Multimedia Tools and Applications manuscript No.
(will be inserted by the editor)

Augmenting data with GANs to segment melanoma skin lesions

Federico Pollastri · Federico Bolelli · Roberto Paredes · Costantino Grana

Received: date / Accepted: date

Abstract This paper presents a novel strategy that employs Generative Adversarial Networks (GANs) to augment data in the skin lesion segmentation task, which is a fundamental first step in the automated melanoma detection process. The proposed framework generates both skin lesion images and their segmentation masks, making the data augmentation process extremely straightforward. In order to thoroughly analyze how the quality and diversity of synthetic images impact the efficiency of the method, we remodel two different well known GANs: a Deep Convolutional GAN (DCGAN) and a Laplacian GAN (LAPGAN). Experimental results reveal that, by introducing such kind of synthetic data into the training process, the overall accuracy of a state-of-the-art Convolutional/Deconvolutional Neural Network for melanoma skin lesion segmentation is increased.

Keywords Deep Learning · Convolutional Neural Networks · Adversarial Learning · Skin Lesion Segmentation

1 Introduction

Malignant melanoma is the most dangerous type of skin cancer, with a substantial death rate. It can be cured with prompt excision if detected in the early stage, making fast diagnosis extremely important [6]. However, early detection is very hard to obtain, and even in specialized centers the accuracy of

Federico Pollastri, Federico Bolelli (✉), Costantino Grana
DIEF, Università degli Studi di Modena e Reggio Emilia, Italy
Tel.: +39-059-2056265
Fax: +39-059-2056129
E-mail: {name.surname}@unimore.it

Roberto Paredes
PRHLT Research Center, Universitat Politécnica de Valencia, Spain
E-mail: rparedes@dsic.upv.es

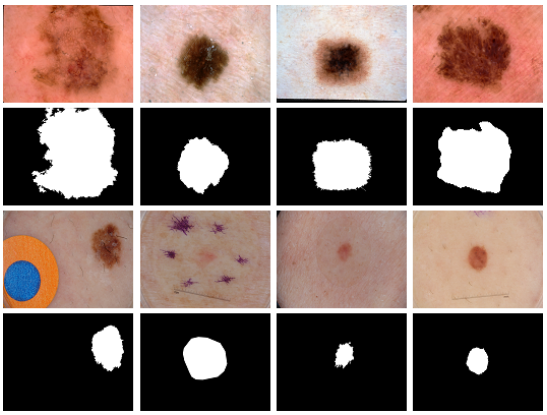


Fig. 1 Samples from the ISIC dataset: dermoscopic skin images coupled with their ground truth segmentation mask.

the clinical diagnosis for melanoma, achieved with the unaided eye, is slightly better than 60% [12]. A great tool to improve the clinical decision making can be automated analysis through dermoscopic images, which are obtained by a non-invasive in vivo examination with a microscope, exploiting incident light and oil/gel immersion to make subsurface structures of the skin accessible to visual examination.

This is why the International Skin Imaging Collaboration (ISIC) has begun to aggregate a large-scale, publicly accessible dataset of dermoscopic images (Fig. 1). This dataset enabled the development of the first public benchmark challenge on dermoscopic image analysis in 2016. The goal of the challenge was to provide a fixed dataset snapshot to support development of automated melanoma diagnosis algorithms across three tasks of lesion analysis: segmentation, dermoscopic feature detection, and classification. In 2017, ISIC hosted the second edition of this challenge, providing an expanded dataset [7].

The segmentation task can be especially complicated due to the great variety of characteristics shown in skin lesions and skin itself among different people. The definition of skin lesion borders is also very subjective. Moreover, manual image segmentation is a very time consuming job that requires the work of a competent specialist. State-of-the-art algorithms for lesion segmentation are based on supervised machine learning techniques, it is then crucial to find a cheaper way to obtain segmented images useful to train the models.

Therefore, the focus of this paper is the "data augmentation" process of a Convolutional-Deconvolutional Neural Network (CDNN), designed to automatically map dermoscopic images into lesion segmentation masks. In particular, we propose a framework to generate both skin lesion images and their segmentation masks by means of a Generative Adversarial Network (GAN). Differently from the usual approach, in which GANs are used to generate unlabeled images, tricky to include in the training process, the ability to reproduce image-mask couples makes the data augmentation process extremely straight-

forward [22]. Indeed, data generated in this way can be used to perform an initialization of deep networks, always crucial in deep learning [20]. After some epochs, original samples are used to fine-tune the network parameters.

Experimental results show that adding GAN-generated data in the training process effectively improves the segmentation accuracy of state-of-the-art CDNNs.

The rest of the paper is organized as follows: in Section 2 a brief review of the learning strategies exploited in our work is reported. Section 3 summarizes two different GANs employed to generate synthetic images, whereas Section 4 focuses on the automated segmentation model that will benefit from said generated data. The training process and experimental results of our CDNNs are presented in Section 5. In this Section, the effectiveness of the GAN-generated data on a shrunk dataset are also explored. Finally, in Section 6 conclusions are drawn.

2 Related Work

Deep learning algorithms, Convolutional Neural Networks (CNNs) in particular, have become the methodology of choice for analyzing medical images. One of the main reasons is most certainly their ability to extract features on their own [15], as opposed to many previous approaches based on hand-crafted features [3,16,17]. In classical Convolutional Neural Networks, convolutions and non-linearities are interleaved with pooling operations. The purpose of these subsampling units is to enlarge the convolved receptive field and to decrease the feature maps size. These two effects are respectively important to obtain information from wider areas for the final prediction, and to augment the number of feature maps without the memory requirements becoming excessive. Unfortunately, resolution is a key element in segmentation applications, where every pixel is expected to be assigned to a specific class. To get a prediction with the resolution of the input image, [18] proposed Fully Convolutional Networks, consisting in the addition of a deconvolution [27] part after the convolutional one. The idea is that the feature maps size is progressively decreased in the convolutional part, while upsampling operators and fractionally-strided convolutions increase it back to the input resolution. In [24] an extension of this idea, called U-Net, was proposed. The authors increased the number of feature channels in the upsampling path, yielding a u-shaped architecture, and introduced a concatenation of early extracted feature maps to the output of every upsampling layer. This particular trait of the U-Net architecture increases the ability of the model to maintain spatial information that could otherwise be lost during the contracting path.

An important operation which should precede the network training is data augmentation. This process consists in generating new data items by applying very simple transformations to existing training samples [13]. Data augmentation is very often exploited in deep learning to artificially enlarge a dataset without needing new data and, most importantly, without compromising the

consistency of the training set. In skin lesion segmentation, a huge variety of data augmentation steps can be applied to every single image, including random rotation, flipping, shifting and scaling [26].

Instead of relaying on manually defined transformations, it would be nice to learn how to augment data in the same deep-learning framework. Generative Adversarial Networks [9] are a model made up by two different networks: a generator and a discriminator. The task of the discriminator is learning to distinguish real samples from generated ones, while the generator tries to generate good enough samples to fool the discriminator. The notable effect of this architecture is that the two networks improve together, learning at the same time while competing. Neff *et al.* [21], for instance, employed GANs to generate annotated images in order to simply mix them with the original data, overlooking the influence of the generated samples' quality. Moreover, the work by [2] explores the possibility of exploiting GANs to generate skin lesion images. That paper focuses on the generation of skin lesion images, tackling the heavy class imbalance that afflicts the classification task, without including any segmentation mask.

3 Data Augmentation with GANs

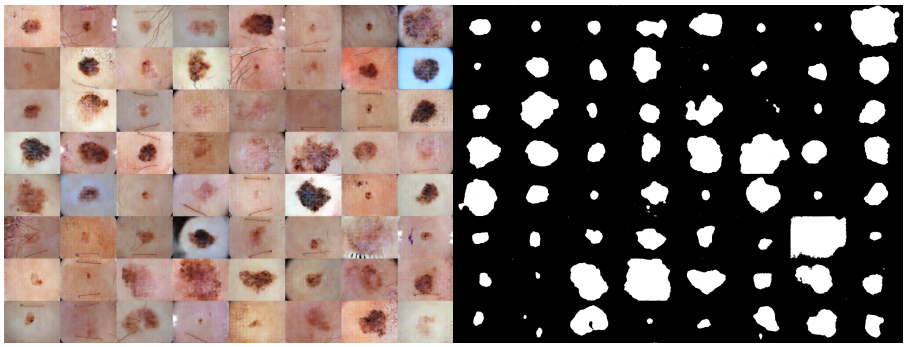
The segmentation task could greatly benefit from a bigger and wider dataset, this is why we approach the problem focusing on the data augmentation process. GANs are often used to create unlabeled examples [25, 28, 1], which cannot be directly employed for the training of a supervised algorithm. We improve the role of the GAN in the training process by implementing an architecture which generates both the image and its segmentation mask, making it extremely easy to exploit new synthetic images as additional training data. In the proposed GANs, both the input and the output are 4 channel images: the first three channels are the R, G and B components and the fourth one is the binary segmentation mask. This approach provides a tool for generating new skin lesion images coupled with a coherently learned segmentation mask. In order to train our GANs, we choose to remove the 118 non-dermoscopic images from the 2017 ISIC challenge training set, given their very different nature. This approach sets the final training dataset size to 1882 images.

3.1 DCGAN

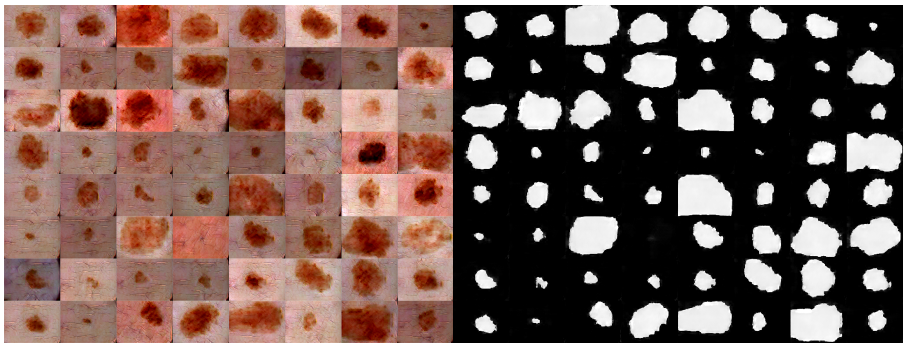
We modify the DCGAN proposed in [23] in order to use the described 4D input/output data in both the generator and the discriminator.

The two networks exploit four convolutional/deconvolutional layers, with a kernel size of 5×5, and a stride of two. Each layer, but the last two, is followed by a Leaky Rectified Linear Unit, LReLU [19], and batch normalization [10]. A sigmoid activation function is added after the fully-connected layer at the end of the discriminator to perform the binary classification (real or fake). A

hyperbolic tangent serves as the activation function of the generator. After the training process is completed, our generator is able to employ 100-dimensional random vectors to create 192×256 RGB synthetic skin lesion images and their 192×256 binary segmentation masks (Fig. 2a and Fig. 2b).



(a) (b)



(c) (d)

Fig. 2 DCGAN-generated skin lesion samples (a) and their segmentation masks (b). LAPGAN-generated skin lesion samples (c) and their segmentation masks (d).

3.2 LAPGAN

We implement a 5 levels Laplacian Generative Adversarial Network (LAPGAN) by designing and training 5 different, independent GANs [8]. The network representing the first level of the laplacian pyramid generates a 6×8 pixels dermoscopic image from a 100-dimensional random vector. In this case, both the generator and the discriminator are formed by two fully connected layers. The following pyramid levels have a different task: they learn how to improve the resolution of upsampled and blurred images, and each one focuses on images with a different fixed size. The generators of the four last GANs are

three layers CNN, whereas discriminators are composed by two convolutional layers and a fully connected one.

LReLU activation functions serve after each intermediate layer, while the output of the discriminators is affected by a sigmoid function. No activation function is added to the last layer of the generators.

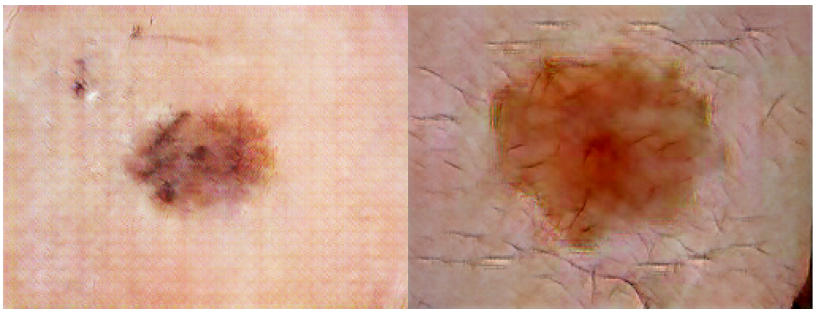
Starting from the 6×8 pixels data, each image is upsampled and fed into the next pyramid level, following the sampling process illustrated in the appendix. An extra iteration with the last GAN is implemented to obtain 192×256 images and their segmentation masks (Fig. 2c and Fig. 2d).

3.3 Samples Quality Discussion

Both the DCGAN and the LAPGAN are trained to generate 192×256 images and segmentation masks.

The DCGAN benefits from its end-to-end training process, learning to produce visually pleasing skin lesion samples, with good diversity. Most images present realistic details, like the well delivered presence of hair, black corner representing real camera characteristics or pen marks. The images, however, suffer from a heavy checkerboard effect caused by the fractional strided convolution layers implemented in the framework (Fig. 3a).

The LAPGAN is able to get rid of the checkerboard effect as a result of the absence of previously described layers. On the other hand, this model introduces noisy lesion borders and artifacts such as unnatural square green patches and low quality hair (Fig. 3b). Moreover, images generated by the LAPGAN look much similar one to another, and the particular nature of a LAPGAN makes the training process extremely hard. The five different required networks need constant supervision and hyperparameters adjustment, making good results very painful to obtain, especially when compared to the DCGAN.



(a) (b)

Fig. 3 (a) detail image of the DCGAN output, and (b) detail image of the LAPGAN output.

Overall, both models produce segmentation masks that look very coherent with the respective generated images. We thus proceed to evaluate the impact of the two frameworks on the accuracy of lesion segmentation architectures, when they are included in the training process.

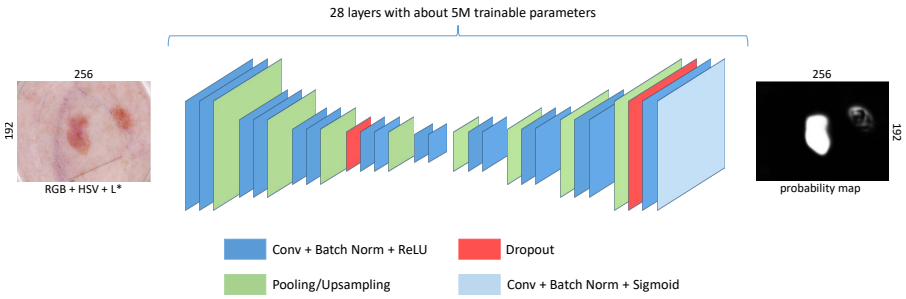


Fig. 4 Baseline CDNN model.

4 Dermoscopic Image Segmentation

4.1 Baseline Architecture

In order to avoid trivial comparisons, we select and reimplement the baseline CDNN from the architecture that obtained the highest score in 2017 ISIC challenge [26]. This network maps the input dermoscopic image to a posterior probability map. The network contains 28 layers with about 5M trainable parameters. The stride is fixed at 1 and Rectified Linear Units are used as activation functions for each convolutional/deconvolutional layer but the output, which implements a sigmoid.

As in the original proposal, every image is resized to 192×256 and the original RGB channels are augmented with both the HSV and L* channels. In accordance with the original paper, data augmentation is obtained by randomly flipping, rotating, shifting, scaling and changing color contrasts in original images and ground truths.

The loss function designed by [26] is:

$$L = 1 - \frac{\sum_{i,j} t_{ij} p_{ij}}{\sum_{i,j} t_{ij}^2 + \sum_{i,j} p_{ij}^2 - \sum_{i,j} t_{ij} p_{ij}}, \tag{1}$$

where t_{ij} is the target value of the pixel at coordinates (i, j), and p_{ij} is the real output. Note that t_{ij} is either 0 or 1, while p_{ij} is a real number in range $[0, 1]$.

The authors state that "a bagging-type ensemble strategy is implemented to combine outputs of 6 CDNNs", without further details.

Ensemble methods require different techniques to be combined and used together, so we choose to introduce variations in the original CDNN (Fig. 4) by studying the main hyperparameters of the model. The final purpose is to obtain many networks different from one another, and then combine their output through a simple ensemble method. These networks, named CDNN_0 through CDNN_6, are summarized in Table 1 and described in the following section.

Table 1 Hyperparameters of the Neural Networks analyzed.

NN	**Images Dimension**	**Images Channels**	**Loss**
CDNN_0	192×256	7	Eq. 1
CDNN_1	192×256	3	Eq. 1
CDNN_2	192×256	9	Eq. 1
CDNN_3	96×128	7	Eq. 1
CDNN_4	384×512	7	Eq. 1
CDNN_5	192×256	7	Eq. 3
CDNN_6	192×256	7	Eq. 4

4.2 Hyperparameters Analysis

The baseline architecture (CDNN_0) is mainly affected by three hyperparameters: input image size, color channels and loss function. We describe a set of possible variations over these parameters (Table 1), in order to stress how they influence the final performance results.

Image Channels. In the baseline model both HSV and L* channels were added. These channels are obtained by means of non-linear transformations. To address the question whether the model is able to independently learn these transformations autonomously, we train two networks: one with just the three RGB channels (CDNN_1) and one with every channel from RGB, HSV and CIELAB spaces (CDNN_2).

Resizing Dimensions. We train two additional networks with input images resized to 96×128 and 384×512 (CDNN_3 and CDNN_4), which are respectively half and double the original size along each axis. Since we do not change the stride and the size of convolutional filters, the scaling factor between layers remains the same. This provides the two networks with a different encoded representation size.

Loss Function. It is interesting to point out the common misconception displayed in the work by [26]. The distance measure in Eq. 1 is said to be "based on the Jaccard distance". In fact, Eq. 1 is the Tanimoto distance, which is a proper distance when both vectors have only positive elements, and it is equal to the Jaccard distance only with binary vectors [14]. In our case,

only the target is binary, but the prediction is a real value between 0 and 1. The correct (generalized) Jaccard distance on real positive vectors is defined as:

$$d_J = 1 - \frac{\sum_{i,j} \min(t_{ij}, p_{ij})}{\sum_{i,j} \max(t_{ij}, p_{ij})}. \tag{2}$$

Since t_{ij} is still binary, Eq. 2 can also be computed as:

$$d_J = 1 - \frac{\sum_{i,j} t_{ij} p_{ij}}{\sum_{i,j} t_{ij} + \sum_{i,j} p_{ij} - \sum_{i,j} t_{ij} p_{ij}}, \tag{3}$$

which may be the reason for the common confusion (note the missing squares with respect to Eq. 1).

We thus train two more variations of the model. The first one using the proper Jaccard distance and the other one using the mean squared error function (CDNN_5 and CDNN_6):

$$MSE = \frac{1}{n} \sum_{i,j} (t_{ij} - p_{ij})^2 \tag{4}$$

where n is the total number of pixels.

5 Experimental Results

In order to employ GAN-generated images discussed in Section 3, we design a two-step training process. In the first phase, each CDNN is trained with synthetic data for a total of 400 epochs [1]. Instead of generating a fixed number of samples, GANs are required to provide new couples image-mask for each training batch. In the second step, the network is fed with real data, for a total of 100 epochs (Fig. 5).

This learning process proved itself to be the best way to exploit such kind of GAN-generated data: further experiments show that mixing real and synthetic images in a single training process reduce the accuracy of 0.020 on average.

In order to prove the effectiveness of the GANs on the final accuracy, we also train a version of each CDNN using only real data, for a total of 400 epochs.

Following the approach introduced in Section 3, we train each CDNN with the 1882 dermoscopic images of the 2017 ISIC challenge training set, choosing to remove the 118 non-dermoscopic ones. The batch size is set to 16 and the learning rate is fixed as 0.0003 at the beginning of each training process and it is then affected by the Adam optimizer [11]. Given the importance of data augmentation, both real and synthetic images are randomly rotated, flipped, shifted, and scaled. The color contrast is also changed.

[1] An epoch is the number of batches needed to feed the network with 1882 images.

Table 2 Comparison of the experimental results on $CDNN_0$ through $CDNN_6$, when trained exploiting different augmentation strategies. Bold values represent the best performing approaches for the networks included in the ensemble method described in Section 5.2.

NN	Original Data	DCGAN Augmented	LAPGAN Augmented	Both GANs Augmented
$CDNN_0$	0.746	**0.751**	0.739	0.742
$CDNN_1$	0.759	0.753	**0.764**	0.752
$CDNN_2$	0.746	0.756	**0.762**	0.750
$CDNN_3$	0.755	0.762	**0.764**	0.756
$CDNN_4$	0.717	—	—	—
$CDNN_5$	0.739	0.752	0.740	0.747
$CDNN_6$	0.731	0.749	**0.756**	0.751

Experimental results are summed up in Table 2. The first column identifies each network with a reference to Table 1, whereas every other one reports the Jaccard Index[2] obtained on the 2017 ISIC challenge public test set.

The column *Original Data* shows the accuracy of the networks trained with only real dermoscopic images. Most of the variations explored with the hyper-parameters analysis obtain results close to our baseline network ($CDNN_0$). The only case where an actual difference is noticeable is the $CDNN_4$ network, the one with larger images, which obtains a Jaccard index of 0.717. Because of its low performance, it is removed from further analysis. The last three columns show the results obtained after including the two GANs in the training process. The heavy checkerboard effect, noticeable in samples generated using a DCGAN, has a negative impact on the model accuracy.

[2] The official accuracy measure of the ISIC challenge [7].

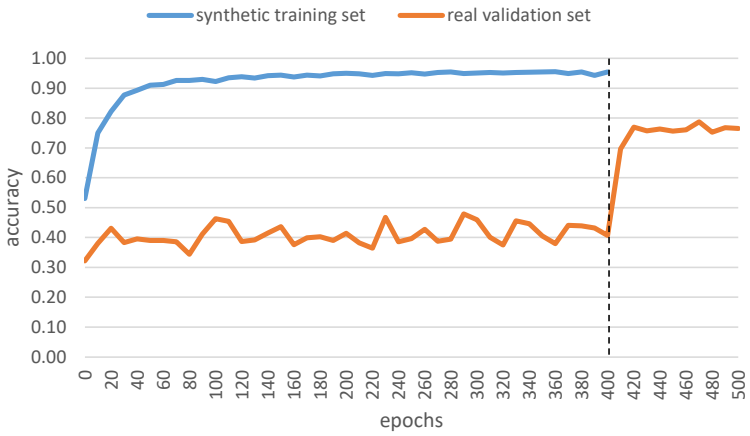


Fig. 5 Training process of the $CDNN_0$ exploiting both synthetic (before the dotted line) and real data (after the dotted line). The blue line is the accuracy obtained on the synthetic training set, whereas the orange line identifies the accuracy obtained on the real validation set.

Images generated through the LAPGAN get the best response from the architecture, despite the low visual quality and variability offered. Finally, the last column of Table 2 displays that alternating the two GANs during the first training phase has almost no positive outcome.

5.1 About Flexibility

In order to further analyze the proposed augmentation method, we choose to investigate the effect of GAN-generated samples on the training process of U-Net [24], a CDNN specifically developed for the segmentation of bio-medical images. Given the ability of this architecture to achieve good performances with only few annotated images available for training, U-Net is considered to be an excellent baseline for many medical image segmentation tasks.

The standard U-Net loss is the Cross Entropy:

$$CE = -\sum_{i,j} t_{i,j} \log(p_{i,j}) \tag{5}$$

which, when the network is outlined to predict only two classes (background and foreground), can be also defined as Binary Cross Entropy:

$$BCE = -\sum_{i,j} t_{i,j} \log(p_{i,j}) + (1 - t_{i,j}) \log(1 - p_{i,j}) \tag{6}$$

Table 3 summarizes the results obtained by using U-Net with two different losses: U-Net standard loss (Eq. 6) and our baseline loss (Eq. 1), validating the superiority of LAPGAN-generated samples highlighted in Section 5, and proving the Binary Cross Entropy to be the best performing loss for U-Net.

Table 3 Comparison of the experimental results on U-Net, when trained exploiting different augmentation strategies. Bold values represent the best performing approaches.

Loss	**Original Data**	**DCGAN Augmented**	**LAPGAN Augmented**	**Both GANs Augmented**
Eq. 6	0.753	0.755	**0.767**	0.764
Eq. 1	0.744	0.755	**0.762**	0.760

5.2 Overall Accuracy

For each pixel, the probability of it being part of the skin lesion is obtained as the mean value across the selected CDNNs (*i.e.* all the networks presented in Section 4.2 but CDNN_4). The output is then binarized with a dual-threshold method. A high threshold (0.8) is followed by Connected Components Labeling [4,5] and the biggest object center is assumed to be the tumor center. Afterwards, a lower threshold (0.4) is applied and the final segmentation mask

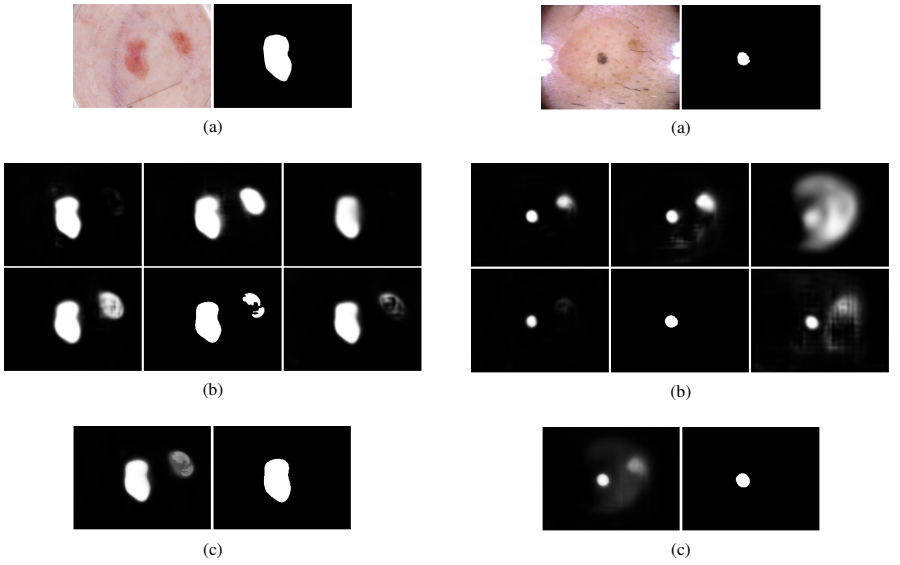


Fig. 6 (a) Input images and their ground truths. (b) The output prediction of our CDNNs. Top row shows, from left to right, the output of $CDNN_1$, $CDNN_2$ and $CDNN_3$. Bottom row shows $CDNN_0$, $CDNN_5$ and $CDNN_6$. (c) Outputs ensemble before and after binarization.

is given by the region which contains the tumor center, as shown in Fig. 6. When the first high threshold does not find any object, we only apply the second one and keep its result as segmentation mask.

It is interesting to point out that, even though the accuracy of the various CDNNs is almost uniform, different networks specialize on different features, improving the segmentation accuracy on some images rather than on others (Fig. 6). This greatly increases the effectiveness of our ensemble method.

$CDNN_3$ is the only network that, even with great results obtained when trained with the original data, always benefits from the proposed data augmentation. This can be explained with the resizing process applied to every synthetic image during training, which halves image dimensions along each axis smoothing GAN-generated artifacts discussed in Section 3.3 and highlighted in Fig. 3.

The loss designed in Eq. 3 rewards more marked predictions, discouraging values too distant from both 0 and 1 in the output image. This behaviour amplifies the impact of $CDNN_5$ when it is employed in an ensemble method, thus reducing the overall accuracy. Hence, to obtain the final score, we also remove $CDNN_5$ and apply the ensemble strategy on the output results of best performing networks (*i.e.* networks corresponding to the bold values in Table 2), obtaining a final jaccard index of **0.789**, which is above the state-of-the-art accuracy of 0.782 [22] and the challenge winner result of 0.765[3].

[3] 2017 challenge scoreboard at `https://bit.ly/2yUhs3O`

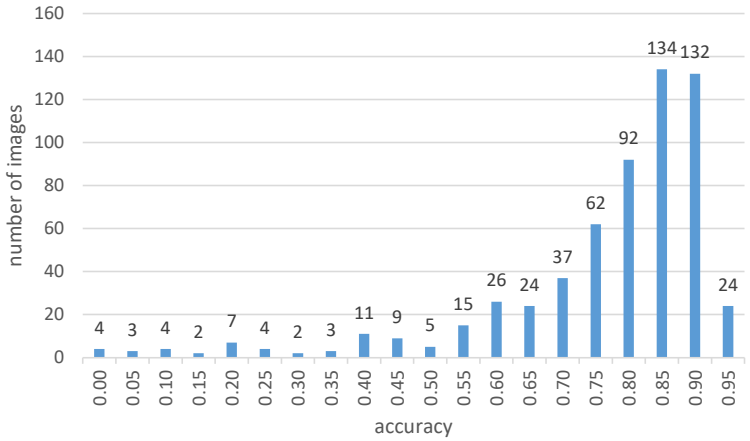


Fig. 7 Histogram of Jaccard Index values for individual images from our top performing architecture.

As stated in [7], segmentation masks that achieve a Jaccard Index of 0.8 or above tend to appear visually correct, whereas when the accuracy falls under a threshold somewhere between 0.6 and 0.7, the correctness of the segmentation can be debated. The histogram in Fig. 7 illustrates the accuracy obtained by our architecture on each skin lesion image in the test set. We obtain a Jaccard Index below 0.6 for 69 images, and below 0.7 for 119, suggesting a failure rate between 11% and 20% on the 600 samples test set.

5.3 About Scalability

The huge cost of well-annotated data for medical deep-learning-based approaches is a well known problem. The proposed method can serve as a great tool to obtain cheap medical images in many different fields.

To explore the whole effectiveness of the described framework, we extrapolate three different subsets of the ISIC dataset, respectively composed by 500, 1000 and 1500 images. For each subset, a DCGAN is trained and then employed in the two-step training process of our baseline network, CDNN_0.

Training a GAN with a small set of images is not a trivial task: quality and diversity of the generated images decrease with the size of the training set. However, given a small amount of original annotated samples, synthetic images can improve the segmentation accuracy despite their low overall quality, as shown in Fig. 8. The chart reveals that, given an early stage project with a limited dataset, one would need up to 58% more annotated images to obtain the improvement given by our proposal. Moreover, it is clear how the complexity of the problem adds a lot of relevance to every apparently small accuracy increase.

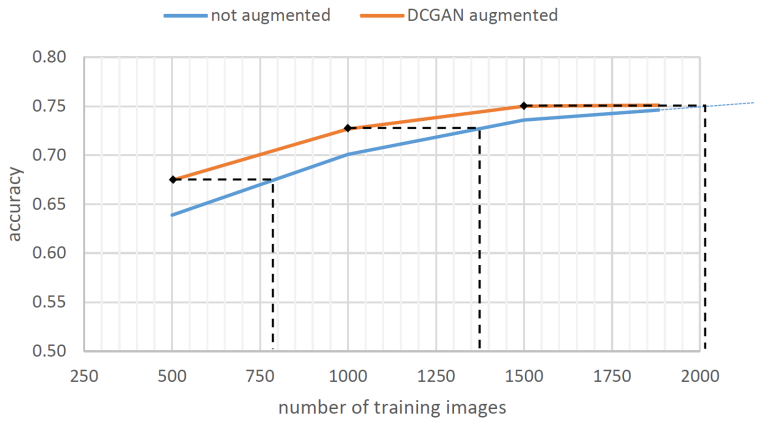


Fig. 8 Accuracy obtained on the test set by $CDNN_0$, trained employing reduced subsets of the dataset. The chart analyzes the effectiveness of the proposed augmentation process (using DCGAN) with respect to the availability of a larger dataset.

Despite the lower results obtained on $CDNN_0$ (Table 2), we extend the scalability experiment to the LAPGAN. Fig. 9 displays how, on a dataset constituted by only 500 annotated images, LAPGAN-generated samples widely outperform their DCGAN-generated counterpart. As the number of annotated images rises, so do the performances of the DCGAN, suggesting that the DCGAN actually requires to be trained with more annotated images in order to produce data useful for our cause. When evaluating the method over a 500 images dataset, a LAPGAN can be employed to obtain an accuracy boost equivalent to 138% more real annotated images.

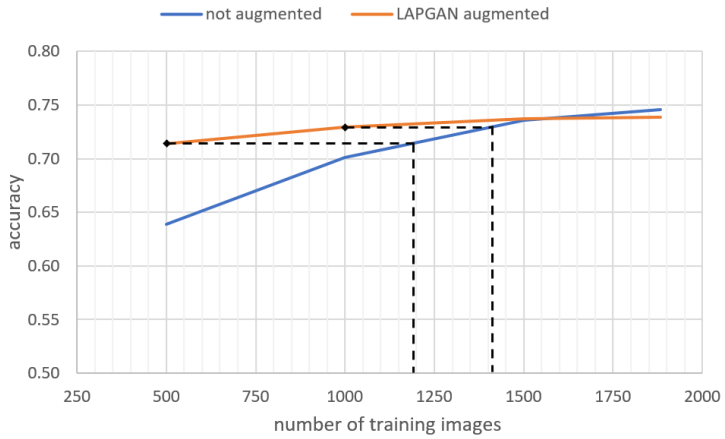


Fig. 9 Accuracy obtained on the test set by $CDNN_0$, trained employing reduced subsets of the dataset. The chart analyzes the effectiveness of the proposed augmentation process (using LAPGAN) with respect to the availability of a larger dataset.

6 Conclusion

This paper proposed a new method to exploit GANs in the data augmentation process, mitigating the need of medical manually annotated data, which are very expansive to obtain. In order to improve results on the skin lesion segmentation task, GANs are used to generate both skin lesion images and their segmentation masks, creating a tool for *automatic data augmentation* that can be integrated in any supervised learning model. An improved version of the state-of-the-art architecture for automated skin lesion segmentation has been also developed and described in the paper. We designed a two-step training process to exploit synthetic data, and compared the impact that two different types of state-of-the-art GANs can have on the final accuracy, obtaining results above the state-of-the-art. Finally, a stress test revealed the effectiveness that the proposed framework could have on similar tasks with a restricted dataset, revealing a winning trade-off between the quality of generated images and the segmentation accuracy.

References

1. Antoniou, A., Storkey, A., Edwards, H.: Data Augmentation Generative Adversarial Networks. arXiv preprint arXiv:1711.04340 (2017)
2. Baur, C., Albarqouni, S., Navab, N.: MelanoGANs: High Resolution Skin Lesion Synthesis with GANs. arXiv preprint arXiv:1804.04338 (2018)
3. Bengio, Y., Courville, A., Vincent, P.: Representation Learning: A Review and New Perspectives. IEEE transactions on pattern analysis and machine intelligence **35**(8), 1798–1828 (2013)
4. Bolelli, F., Baraldi, L., Cancilla, M., Grana, C.: Connected Components Labeling on DRAGs. In: International Conference on Pattern Recognition (2018)
5. Bolelli, F., Cancilla, M., Grana, C.: Two More Strategies to Speed Up Connected Components Labeling Algorithms. In: International Conference on Image Analysis and Processing, pp. 48–58. Springer (2017)
6. Celebi, M.E., Wen, Q., Iyatomi, H., Shimizu, K., Zhou, H., Schaefer, G.: A State-of-the-Art Survey on Lesion Border Detection in Dermoscopy Images. Dermoscopy Image Analysis pp. 97–129 (2015)
7. Codella, N.C., Gutman, D., Celebi, M.E., Helba, B., Marchetti, M.A., Dusza, S.W., Kalloo, A., Liopyris, K., Mishra, N., Kittler, H., et al.: Skin lesion analysis toward melanoma detection: A challenge at the 2017 International symposium on biomedical imaging (ISBI), hosted by the international skin imaging collaboration (ISIC). arXiv preprint arXiv:1710.05006 (2017)
8. Denton, E.L., Chintala, S., Fergus, R., et al.: Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. In: Advances in neural information processing systems, pp. 1486–1494 (2015)
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Nets. In: Advances in neural information processing systems, pp. 2672–2680 (2014)
10. Ioffe, S., Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv preprint arXiv:1502.03167 (2015)
11. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980 (2014)
12. Kittler, H., Pehamberger, H., Wolff, K., Binder, M.: Diagnostic accuracy of dermoscopy. The lancet oncology **3**(3), 159–165 (2002)

13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. In: Advances in neural information processing systems, pp. 1097–1105 (2012)
14. Lipkus, A.H.: A proof of the triangle inequality for the Tanimoto distance. Journal of Mathematical Chemistry **26**(1), 263–265 (1999). DOI 10.1023/A:1019154432472
15. Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., Van Der Laak, J.A., Van Ginneken, B., Sánchez, C.I.: A survey on deep learning in medical image analysis. Medical image analysis **42**, 60–88 (2017)
16. Liu, Y., Nie, L., Han, L., Zhang, L., Rosenblum, D.S.: Action2Activity: Recognizing Complex Activities from Sensor Data. In: Twenty-fourth international joint conference on artificial intelligence (2015)
17. Liu, Y., Nie, L., Liu, L., Rosenblum, D.S.: From action to activity: Sensor-based activity recognition. Neurocomputing **181**, 108–115 (2016)
18. Long, J., Shelhamer, E., Darrell, T.: Fully Convolutional Networks for Semantic Segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440 (2015)
19. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier Nonlinearities Improve Neural Network Acoustic Models. In: ICML Workshop on Deep Learning for Audio, Speech, and Language Processing (WDLASL 2013) (2013)
20. Mishkin, D., Matas, J.: All you need is a good init. In: International Conference on Learning Representations (ICLR) 2016 (2016)
21. Neff, T., Payer, C., Štern, D., Urschler, M.: Generative Adversarial Network based Synthesis for Supervised Medical Image Segmentation. In: OAGM & ARW Joint Workshop 2017 on "Vision, Automation & Robotics". Verlag der Technischen Universität Graz (2017)
22. Pollastri, F., Bolelli, F., Grana, C.: Improving Skin Lesion Segmentation with Generative Adversarial Networks. In: 31st International Symposium on Computer-Based Medical Systems (2018)
23. Radford, A., Metz, L., Chintala, S.: Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv preprint arXiv:1511.06434 (2015)
24. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional Networks for Biomedical Image Segmentation. In: International Conference on Medical image computing and computer-assisted intervention, pp. 234–241. Springer (2015)
25. Springenberg, J.T.: Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks. arXiv preprint arXiv:1511.06390 (2015)
26. Yuan, Y., Chao, M., Lo, Y.C.: Automatic skin lesion segmentation with fully convolutional-deconvolutional networks. arXiv preprint arXiv:1703.05165 (2017)
27. Zeiler, M.D., Taylor, G.W., Fergus, R.: Adaptive Deconvolutional Networks for Mid and High Level Feature Learning. In: Computer Vision (ICCV), 2011 IEEE International Conference on, pp. 2018–2025. IEEE (2011)
28. Zheng, Z., Zheng, L., Yang, Y.: Unlabeled Samples Generated by GAN Improve the Person Re-Identification Baseline in Vitro. arXiv preprint arXiv:1701.07717 **3** (2017)

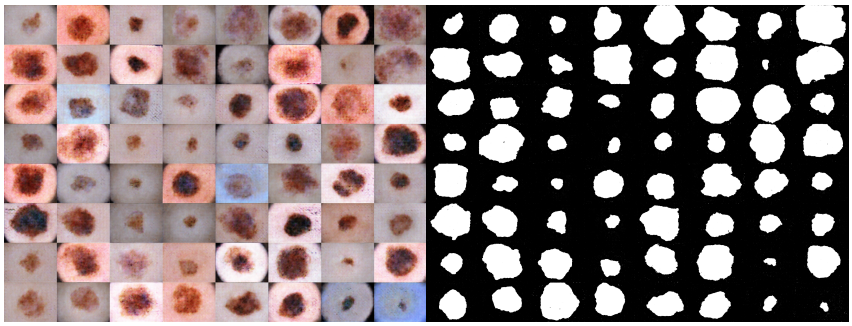
Appendix

A large dataset is a crucial asset for any GAN training process: additional images allow the network to learn how to generate more realistic samples, with realistic details and less similar one another. Fig. 10, Fig. 11, and Fig. 12 show how enlarging the training dataset improves the output results of the DCGAN. Increasing the ammount of samples from 500 to 1000, the DCGAN becomes able to provide a wider variety of skin lesions, with different sizes and shapes and with various textures. After reaching 1500 training images, the framework delivers high-resolution hair in samples that present much fewer noisy artifacts.

The sampling process of the LAPGAN is further examined in Fig. 13. We merge five independent GANs to form one LAPGAN, which is divided into six different pyramid levels. In the first level, the GAN named G_0 transforms a noise vector Z_0 into a 6×8 pixels skin lesion sample coupled with its segmentation mask, employing fully-connected layers for both the generator and the discriminator.

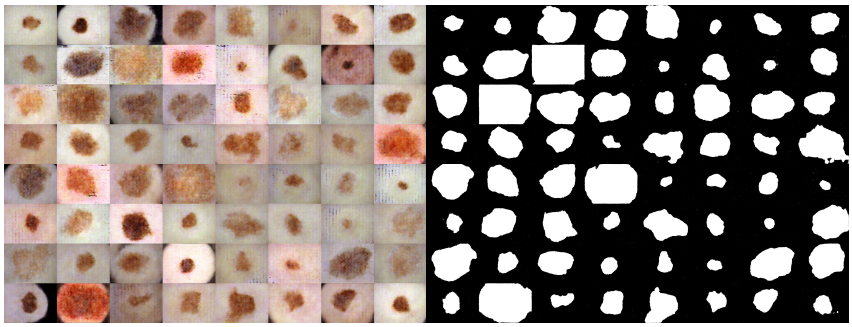
In the next pyramid level, the two outputs of G_0 are upsampled and fed, together with a new source of noise, to G_1, a GAN that exploits convolutional layers in both of its two subnetworks. The output of G_1 are two residual images (skin lesion and segmentation mask) to be added to the expanded low resolution samples, provided by the previous pyramid level. This approach allows to enlarge images generated by the previous layer without lowering the resolution. Each following level has the same structure as the one employing G_1. However, G_4 is used to provide residual images for both of the two last pyramid levels.

Fig. 13 illustrates how G_3 adds noise in each sample, generating hair poorly, whereas G_4 does a great job improving the resolution and the realism of every image. As the image dimensions grow, target residual images of adjacent pyramid levels become more similar one another, allowing us to exploit the same GAN in more than one layer of the pyramid.



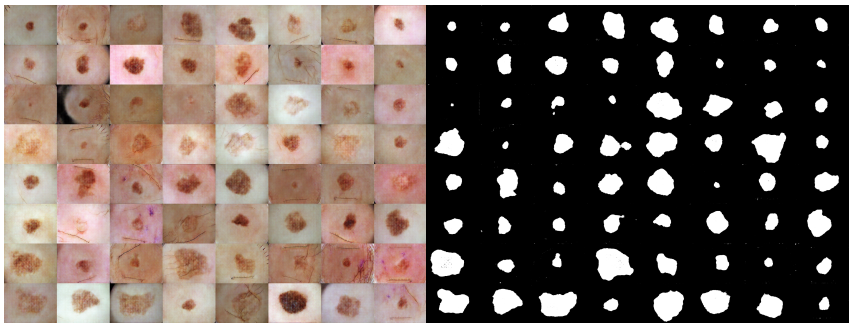
(a) (b)

Fig. 10 DCGAN-generated skin lesion samples (a) and their segmentation masks (b), generated by a GAN trained with 500 dermoscopic samples.



(a) (b)

Fig. 11 DCGAN-generated skin lesion samples (a) and their segmentation masks (b), generated by a GAN trained with 1000 dermoscopic samples.



(a) (b)

Fig. 12 DCGAN-generated skin lesion samples (a) and their segmentation masks (b), generated by a GAN trained with 1500 dermoscopic samples.

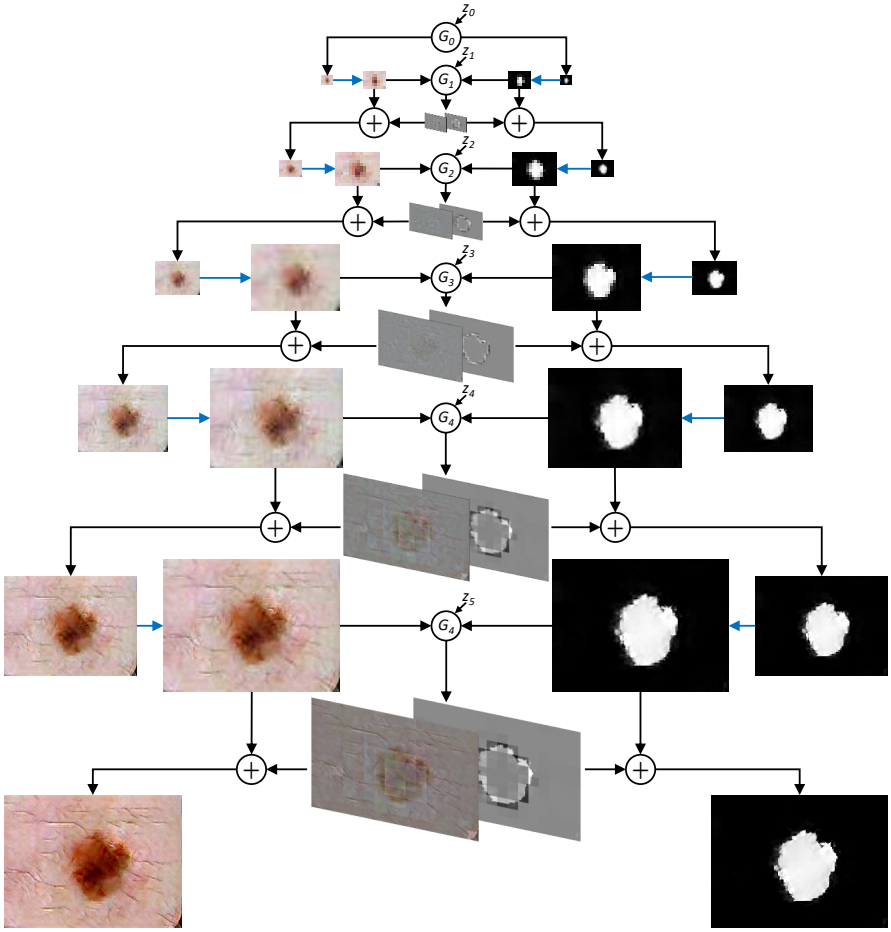


Fig. 13 A visual representation of our LAPGAN sampling process. For each level but the first one images are upsampled (blue arrows) and fed, together with a new source of noise, to a Convolutional Generative Adversarial Network, which serves to generate residual images. We add an extra step employing G_4 for a second time at the end of the process, in order to obtain 192×256 pixels samples.