# Synthetic Medical Image Augmentation for Classification of Pigmented Lesions

Project Report
Nicolas Essipova, Peter Lakatos, Marios Chatiras
Last edited 27th January 2020

## 1 Project Description

Generative adversarial networks, or GANs, are a class of machine learning systems that are used to generate new data given a data set, with the same statistical properties. Our work aims to display the effectiveness of synthetic data generation as a form of image augmentation technique to improve the predictive classification performance of a convolutional neural network, or CNN. Our work demonstrates the technique with and without the synthetic image augmentation, for comparative purposes.

## 2 Implementation

We approach this problem with our own variation of implementation of the technique and methodology first introduced in a paper by Frid-Adar et al. 2018 "GAN-based Synthetic medical Image Augmentation for Increased CNN Performance in Liver Lesion Classification" [1].

We begin with first describing some statistical properties of our data and then prepare our data to fit within the scope of our project. To reduce reliance on computational power and as our goal is merely to demonstrate the effectiveness of the technique; we work with significantly lower resolution and fit the data through networks of lower topological complexity, applying this approach to both the CNN and the DCGAN.

We proceed our experiment with training our convolutional neural network without the use of synthetic data – however, we did use image data generator (arguments outlined under '3. Tools') for both the training and test dataset. We trained the model until it began showing signs of overfitting on our generated test data set and utilized early stopping to acquire our best performing model. We noted down the results on validation data that was previously withheld.

We then trained our DCGAN with a selected category of data until the generator no longer improved in terms of the quality of images that it generated. Then we funnelled them into the data loader that we used to train our CNN. We retrained our CNN and noted down our results.

## 3 Tools

We utilized Keras as our computational graph library, running on TensorFlow 2.x as backend for our implementations of CNN and DCGAN. For visualization, we used PyPlot and Seaborn. For preprocessing, we used both Scikit-Learn and Keras. Our computations were carried out using Jupyter Notebook within Colab, running on the Google Cloud Platform with K80 GPU.

# 4 Results & Discussion

When we investigated the statistical properties of our data set, we first and foremost noticed that the categories ‘Melanoma’ and ‘Bening keratosis-like lesions’ were almost identical (~1100 examples), while the category ‘Basal cell carcinoma’ had almost half the volume (~500). We disregarded the category with highest volume as it was disproportionally skewed (~6700<). We limited ourselves to these three categories to keep the simplicity of our project and adding more categories would not have added any value in reproducing the research results.

Since ‘Basal cell carcinoma’ had almost half the volume of the latter two categories, we decided to use that category as our subject for image data synthesis using DCGAN to create an almost equal volume for all three categories, with which we will retrain our CNN with.

We then look at a sample of these three categories to see if we could build an intuition of how the data is structured for each category. Thereafter we investigated some statistical properties of our three categories: the RGB and grey means. What we saw is that all three categories had about the same mean red values, with melanoma having a very slight thicker right tail and lower peak. For green mean, we noticed that benign keratosis-like lesions had a noticeably lower peak than the other two categories. For blue mean, we noticed that all three categories had a nearly identical distribution, except the carcinoma which had a peculiarly small downward valley at the peak. None of these findings were particularly useful for us but does imply that there are statistical differences which the CNN may take advantage of to increase its recall for each category. Perhaps the DCGAN may take advantage of the small valley at its blue mean distribution but we did not test the significance of this hypothesis.

We then looked at the grey means for all three categories and found a very noticeable shift of the basal cell carcinoma category in relation to the other two categories. This turned out to be very convenient for us. If the DCGAN manages to capture this statistical property of the basal cell carcinoma, then the synthesized images may already prove useful in retraining the CNN even if the images are not perfectly realistic.

Our data set contained a total of 2726 images. 1113 examples belonged to category ‘Melanoma’. 1099 examples belonged to category ‘Benign keratosis-like lesions’, and 514 examples belonged to category ‘Basal cell carcinoma’. We reserved a small amount for validation set and then split our data with 70% going to training set and 30% going to test set. We created two image data generators, one for training and one for testing. For training, we used the following arguments: [featurewise_center = False, samplewise_center = False, rescale = 1./255, rotation_range = 45, width_shift_range = 0.1, height_shift_range = 0.1, shear_range = 0.01, zoom_range = [0.9, 1.25], brightness_range = [0.7, 1.3], horizontal_flip = True, vertical_flip = False, fill_mode = 'reflect', data_format = 'channels_last', validation_split = 0.25]. For our test image data generator, we used the following arguments: [rescale = 1./255, fill_mode = 'reflect', data_format = 'channels_last']. We used no image data generator for validation set. We used a mini batch size of 64 image examples that have been scaled down 28x28x3 dimensions, retaining the RGB values as we reason they are important for both CNN classification and DCGAN generation, considering the statistical means previously observed.

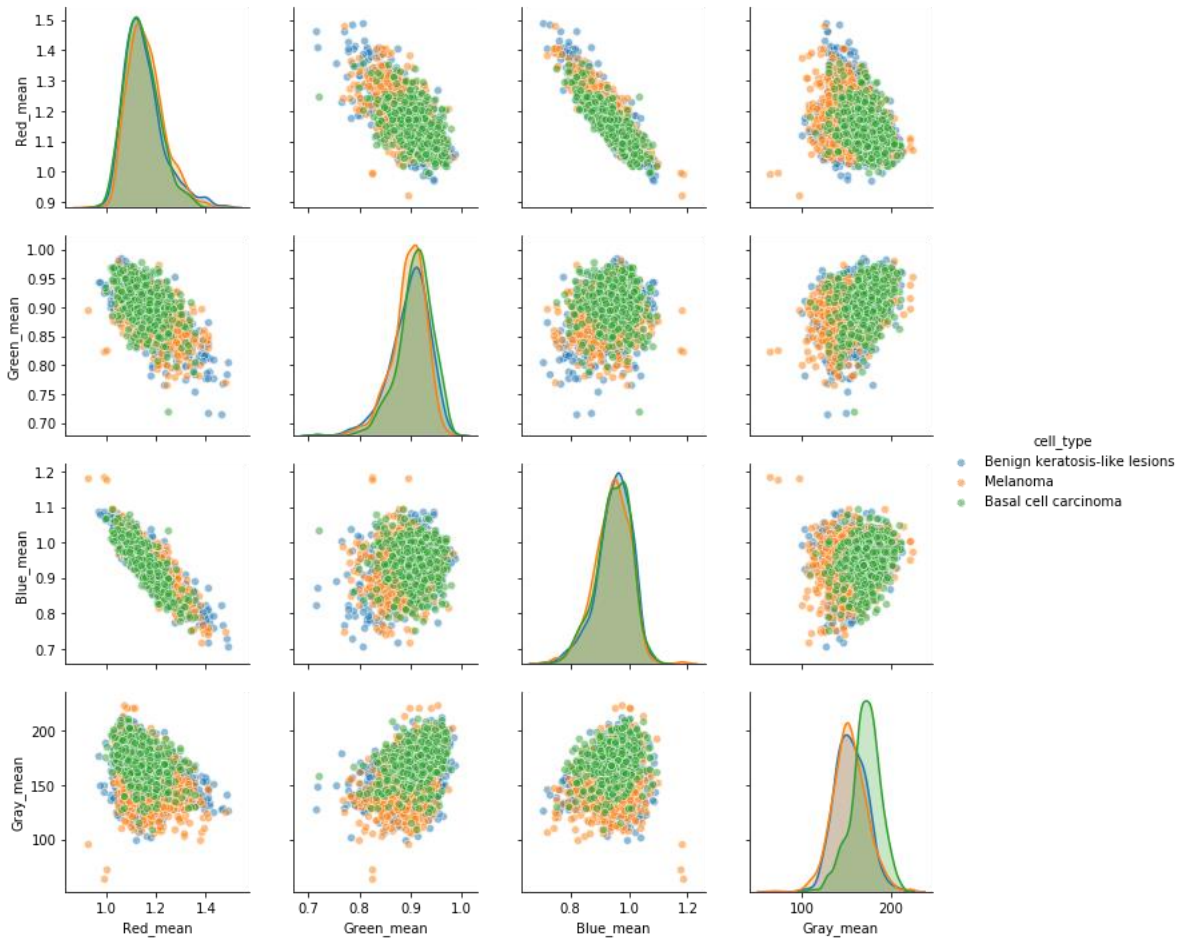
We then trained our CNN and noted down our results. We set epochs to 200 and used early stopping when we noticed that either loss or test set results started to diverge. During our first training, loss started to oscillate upwards after about 100 epochs. We compiled and trained our

CNN from scratch again and observed our best model around epoch 70. We tested it with validation set and got 59% accuracy. We reasoned that we probably could have gotten better accuracy due to the oscillation that occurred previously by reducing our learning rate but decided to proceed as we deemed that it should not make much of a difference to demonstrate our results using synthetic images. If we failed with the demonstration, then we would retrain with lower learning rate and more epochs.

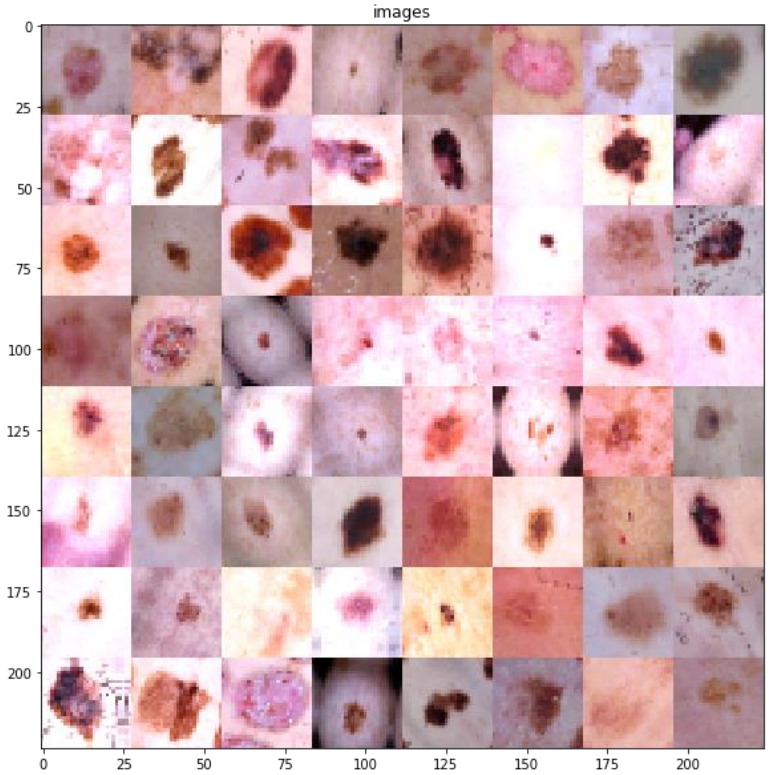
We then ran our DCGAN to exclusively generate images of the 'basal cell carcinoma' data and we ran it for about ~2000 epochs, as it ceased to improve significantly at that point. We generated new images using noise and loaded them into the training loader. We retrained our CNN and found our accuracy peaked at about 61%. Important to note that we did not test whether our results were statistically significant – as that would have required more time.

# 4 Images

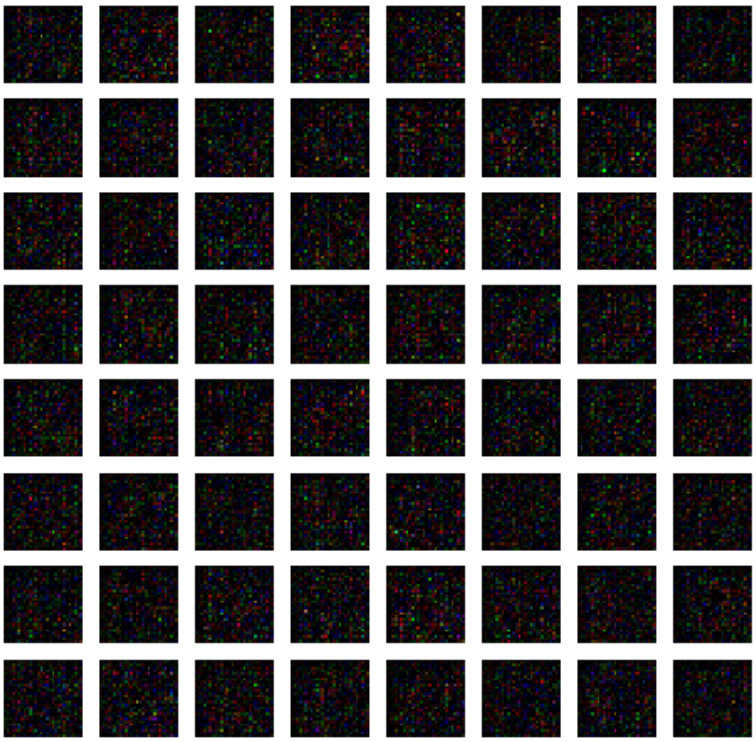
Colour distributions of our dataset (not including synthetic images):



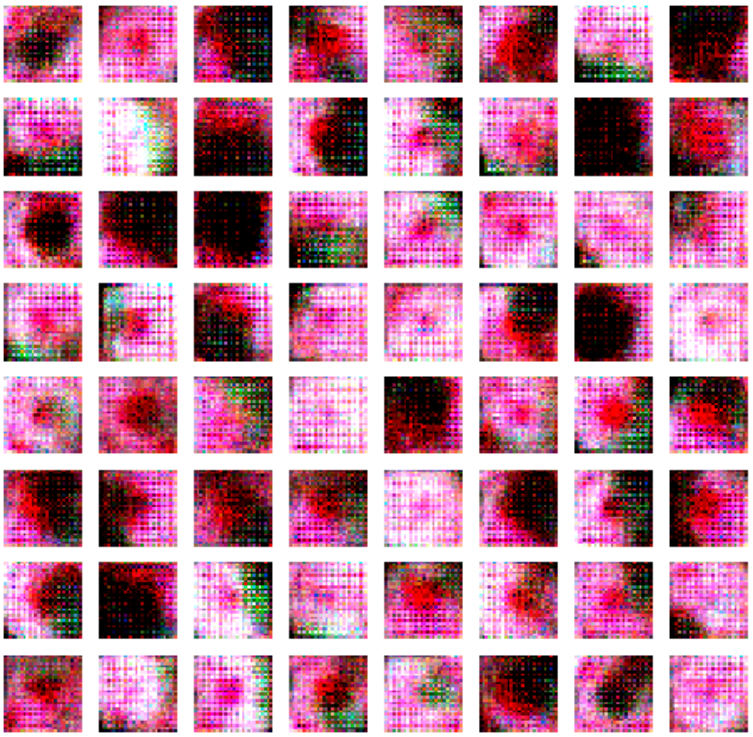
Training image data generator sample:



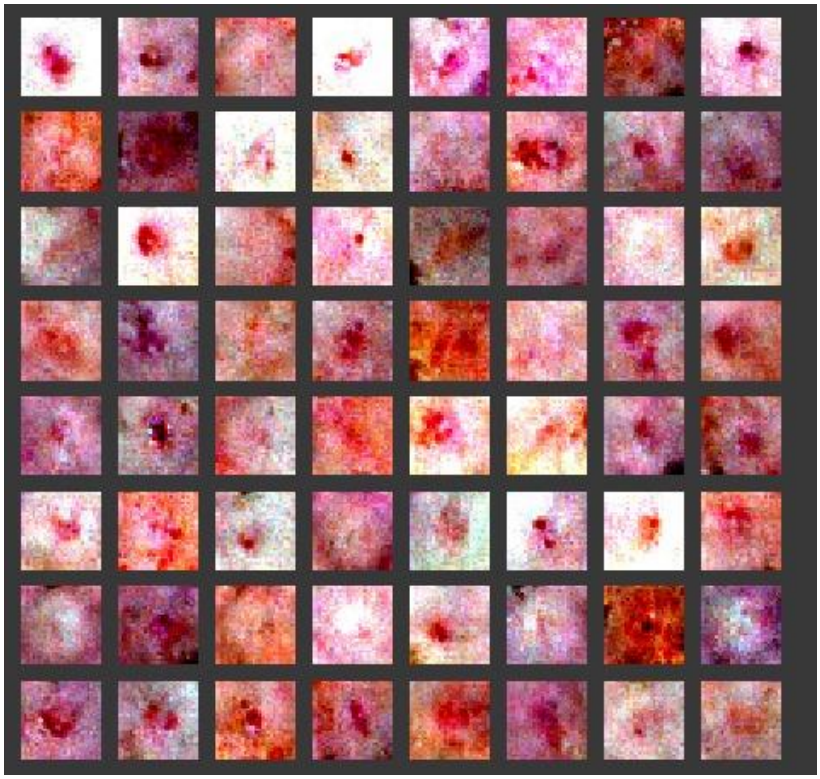

First epoch generated output from DCGAN of basal cell carcinoma:



Early epoch generated output from DCGAN (circa 100 epochs):



Later stage epoch generated output from DCGAN (circa 1000 epochs):



# 5 Data

The dataset is from the International Skin Imaging Collaboration effort and it is for the diagnosis of pigmented skin lesions to spot signs of melanoma, which is the deadliest form of skin cancer. The dataset contains 10.000 dermatoscopic images and consists of all types of pigmented lesion categories [3].

# References

[1] Frid-Adar et al. “GAN-based Synthetic Medical Image Augmentation for increased CNN Performance in Liver Lesion Classification”, 2018. Available at: https://arxiv.org/abs/1803.01229 [Accessed 061219]

[2] Alec Radford, Luke Metz, Soumith Chintala. “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, 2015. Available at: https://arxiv.org/abs/1511.06434 [Accessed 061219].

[3] Skin Cancer MNIST: HAM10000. Available at, along with detailed data source and licensing: https://www.kaggle.com/kmader/skin-cancer-mnist-ham10000 [Accessed 061219].