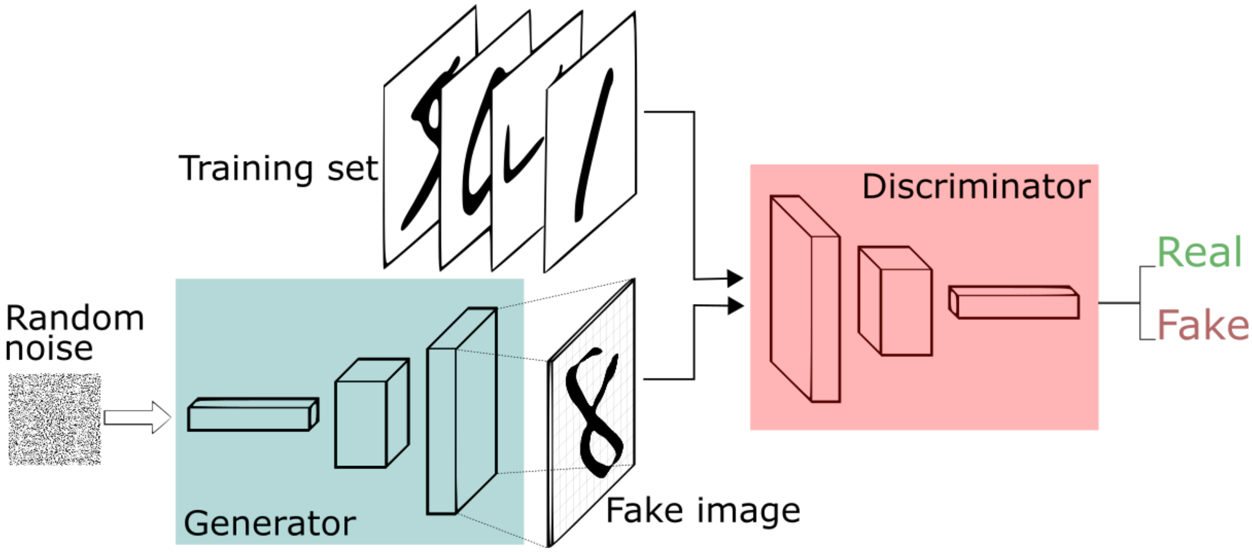


Generative Adversarial Networks for Data Augmentation

31.07.2020

Elena Kosheleva (2577871)
Hasan Md Tusfiqur Alam (2571663)
Georgii Ivannikov (2577828)

Abstract

A common refrain for computer vision researchers is that modern deep neural networks are always hungry for more labeled data — current state-of-the-art CNNs need to be trained on datasets. Data is even scarce in some specific domains especially in medical domains. Generative Adversarial Networks (GANs) can generate realistic images and those images can be augmented with real dataset as a method to improve the accuracy of different classification tasks. In our project we tried different GAN models (DCGANs, Wassertein GANs) to generate images and augmented to address the problem of class imbalance and to observe whether it can help to increase the accuracy for classification tasks.

Introduction

Generative Adversarial Networks have been improving significantly over the years and can now generate very realistic images from little training data. The idea was to use GANs to improve classification for datasets with unbalanced classes. It was decided to run our experiments on medical images, since there is always insufficient data [3] in medical imaging tasks. Our hypothesis is that by adding generated images to the training data of low-frequency classes, we can achieve accuracy improvement on those classes. The aim of the project was to prove or disprove this hypothesis. Main dataset for this work was the ISIC2019 dataset, it included 25,331 dermoscopic images among 8 different diagnostic categories, an exemplary set is given on **Figure 1**.

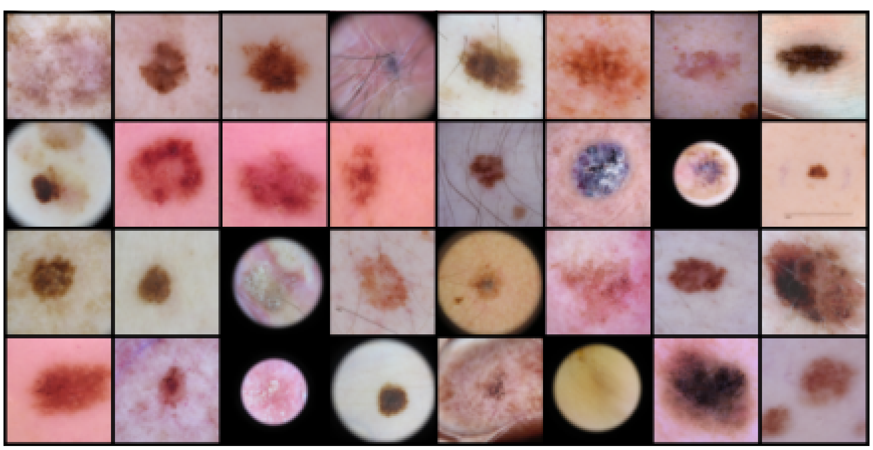
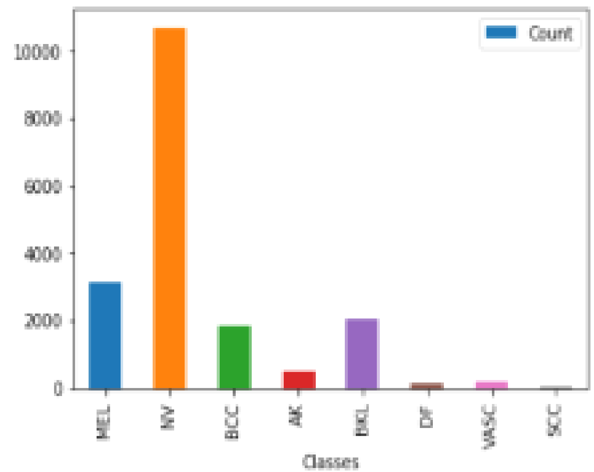


Figure 1.

Dataset was chosen because of the unequal class distribution, as on **Schema 1**.



Label	Count
MEL	4521
NEV	12875
BCC	3223
AK	867
BKL	2624
DF	239
VASC	253
SCC	179

Schema 1. Class distribution

Related work

The project was inspired by a "*Using Generative Adversarial Networks (GANs) for Data Augmentation in Colorectal Images"*[1] article from University of Seattle. Some of the work has already been done in this field. Also multiple GANs implementations are available online, e.g. Nightromes *"really-awesome-gan"* [5].

Big problem was also to find a decent dataset and an honorable mention is *"Chest-xray-pneumonia-dataset"*[6], which however was neglected later in the project and we used the ISIC skin cancer dataset.

Proposed method explained

Generative Adversarial Networks consist of two parts, namely the Discriminator (shown as D in Fig.3) and the Generator (G in Fig 3.), which try to overthrow one another. The Generator takes as input a random noise vector (z in Fig.3) and outputs an image, while the Discriminator tries to distinguish between real training data (real in Fig.3) and generated images. The Discriminator learns to full the Generator, thus generating more realistic images with each training step. A Conditional GAN additionally takes a class label into consideration, which allows it to generate images of a specific class.

On the **Figure 3** the conceptual model of a Conditional GAN[2] is shown.

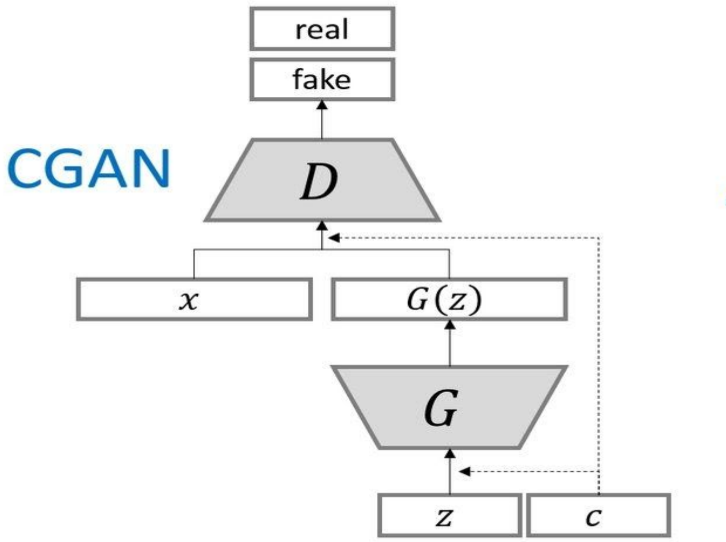


Figure 3.

As a starting point we took an implementation of DCGAN[8][9] (Fig. 4). However, a straightforward approach of turning it into a conditional model resulted in an unbalanced model: given the class label, the discriminator was overpowering the generator and the training of this model was unsuccessful. After considering existing conditional GAN architectures, a Wasserstein GAN was chosen, which was able to generate satisfactory results.

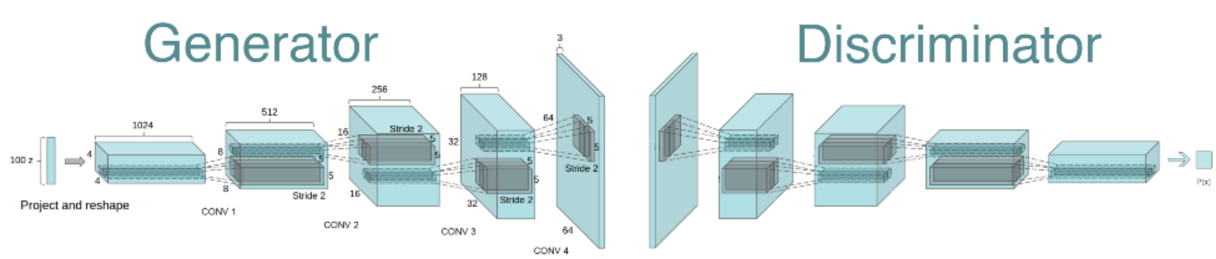
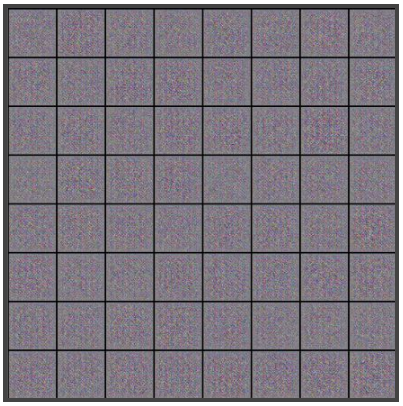


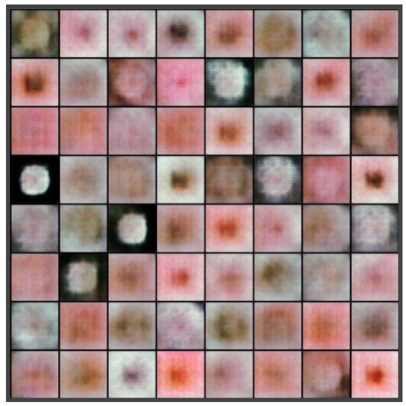
Figure 4.

Experimental results

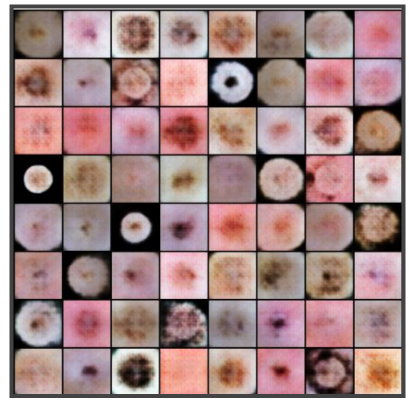
On the images (Figures 5-7) the results, generated by DCGAN. Some of the images look vague, since the class labels are not taken into account.



Weights mapping
Figure 5.



Generated results.
Figure 6.



Generated results.
Figure 7.

Best results were obtained using Conditional Wasserstein GANs as in example in Figure 8. We generated labeled images for each of the classes.

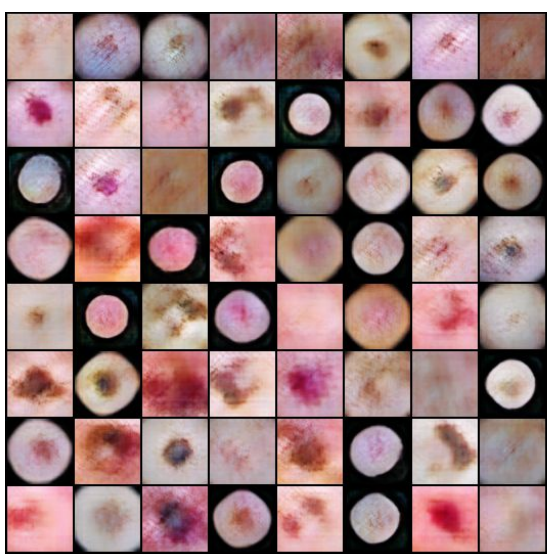


Figure 8.

The difference between the generated and real images can be seen in **Table 1**.

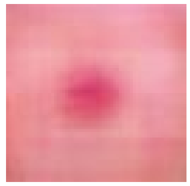
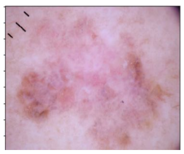
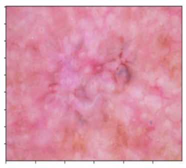
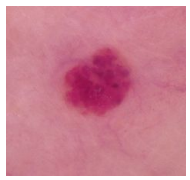
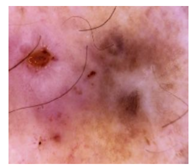
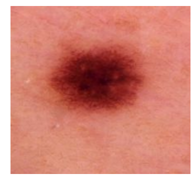
SCC	AK	VASC	DF	NEVUS	
					GAN generated
					Original images

Table 1. Comparison of generated and real images

In **Table 1,** comparison between the generated and real images for few classes are mentioned. Class SCC, AK, VASC, DF are classes with very few images. NEVEUS is the most frequent class which consists of 50% of the dataset.

Classification results

Classification model:

We tried different pretrained classification models (AlexNet, vgg16, vgg11). Due to the complexity of the classification task for this dataset, the validation accuracy for the classification task achieved is low. With more fine tuning of hyperparameters, better accuracy is achievable.

We used pretrained vgg11 with batch normalisation (vgg11_bn) as the base classification model. We trained the classifier using the imbalanced dataset without any augmentation and achieved accuracy of 33%. We have kept 30% of the data as a validation set and trained using 70% of the dataset. Stratification is used to maintain the proportionality for every class. **Table 2** consists of the classification results for each of the 8 classes. As we can see for the 4 classes (AK, DF, VASC, SCC) with very few images our classification model was't learning at all. We've also tried common augmentation methods e.g Fivecrop (cropping images into four corners and in center) but haven't achieved better accuracy. Surely, higher accuracy was achievable by fine tuning but due to time constraint we haven't properly investigated that area.

	Precision	Recall	F1-score
MEL	0.22	0.07	0.11
NV	0.34	0.77	0.47
BCC	0.33	0.08	0.11
AK	***0.00***	***0.00***	***0.00***
BKL	0.09	0.06	0.07
DF	***0.00***	***0.00***	***0.00***
VASC	***0.00***	***0.00***	***0.00***
SCC	***0.00***	***0.00***	***0.00***

Table 2: classification results with initial dataset

Classification result with Generated Data Augmentation:

We augmented GAN generated images for classes AK, DF, VASC and SCC. For each of these classes 128 images were generated and augmented. We can observe slight improvement in precision and recall for these classes. The overall accuracy increased to 39%

	Precision	Recall	F1-score
MEL	0.22	0.07	0.11
NV	0.34	0.77	0.47
BCC	0.33	0.08	0.11
AK	***0.05***	***0.01***	***0.02***
BKL	0.04	0.06	0.04
DF	***0.06***	***0.00***	***0.00***
VASC	***0.04***	***0.02***	***0.03***
SCC	***0.05***	***0.00***	***0.00***

Table 3: Classification results with GAN Augmented data

Additionally, We trained a 4-class classification model only with the generated labeled data using Conditional Wasserstein GAN. Results **Table 4** shows the moderate classification performance of the model.

	Precision	Recall	F1-score
AK	0.24	0.22	0.23
DF	0.19	0.68	0.29
VASC	0.32	0.11	0.16
SCC	0.16	0.03	0.05

Table 4: 4-class classification result for GAN Generated data

Conclusions and Future work

As a result of our project, we were able to slightly increase classification accuracy on low-frequency classes using GAN augmentation. For each of the classification models, there is room for much more improvement for the overall accuracy. But due to time constraint and the complexity of the tasks we mostly focused on generating realistic images using different variants of GANs. But we can observe there has been a sight of improvement in classification with GAN augmentation. Based on these promising results, we could try different strategies of mixing real and augmented images (i.e. in proportion 1:2 or 1:3) and run a series of experiments to see which strategy works best.

.

References

1. Using GANs for data augmentation: https://medium.com/health-data-science/using-generative-adversarial-networks-gans-for-data-augmentation-in-colorectal-images-565deda07a22
2. GANs architecture: https://github.com/jalola/improved-wgan-pytorch
3. Problem: https://challenge2019.isic-archive.com
4. Dataset: https://www.isic-archive.com/#!/topWithHeader/onlyHeaderTop/gallery
5. Really-awesome-gan: https://github.com/nightrome/really-awesome-gan
6. Second dataset: https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia
7. Discussion: https://www.cbronline.com/news/deepfakes-pornography
8. DCGAN: https://github.com/sarankrish/gans-data-augmentation
9. GAN Example: https://github.com/eriklindernoren/PyTorch-GAN

Reports to indicate assignments for each group member

Elena Kosheleva	Hasan Md Tusfiqur Alam	Georgii Ivannikov
Dataset search CGAN implementation Wasserstein GAN implementation	Dataset search Classificator implementation Augmentation and GAN implementation Validation	Dataset search Environment setting Wasserstein GAN implementation