen Li, and Luc Van Gool. ROAD: Reality Oriented Adaptation for Semantic Segmentation of Urban Scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 14

[29] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2014. 132, 150, 156

[30] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In *Advances in Neural Information Processing Systems*, 2015. 17

[31] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 50, 51, 52

[32] Elliot J Crowley, Omkar M Parkhi, and Andrew Zisserman. Face Painting: Querying Art with Photos. In *Proceedings of the British Machine Vision Conference*, 2015. 172

[33] Bo Dai, Sanja Fidler, Raquel Urtasun, and Dahua Lin. Towards Diverse and Natural Image Descriptions via a Conditional GAN. In *Proceedings of the International Conference on Computer Vision*, 2017. 11

[34] Aditya Deshpande, Jyoti Aneja, Liwei Wang, Alexander Schwing, and David A Forsyth. Diverse and controllable image captioning with part-of-speech guidance. *arXiv preprint arXiv:1805.12589*, 2018. 11, 105

[35] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 10, 68

[36] Lei Ding and Alper Yilmaz. Learning relations among movie characters: A social network perspective. In *Proceedings of the European Conference on Computer Vision*, 2010. 13

[37] Samuel Dodge and Lina Karam. Visual Saliency Prediction Using a Mixture of Deep Neural Networks. *IEEE Transactions on Image Processing*, 27(8):4080–4090, 2018. 8, 41, 42

[38] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 9, 11, 12, 22

[39] Jianfeng Dong, Xirong Li, and Cees GM Snoek. Word2VisualVec: Image and Video to Sentence Matching by Visual Feature Prediction. *arXiv preprint arXiv:1604.06838*, 2016. 14, 141, 142, 144

[40] Aviv Eisenschtat and Lior Wolf. Linking image and text with 2-way nets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 141, 142, 143

[41] Martin Engilberge, Louis Chevallier, Patrick Pérez, and Matthieu Cord. Finding beans in burgers: Deep semantic-visual embedding with localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 13, 137

[42] Erkut Erdem and Aykut Erdem. Visual saliency estimation by nonlinearly integrating features using region covariances. *Journal of Vision*, 13(4):11–11, 2013. 8

[43] Mark Everingham, Josef Sivic, and Andrew Zisserman. "Hello! My name is... Buffy"–Automatic Naming of Characters in TV Video. In *Proceedings of the British Machine Vision Conference*, 2006. 12

[44] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 29, 51

[45] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. VSE++: Improving Visual-Semantic Embeddings with Hard Negatives. In *Proceedings of the British Machine Vision Conference*, 2018. 13, 129, 130, 133, 137, 142, 143, 150, 158

[46] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. In *Advances in Neural Information Processing Systems*, 2018. 129, 175

[47] Chuang Gan, Zhe Gan, Xiaodong He, Jianfeng Gao, and Li Deng. Stylenet: Generating attractive visual captions with styles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 11

[48] Noa Garcia and George Vogiatzis. How to Read Paintings: Semantic Art Understanding with Multi-Modal Retrieval. In *Proceedings of the European Conference on Computer Vision Workshops*, 2018. 14, 147, 155

[49] James J Gibson. *The Ecological Approach to Visual Perception: Classic Edition*. Psychology Press, 2014. 175

[50] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2010. 32, 58, 76

[51] Stas Goferman, Lihi Zelnik-Manor, and Ayellet Tal. Context-aware saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):1915–1926, 2012. 8

[52] Yoav Goldberg and Joakim Nivre. Training deterministic parsers with non-deterministic oracles. *Transactions of the Association of Computational Linguistics*, 1:403–414, 2013. 92

[53] Ke Gong, Xiaodan Liang, Xiaohui Shen, and Liang Lin. Look into Person: Self-supervised Structure-sensitive Learning and A New Benchmark for Human Parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 50, 51, 52

[54] Siavash Gorji and James J Clark. Going from image to video saliency: Augmenting image salience with dynamic attentional push. In *Proceedings*

of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 9

[55] Costantino Grana, Daniele Borghesani, and Rita Cucchiara. Automatic segmentation of digitalized historical manuscripts. *Multimedia Tools and Applications*, 55(3):483–506, 2011. 153

[56] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. DRAW: A recurrent neural network for image generation. In *Proceedings of the International Conference on Learning Representations*, 2015. 16

[57] Zenzi M Griffin and Kathryn Bock. What the eyes say about speaking. *Psychological Science*, 11(4):274–279, 2000. 49

[58] Jiuxiang Gu, Jianfei Cai, Shafiq R Joty, Li Niu, and Gang Wang. Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 13, 137

[59] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. MS-Celeb-1M: A dataset and benchmark for large-scale face recognition. In *Proceedings of the European Conference on Computer Vision*, 2016. 114, 120

[60] Hadi Hadizadeh and Ivan V Bajic. Saliency-Aware Video Compression. *IEEE Transactions on Image Processing*, 23(1):19–33, 2014. 15

[61] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. In *Advances in Neural Information Processing Systems*, 2006. 7, 15, 42, 43, 62

[62] Kaiming He and Jian Sun. Convolutional Neural Networks at Constrained Time Cost. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 26

[63] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 76

[64] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 8, 26, 57, 75, 98, 137, 138, 155

[65] Sen He, Nicolas Pugeault, Yang Mi, and Ali Borji. What Catches the Eye? Visualizing and Understanding Deep Saliency Models. *arXiv preprint arXiv:1803.05753*, 2018. 41

[66] Sen He, Hamed R Tavakoli, Ali Borji, and Nicolas Pugeault. Human Attention in Image Captioning: Dataset and Analysis. In *Proceedings of the International Conference on Computer Vision*, 2019. 11

[67] Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. Image Captioning: Transforming Objects into Words. *arXiv preprint arXiv:1906.05963*, 2019. 10, 80

[68] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997. 22, 134

[69] Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013. 56, 136

[70] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. CyCADA: Cycle-Consistent Adversarial Domain Adaptation. In *Proceedings of the International Conference on Machine Learning*, 2018. 14

[71] Matthew Honnibal, Yoav Goldberg, and Mark Johnson. A non-monotonic arc-eager transition system for dependency parsing. In *Proceedings of the Conference on Computational Natural Language Learning*, 2013. 92

[72] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 11

[73] Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. Natural language object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 11

[74] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on Attention for Image Captioning. In *Proceedings of the International Conference on Computer Vision*, 2019. 10, 78, 80, 81, 83

[75] Xun Huang, Chengyao Shen, Xavier Boix, and Qi Zhao. SALICON: Reducing the Semantic Gap in Saliency Prediction by Adapting Deep Neural Networks. In *Proceedings of the International Conference on Computer Vision*, 2015. 8, 16, 41, 42

[76] Yan Huang, Wei Wang, and Liang Wang. Instance-aware Image and Sentence Matching with Selective Multimodal LSTM. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 142, 143

[77] Yan Huang, Qi Wu, Chunfeng Song, and Liang Wang. Learning semantic concepts and order for image and sentence matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 13, 137

[78] Yao-Hung Hubert Tsai, Yi-Ren Yeh, and Yu-Chiang Frank Wang. Learning cross-domain landmarks for heterogeneous domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 152

[79] Jing Huo, Wenbin Li, Yinghuan Shi, Yang Gao, and Hujun Yin. WebCaricature: A Benchmark for Caricature Face Recognition. In *Proceedings of the British Machine Vision Conference*, 2018. 173

[80] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-Domain Weakly-Supervised Object Detection through Progressive Domain Adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 14

[81] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998. 7, 15, 43

[82] Saumya Jetley, Naila Murray, and Eleonora Vig. End-to-End Saliency Mapping via Probability Distribution Prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 8, 16, 35, 42

[83] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. Salient Object Detection: A Discriminative Regional Feature Integration Approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 15

[84] Ming Jiang, Shengsheng Huang, Juanyong Duan, and Qi Zhao. SALICON: Saliency in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 8, 10, 28, 34, 35, 36, 37, 38, 40, 41, 44, 45, 56

[85] Wenhao Jiang, Lin Ma, Yu-Gang Jiang, Wei Liu, and Tong Zhang. Recurrent Fusion Network for Image Captioning. In *Proceedings of the European Conference on Computer Vision*, 2018. 80, 81, 83

[86] SouYoung Jin, Hang Su, Chris Stauffer, and Erik Learned-Miller. End-to-end Face Detection and Cast Grouping in Movies Using Erdos-Rényi Clustering. In *Proceedings of the International Conference on Computer Vision*, 2017. 13

[87] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. DenseCap: Fully convolutional Localization Networks for Dense Captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 9, 11

[88] Tilke Judd, Frédo Durand, and Antonio Torralba. A benchmark of computational models of saliency to predict human fixations. In *MIT Technical Report*, 2012. 29, 30, 42

[89] Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba. Learning to predict where humans look. In *Proceedings of the International Conference on Computer Vision*, 2009. 8, 15, 17, 20, 29, 37, 38, 42, 45, 50

[90] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 22, 47, 75, 101, 119, 136, 155

[91] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 118

[92] Lei Ke, Wenjie Pei, Ruiyu Li, Xiaoyong Shen, and Yu-Wing Tai. Reflective Decoding Network for Image Captioning. In *Proceedings of the International Conference on Computer Vision*, 2019. 83

[93] Diederik Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations*, 2015. 58, 76, 138, 156

[94] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. In *Proceedings of Neural Information Processing Systems Workshops*, 2014. 13, 119, 129, 130

[95] Benjamin Klein, Guy Lev, Gil Sadeh, and Lior Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 141, 142, 143

[96] Christof Koch and Shimon Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of Intelligence*. Springer, 1987. 7

[97] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-Captioning Events in Videos. In *Proceedings of the International Conference on Computer Vision*, 2017. 12

[98] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017. 75, 98

[99] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, 2012. 8

[100] Srinivas SS Kruthiventi, Kumar Ayush, and R Venkatesh Babu. DeepFix: A Fully Convolutional Neural Network for predicting Human Eye Fixations. *IEEE Transactions on Image Processing*, 26(9):4446–4456, 2017. 8, 16, 17, 20, 42

[101] Srinivas SS Kruthiventi, Vennela Gudisa, Jaley H Dholakiya, and R Venkatesh Babu. Saliency Unified: A Deep Architecture for Simultaneous Eye Fixation Prediction and Salient Object Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 8, 35, 41

[102] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. 98, 99, 103, 113

[103] Matthias Kümmerer, Lucas Theis, and Matthias Bethge. DeepGaze I: Boosting saliency prediction with feature maps trained on ImageNet. In *Proceedings of the International Conference on Learning Representations Workshops*, 2015. 8, 16, 17, 20

[104] Matthias Kümmerer, Thomas SA Wallis, and Matthias Bethge. Information-Theoretic Model Comparison Unifies Saliency Metrics. *National Academy of Sciences*, 112(52):16054–16059, 2015. 29, 39

[105] Matthias Kümmerer, Thomas SA Wallis, Leon A Gatys, and Matthias Bethge. Understanding low-and high-level contributions to fixation prediction. In *Proceedings of the International Conference on Computer Vision*, 2017. 8, 17, 20, 40, 41, 42, 44, 45

[106] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*, 2018. 75

[107] Federico Landi, Lorenzo Baraldi, Massimiliano Corsini, and Rita Cucchiara. Embodied Vision-and-Language Navigation with Dynamic Convolutional Filters. In *Proceedings of the British Machine Vision Conference*, 2019. 175

[108] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Kumar Singh, and Ming-Hsuan Yang. Diverse Image-to-Image Translation via Disentangled Representations. In *Proceedings of the European Conference on Computer Vision*, 2018. 171

[109] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *Proceedings of the European Conference on Computer Vision*, 2018. 13, 14

[110] Guanbin Li and Yizhou Yu. Visual Saliency Based on Multiscale Deep Features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 15

[111] Guang Li, Linchao Zhu, Ping Liu, and Yi Yang. Entangled Transformer for Image Captioning. In *Proceedings of the International Conference on Computer Vision*, 2019. 10, 81, 83

[112] Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. Visual Semantic Reasoning for Image-Text Matching. In *Proceedings of the International Conference on Computer Vision*, 2019. 14

[113] Yin Li, Xiaodi Hou, Christof Koch, James Rehg, and Alan Yuille. The secrets of salient object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 29, 41, 43

[114] Zhenyang Li, Kirill Gavrilyuk, Efstratios Gavves, Mihir Jain, and Cees GM Snoek. VideoLSTM Convolves, Attends and Flows for Action Recognition. *Computer Vision and Image Understanding*, 166:41–50, 2018. 17

[115] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the Annual Meeting on Association for Computational Linguistics Workshops*, 2004. 57, 75, 102, 143

[116] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *Proceedings of the European Conference on Computer Vision*, 2014. 10, 28, 48, 52, 56, 65, 66, 67, 75, 101, 136

[117] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, 2017. 171

[118] Nian Liu and Junwei Han. A Deep Spatial Contextual Long-term Recurrent Convolutional Network for Saliency Detection. *IEEE Transactions on Image Processing*, 27(7):3264–3274, 2018. 8, 39, 41, 42

[119] Nian Liu, Junwei Han, Dingwen Zhang, Shifeng Wen, and Tianming Liu. Predicting eye fixations using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 8, 42, 43

[120] Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Improved Image Captioning via Policy Gradient Optimization of SPIDEr. In *Proceedings of the International Conference on Computer Vision*, 2017. 9

[121] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. In *Proceedings of the International Conference on Computer Vision*, 2015. 173

[122] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the International Conference on Machine Learning*, 2017. 14

[123] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 10, 95

[124] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Neural Baby Talk. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 10, 11, 86, 92, 103, 106, 107

[125] Chih-Yao Ma, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira. The Regretful Agent: Heuristic-Aided Navigation through Progress Estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 175

[126] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 122, 156

[127] Vijay Mahadevan and Nuno Vasconcelos. Biologically Inspired Object Tracking Using Center-Surround Saliency Mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):541–554, 2013. 15

[128] Christopher D Manning, Christopher D Manning, and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999. 92

[129] Manuel Jesús Marín-Jiménez, Andrew Zisserman, Marcin Eichner, and Vittorio Ferrari. Detecting people looking at each other in videos. *International Journal of Computer Vision*, 106(3):282–296, 2014. 13

[130] Stefan Mathe and Cristian Sminchisescu. Actions in the Eye: Dynamic Gaze Datasets and Learnt Saliency Models for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(7):1408–1424, 2014. 9

[131] Alexander Mathews, Lexing Xie, and Xuming He. SemStyle: Learning to Generate Stylised Image Captions using Unaligned Text. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 11

[132] Alexander Patrick Mathews, Lexing Xie, and Xuming He. SentiCap: Generating Image Descriptions with Sentiments. In *Proceedings of the Conference on Artificial Intelligence*, 2016. 11

[133] Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. Learning latent permutations with gumbel-sinkhorn networks. In *Proceedings of the International Conference on Learning Representations*, 2018. 97

[134] Antoine Miech, Jean-Baptiste Alayrac, Piotr Bojanowski, Ivan Laptev, and Josef Sivic. Learning from video and text via large-scale discriminative clustering. In *Proceedings of the International Conference on Computer Vision*, 2017. 13

[135] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, 2013. 132, 139

[136] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The Role of Context for Object Detection and Semantic Segmentation in the Wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 50, 51, 52

[137] Naila Murray, Maria Vanrell, Xavier Otazu, and C Alejandro Parraga. Saliency estimation using a non-parametric low-level vision model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 8

[138] Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. Dual attention networks for multimodal reasoning and matching. In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition, 2017. 130, 136, 142

[139] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970. 97

[140] Junting Pan, Cristian Canton, Kevin McGuinness, Noel E O'Connor, Jordi Torres, Elisa Sayrol, and Xavier Giro-i Nieto. SalGAN: Visual Saliency Prediction with Generative Adversarial Networks. *arXiv preprint arXiv:1701.01081*, 2017. 8, 16, 41, 42

[141] Junting Pan, Kevin McGuinness, Elisa Sayrol, Noel O'Connor, and Xavier Giró-i Nieto. Shallow and Deep Convolutional Networks for Saliency Prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 8, 16, 35, 41, 42, 44, 45

[142] Pingbo Pan, Zhongwen Xu, Yi Yang, Fei Wu, and Yueting Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 12

[143] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 12

[144] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting on Association for Computational Linguistics*, 2002. 57, 75, 102, 143

[145] Alonso Patron-Perez, Marcin Marszalek, Ian Reid, and Andrew Zisserman. Structured learning of human interactions in TV shows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12):2441–2453, 2012. 13

[146] Marco Pedersoli, Thomas Lucas, Cordelia Schmid, and Jakob Verbeek. Areas of attention for image captioning. In *Proceedings of the International Conference on Computer Vision*, 2017. 10

[147] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2014. 77, 98, 118, 132, 139, 156

[148] Robert J Peters, Asha Iyer, Laurent Itti, and Christof Koch. Components of Bottom-Up Gaze Allocation in Natural Images. *Vision research*, 45(18):2397–2416, 2005. 30

[149] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k Entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the International Conference on Computer Vision*, 2015. 11, 100

[150] Vignesh Ramanathan, Armand Joulin, Percy Liang, and Li Fei-Fei. Linking people in videos with "their" names using coreference resolution. In *Proceedings of the European Conference on Computer Vision*, 2014. 12, 13

[151] Vasili Ramanishka, Abir Das, Jianming Zhang, and Kate Saenko. Top-down visual saliency guided by captions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 10, 62

[152] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *Proceedings of the International Conference on Learning Representations*, 2015. 9, 74, 96

[153] Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier. Collecting image annotations using Amazon's Mechanical Turk. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics Workshops*, 2010. 57

[154] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015. 75, 98, 101

[155] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 9, 74, 80, 81, 83, 96, 103, 106, 107, 135

[156] Ronald A. Rensink. The Dynamic Representation of Scenes. *Visual Cognition*, 7(1-3):17–42, 2000. 15, 49

[157] Nicolas Riche, Matthieu Duvinage, Matei Mancas, Bernard Gosselin, and Thierry Dutoit. Saliency and Human Fixations: State-of-the-Art and Study of Comparison Metrics. In *Proceedings of the International Conference on Computer Vision*, 2013. 29

[158] Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. Grounding of textual phrases in images by reconstruction. In *Proceedings of the European Conference on Computer Vision*, 2016. 11

[159] Anna Rohrbach, Marcus Rohrbach, and Bernt Schiele. The long-short story of movie description. In *Proceedings of the German Conference on Pattern Recognition*, 2015. 11, 12

[160] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for movie description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 110, 124, 125

[161] Anna Rohrbach, Marcus Rohrbach, Siyu Tang, Seong Joon Oh, and Bernt Schiele. Generating Descriptions with Grounded and Co-Referenced People. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 13

[162] Dmitry Rudoy, Dan B Goldman, Eli Shechtman, and Lihi Zelnik-Manor. Learning video saliency from human gaze using candidate selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 9

[163] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 32, 57, 137

[164] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 113, 120

[165] Vidya Setlur, Saeko Takagi, Ramesh Raskar, Michael Gleicher, and Bruce Gooch. Automatic Image Retargeting. In *Proceedings of the International Conference on Mobile and Ubiquitous Multimedia*, 2005. 15

[166] Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. Cycle-Consistency for Robust Visual Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 129

[167] Rakshith Shetty, Marcus Rohrbach, Lisa Anne Hendricks, Mario Fritz, and Bernt Schiele. Speaking the Same Language: Matching Machine to Human Captions by Adversarial Training. In *Proceedings of the International Conference on Computer Vision*, 2017. 11, 63

[168] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proceedings of the International Conference on Learning Representations*, 2015. 8, 18, 26, 137, 138, 155

[169] Josef Sivic, Mark Everingham, and Andrew Zisserman. "Who are you?" Learning person specific classifiers from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 12

[170] Richard Socher and Li Fei-Fei. Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 9

[171] Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association of Computational Linguistics*, 2(1):207–218, 2014. 119

[172] Yale Song and Mohammad Soleymani. Polysemous visual-semantic embedding for cross-modal retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 68

[173] Yusuke Sugano and Andreas Bulling. Seeing with humans: Gaze-assisted neural image captioning. *arXiv preprint arXiv:1608.05203*, 2016. 10, 15, 49, 62

[174] Sainbayar Sukhbaatar, Edouard Grave, Guillaume Lample, Herve Jegou, and Armand Joulin. Augmenting Self-attention with Persistent Memory. *arXiv preprint arXiv:1907.01470*, 2019. 10

[175] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the International Conference on Computer Vision*, 2019. 68

[176] Meijun Sun, Ziqi Zhou, Qinghua Hu, Zheng Wang, and Jianmin Jiang. SG-FCN: A Motion and Memory-Based Deep Learning Model for Video Saliency Detection. *IEEE Transactions on Cybernetics*, 49(8):2900–2911, 2018. 9

[177] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the International Conference on Machine Learning*, 2017. 82

[178] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the International Conference on Machine Learning*, 2013. 58

[179] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper with Convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 8

[180] Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. "Knock! Knock! Who is it?" probabilistic person identification in TV-series. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 12

[181] Benjamin W Tatler. The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions. *Journal of Vision*, 7(14):4–4, 2007. 17, 19

[182] Hamed R Tavakoli, Ali Borji, Jorma Laaksonen, and Esa Rahtu. Exploiting inter-image similarity and ensemble of extreme learners for fixation prediction using deep features. *Neurocomputing*, 244:10–18, 2017. 8, 42, 62

[183] Hamed R Tavakoli, Rakshith Shetty, Ali Borji, and Jorma Laaksonen. Paying attention to descriptions generated by image captioning models. In *Proceedings of the International Conference on Computer Vision*, 2017. 10, 11, 15, 49, 62

[184] Tijmen Tieleman and Geoffrey Hinton. RMSProp: Divide the gradient by a running average of its recent magnitude. *Coursera Course: Neural Networks for Machine Learning*, 2012. 32

[185] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of CNN activations. In *Proceedings of the International Conference on Learning Representations*, 2016. 156

[186] Atousa Torabi, Christopher Pal, Hugo Larochelle, and Aaron Courville. Using descriptive video services to create a large data source for video annotation research. *arXiv preprint arXiv:1503.01070*, 2015. 110, 111

[187] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the International Conference on Computer Vision*, 2015. 118

[188] Anne M Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136, 1980. 7

[189] Yao-Hung Hubert Tsai, Liang-Kang Huang, and Ruslan Salakhutdinov. Learning Robust Visual-Semantic Embeddings. In *Proceedings of the International Conference on Computer Vision*, 2017. 152

[190] Po-He Tseng, Ran Carmi, Ian GM Cameron, Douglas P Munoz, and Laurent Itti. Quantifying center bias of observers in free viewing of dynamic natural scenes. *Journal of Vision*, 9(7):4–4, 2009. 17, 19

[191] Laurens Van Der Maaten. Accelerating t-SNE using tree-based algorithms. *Journal of Machine Learning Research*, 15(1):3221–3245, 2014. 122

[192] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, 2017. 10, 16, 68, 70, 75, 76, 78, 175

[193] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based Image Description Evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 56, 57, 74, 75, 97, 102, 143

[194] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph Attention Networks. In *Proceedings of the International Conference on Learning Representations*, 2018. 70

[195] Subhashini Venugopalan, Lisa Anne Hendricks, Raymond Mooney, and Kate Saenko. Improving lstm-based video description with linguistic knowledge mined from text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2016. 12

[196] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *Proceedings of the International Conference on Computer Vision*, 2015. 11, 12

[197] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2014. 12

[198] Paul Vicol, Makarand Tapaswi, Lluis Castrejon, and Sanja Fidler. MovieGraphs: Towards Understanding Human-Centric Situations from Videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 13

[199] Eleonora Vig, Michael Dorr, and David Cox. Large-scale optimization of hierarchical features for saliency prediction in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 8, 16, 17, 20, 42, 43

[200] Ashwin K Vijayakumar, Michael Cogswell, Ramprasaath R Selvaraju, Qing Sun, Stefan Lee, David J Crandall, and Dhruv Batra. Diverse Beam Search for Improved Description of Complex Scenes. In *Proceedings of the Conference on Artificial Intelligence*, 2018. 11

[201] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 47

[202] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and Tell: Lessons Learned from the 2015 MSCOCO Image Captioning Challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):652–663, 2017. 9, 47, 91

[203] Dirk Walther, Laurent Itti, Maximilian Riesenhuber, Tomaso Poggio, and Christof Koch. Attentional selection for object recognition - a gentle way. In *International Workshop on Biologically Motivated Computer Vision*, 2002. 15

[204] Josiah Wang, Pranava Madhyastha, and Lucia Specia. Object Counts! Bringing Explicit Detections Back into Image Captioning. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2018. 68

[205] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning Two-Branch Neural Networks for Image-Text Matching Tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):394–407, 2019. 13, 130, 136, 142, 143

[206] Liwei Wang, Alexander Schwing, and Svetlana Lazebnik. Diverse and accurate image description using a variational auto-encoder with an additive gaussian encoding space. In *Advances in Neural Information Processing Systems*, 2017. 11, 105

[207] Wenguan Wang and Jianbing Shen. Deep Visual Attention Prediction. *IEEE Transactions on Image Processing*, 27(5):2368–2378, 2018. 42, 43

[208] Wenguan Wang, Jianbing Shen, Fang Guo, Ming-Ming Cheng, and Ali Borji. Revisiting video saliency: A large-scale benchmark and a new model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 9

[209] Wenguan Wang, Jianbing Shen, and Fatih Porikli. Saliency-aware geodesic video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 9

[210] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced Cross-Modal Matching and Self-Supervised Imitation Learning for Vision-Language Navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 175

[211] Xin Wang, Wenhan Xiong, Hongmin Wang, and William Yang Wang. Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation. In *Proceedings of the European Conference on Computer Vision*, 2018. 129

[212] Yaxiong Wang, Hao Yang, Xueming Qian, Lin Ma, Jing Lu, Biao Li, and Xin Fan. Position Focused Attention Network for Image-Text Matching. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, 2019. 14

[213] J. H. Jr Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963. 114

[214] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. Sampling matters in deep embedding learning. In *Proceedings of the International Conference on Computer Vision*, 2017. 133

[215] Qi Wu, Peng Wang, Chunhua Shen, Anton van den Hengel, and Anthony Dick. Ask Me Anything: Free-form Visual Question Answering Based on Knowledge from External Sources. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 22, 129

[216] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *Proceedings of the International Conference on Machine Learning*, 2015. 10, 16, 17, 49, 54, 55, 58, 59, 60

[217] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *Proceedings of the International Conference on Machine Learning*, 2015. 47

[218] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 14

[219] Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 152

[220] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical Saliency Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 15

[221] Chuan Yang, Lihe Zhang, Ruan Xiang Lu, Huchuan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 29, 41, 43

[222] Min Yang, Wei Zhao, Wei Xu, Yabing Feng, Zhou Zhao, Xiaojun Chen, and Kai Lei. Multitask learning for cross-domain image captioning. *IEEE Transactions on Multimedia*, 21(4):1047–1061, 2018. 14

[223] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-Encoding Scene Graphs for Image Captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 10, 80, 81, 83

[224] Benjamin Z Yao, Xiong Yang, Liang Lin, Mun Wai Lee, and Song-Chun Zhu. I2t: Image parsing to text description. *Proceedings of the IEEE*, 98(8):1485–1508, 2010. 9

[225] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *Proceedings of the International Conference on Computer Vision*, 2015. 12

[226] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring Visual Relationship for Image Captioning. In *Proceedings of the European Conference on Computer Vision*, 2018. 10, 80, 81, 83

[227] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Hierarchy Parsing for Image Captioning. In *Proceedings of the International Conference on Computer Vision*, 2019. 80, 81, 83

[228] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 10

[229] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. 56, 100, 136

[230] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *Proceedings of the International Conference on Learning Representations*, 2016. 26

[231] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 12

[232] Kiwon Yun, Yifan Peng, Dimitris Samaras, Gregory J Zelinsky, and Tamara L Berg. Exploring the role of gaze behavior and object detection in scene understanding. *Frontiers in Psychology*, 4:917, 2013. 11

[233] Kiwon Yun, Yifan Peng, Dimitris Samaras, Gregory J Zelinsky, and Tamara L Berg. Studying relationships between human gaze, description, and computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 11

[234] Yun Zhai and Mubarak Shah. Visual Attention Detection in Video Sequences Using Spatiotemporal Cues. In *Proceedings of the ACM International Conference on Multimedia*, 2006. 8

[235] Jianming Zhang and Stan Sclaroff. Saliency detection: A boolean map approach. In *Proceedings of the International Conference on Computer Vision*, 2013. 8, 42

[236] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 112

[237] Yundong Zhang, Juan Carlos Niebles, and Alvaro Soto. Interpretable Visual Question Answering by Visual Grounding from Attention Supervision Mining. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2018. 68

[238] Qi Zhao and Christof Koch. Learning a Saliency Map using Fixated Locations in Natural Scenes. *Journal of Vision*, 11(3):9–9, 2011. 8, 40

[239] Sheng-hua Zhong, Yan Liu, Feifei Ren, Jinghuan Zhang, and Tongwei Ren. Video saliency detection via dynamic consistent spatio-temporal attention modelling. In *Proceedings of the Conference on Artificial Intelligence*, 2013. 8, 9

[240] Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J Corso, and Marcus Rohrbach. Grounded video description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 110

[241] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the International Conference on Computer Vision*, 2017. 170, 171

[242] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 119