Towards Latent Space Exploration for Classifier Improvement

Paulo Fernandes and **João Correia** and **Penousal Machado**[1]

Abstract.

We propose a framework that combines Generative Adversarial Networks and Evolutionary computation to perform Data Augmentation on small datasets in order to improve the performance of image classifiers trained via supervised learning. In this work, we attest the viability and potential of this framework for real-world problems. The framework is composed of a generator module that uses Generative Adversarial Networks to generate samples from a dataset. It employs an Evolutionary Computation approach to evolve sets of images from the latent space. The fitness function is based on the dissimilarity of the subsets generated by the Generative Adversarial Network. A Supervisor module handles the generated samples and chooses which set should be added to the training dataset. To test the framework, we explore the Human Sperm Head Morphology dataset, a bio-medicine multi-class problem with a small number of samples that provide a challenge to the different supervised classification approaches. We deploy the framework to create an augmented dataset to train a classifier, and after the training, we compute the performance on the test set. We compare with classifiers trained using the base datasets without having the generated samples. Overall, with the preliminary tests, we can improve the performance of the classifiers by up to 4% and on average by 1%, showing the viability and potential of our approach.

1 INTRODUCTION

With the evolution of technology and computer capabilities, Machine Learning has seen significant improvements in recent years. Like so, it became much easier to build and apply larger neural networks, such as Deep Neural Networks, to solve real-world problems. It also became possible to build Deep Generative Models which produce synthetic data by learning from already existing data.

While the availability of data has also been accompanying the evolution of technology, there are still many problems that lack enough data to allow Machine Learning algorithms to be viable solutions for them. The performance of Machine Learning algorithms depends not only on the capability of the model used but also on the quality of the dataset used in the training of the model. This means that training a model with a bad dataset will, most likely, lead to poor performance results. As such, improving the quality of the datasets through Data Augmentation may be a way to improve the performance of the algorithm. With this in mind, is it possible to use generative models to enhance the quality of existing datasets, consequently improving the quality of machine learning algorithms?

One of the ways to train generative models is by using Generative Adversarial Networks. These frameworks are more often used to produce very realistic images that follow the distribution of the training dataset [5]. In general, these work by putting a generator and a discriminator against each other in a min-max game. The discriminator is trained to distinguish images from the original dataset from images created by the generator, while the generator learns from the feedback given by the discriminator on the generated data. In this work, we plan to use Generative Adversarial Networks to generate sets of synthetic images in order to understand if the addition of these instances into the training set of a classification model is able to improve its performance.

As a further matter, it is also essential to address the generation of new samples. Even though a capable model is important in the generation of better images, there is also another variable that impacts the instances generated, which is the latent space. It is unique to each generative model and hides underlying patterns in itself. Usually, to generate an image, a vector from the generative model's latent space is chosen at random for input, which means that there is no knowledge about the output. Since the images generated depend on the input given to the generator, the exploration of the latent space may reveal ways to control the output through the selection of input vectors by certain criteria. This way, we can also assure the quality of the images that will be added to the dataset and their relevance to help with the problem at hand. For instance, we might want to ensure that we are not adding redundant samples to the training set. Like so, performing random Data Augmentation should have a higher chance of undermining the performance of the algorithm, which means that ideally, we should prefer an approach of a supervised generation of instances. Bearing this in mind, we chose to perform this supervision by exploring the latent space using Evolutionary Computation. Using a Genetic Algorithm, we evolve sets of latent vectors which optimize a specific criterion such as the diversity of images in the set. This framework for latent space exploration was explored in [4]. Here we explore the usage of such a framework to generate new samples for the training dataset that may improve the performance of classifiers. As proof of concept for real-world problems, we instantiate the approach in the Human Sperm Head Morphology dataset (HuSHeM) [8], a multiclass problem categorized as small data, that provides a challenge by the lack of samples that exists for the problem.

The remaining of the paper goes as follows: in the next Section, we explain our approach to the problem and the framework used to solve it (Section 3). In Section 4, we describe the experimental setup and analyse and discuss the results. In Section 5 we draw overall conclusions.

[1] CISUC, Department of Informatics Engineering, University of Coimbra,Portugal, email: pcastillo,jncor,machado@dei.uc.pt

2 RELATED WORK

Generative Adversarial Networks are generative models that are trained through a face-off between a generator and a discriminator, mostly used to train a generator that can produce realistic images. The generator is given a noise vector to produce new images, usually, a high dimensional vector that is randomly sampled from a distribution, for example, a gaussian distribution, called the prior. The high dimensional space is called latent space. Some work has already been made to explore the latent space of generative models, and not only with Generative Adversarial Networks, which is the framework that we will use in this work. For instance, latent space exploration was performed in Kernel Principal Component Analysis [11] models, showing navigation through image features and novelty detection; but also in Variational Auto-Encoders, in, for example, mapping genes into a lower-dimensional space to uncover underlying gene expression features in cases of tumour or cancer [10].

Evolutionary Computation has also been used in some works in order to evolve images. For instance, evolving master print templates [7] that, like a master key, are able to match multiple fingerprints. In their work, Roy et al. compared four different Evolutionary Algorithms, namely Hill-Climbing, Covariance Matrix Adaptation Evolution Strategy, Differential Evolution and Particle Swarm Optimization to evolve Synthetic MasterPrints according to the metric proposed by them, the Modified Marginal Success Rate. The samples were generated from two datasets, namely Authentec AES3400 and FVC 2002 DB1-A. Beyond these, and with a two-stage workflow similar to what we implemented in this paper which includes first the unsupervised training of Generative Adversarial Networks and second the evolution of latent space, there are two works that can be mentioned. One that implements Interactive Evolutionary Computation [2] for image generation and another that uses Generative Adversarial Networks and latent space evolution to learn and improve Mario Levels [9] using Covariance Matrix Adaptation Evolution Strategy. Finally, a recent approach to generative models which was inspired by Generative Adversarial Networks, the Generative Latent Optimization [1]. This method replaces the adversarial discriminator with simple reconstruction losses where the focus is to evolve the latent space to match the one learnable noise vector to each one of the images in the training dataset.

3 FRAMEWORK

In this paper, we propose a framework that combines Generative Adversarial Networks and Evolutionary computation to perform Data Augmentation on small datasets in order to improve the performance of image classifiers trained via supervised learning.

For the framework, there are 3 fundamental pieces: (i) a classifier responsible for the classification task, discriminating images into classes; (ii) a generator, responsible for generating new images, from an array of the latent space; (iii) a supervisor, responsible for managing the generation images through the exploration of latent space (as in [4]).

3.1 Classifier

The performance of classifier will measure the performance of this framework. Therefore, we will be looking into comparing the performance of the classifier after the baseline training - using only the original dataset - against the performance of the classifier after the supervised augmented training - with selective addition of synthetic images to the training set. This way, we will be able to assess the quality of our approach and guide the progress of research.

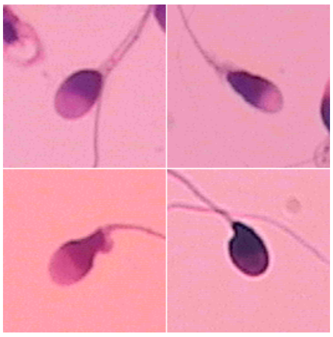

Figure 1. Class example images from original dataset. From left to right, top to bottom: Normal, Tapered, Pyriform and Amorphous.

3.2 Generator

The generator will be obtained through a Deep Convolutional Generative Adversarial Network. These make use of Deep Convolutional layers that better explore space correlation in images [6], which helps to generate better quality images. The training is unsupervised, which means that no information is given to the model to guide the generation of images, the training progresses purely through the differentiation between real(original) and fake(generated) images [5]. Therefore, a generator will be trained for each class of the problem to specifically control the generated images for each one. Each generator will learn to produce images following the distribution for a single class. During training, the generated images are created from random vectors, following a gaussian distribution.

3.3 Supervisor

Lastly, the Supervisor, which is crucial to the optimization of the training dataset. By adding random images with no criteria, there is no way to ensure if these are relevant to the solution of the problem. The introduction of flawed and redundant images might end up undermining the performance of the classification algorithm [3]. One way to control the output of the generators is by controlling its latent space. Selecting generated images through certain criteria will allow for the optimization of the classifier. More specifically, we will be looking into finding sets of images that are as diverse as possible so as to minimize redundancy. The exploration of the latent spaces will be performed through Evolutionary Computation, more specifically using a Genetic Algorithm. Each individual in the algorithm will represent a set of images where its genetic code corresponds to the latent vectors of said set of images. The initial population is created through random sampling from the same Gaussian distribution of the generator training. At each iteration, new populations are created by using Tournament Selection, Uniform Crossover and Random Reset Mutation (which also applies the previous Gaussian distribution to obtain the value of the new genes). The evaluation of individuals and fitness function correspond to the averaging of the similarities between each image in the set and the centroid image of the set that includes the images from that individual together with the images from the original dataset. The similarities between images and centroid are calculated using Normalized Cross-Correlation. Since we are searching for diverse datasets, the objective is to minimize the target function. In the end, we should find a set of images that comes closest to the intended objective and better tackles the issues at hand.

4 EXPERIMENTATION

In order to evaluate our approach, we performed several tests. The conditions on which these tests were carried out are presented in this section.

4.1 Dataset

I order to test our hypothesis we will be using the Human Sperm Head Morphology dataset (HuSHeM) [8]. In the bio-medicine context, Sperm morphology analysis is a key factor in the diagnosis process of male infertility. The dataset is divided into 4 classes of sperm heads images [Figure 1]: Normal (54 instances), Tapered (53 instances), Pyriform (57 instances) and Amorphous (52 instances) for a total of 216 images. A small dataset like this one is an opportunity to explore Data Augmentation approaches. The dataset has no sub-division, as such it was decided that we would use 40 instances of each class for training and cross-validation, leaving the remaining images for testing.

Each image has a original dimensions 131x131x3, but in the experiments we will be working with in dimensions 132x132x1.

4.2 Classifier

The classifier module allows the assertion of the experiment results. The model used was an off-the-shelf model that only required training since the optimization of the model was one of our objectives for this work. The parameters used for the training of the classifier are present in Table 1. The number of epochs chosen ensures that the classifier reaches a point of plateau for the original dataset, where there is no gain in performance. This helps to verify the quality of our solution. On another note, the training also included cross-validation for every test, which means both tests with the original dataset and augmented datasets.

Table 1. Classifier parameters

Parameter	Setting
optimizer	Adam
beta1	0.5
beta2	0.999
learn rate	0.0002
epochs	250
batch size	32
loss function	Binary Cross-Entropy
cross-validation	Stratified
folds	5

4.3 Generator

The generators allow for the creation of new samples to perform data augmentation. A generator is obtained through unsupervised training of a Deep Convolutional Generative Adversarial Neural Network. The model used for the discriminator, as the classifier, is an off-the-shelf model, to which we only added 2 extra layers of convolution. As for the discriminator, the model used was the same as the classifier. The training is performed in each individual class, which means that in this case, in particular, we are going to need 4 different generators to produce samples for each class [Figure 2]. Each generator was trained using the 40 class corresponding samples in the training set. The training parameters were set, as shown in Table 2.

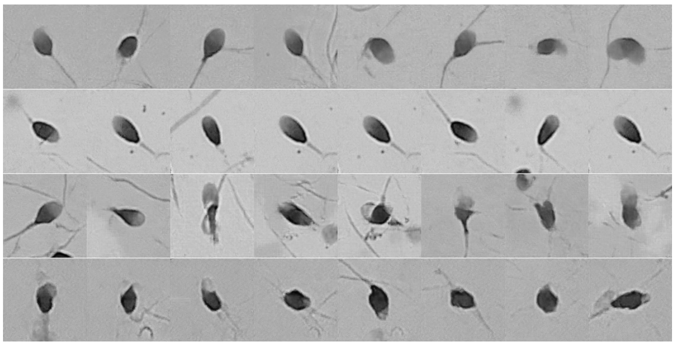

Figure 2. Class example of random images produced by the generator. On each row from, top to bottom: Normal, Tapered, Pyriform, Amorphous

Table 2. Deep Convolutional Generative Neural Network parameters

Parameter	Setting
latent dimension	100
optimizer	Adam
beta1	0.5
beta2	0.999
learn rate	0.0002
epochs	10000
batch size	32
loss function	Binary Cross-Entropy
noise distribution	N(0,1)

4.4 Supervisor

The core step of this work is the supervision of the generation of samples. This is what is going to allow the optimization of the process of performing Data Augmentation and ensure the best results possible. For this, we decided to use Evolutionary Computation, namely a Genetic Algorithm, to explore the latent space of the generators with the intent of finding sets of algorithms that optimize a certain criterion. In this case, specifically, we are looking into maximizing the diversity of the dataset. The supervision is performed for a single generator, or single class, which means that in this problem, we are going to use 4 supervisors to evolve 4 different sets of images that will be added to the original set. The parameters used in the genetic algorithm were as defined in Table 3.

Table 3. Genetic Algorithm Parameters

Parameter	Setting
Population size	20
Number of generations	500
Genotype length	number of images × latent dimension
Elite size	1
Selection method	tournament
Tournament size	3
Crossover operator	uniform crossover
Crossover rate	0.7
Mutation operator	random reset mutation
Mutation distribution	N(0,1)
Mutation rate per gene	0.02

Each individual represents a set of images that are coded into its genetic code in the form of latent vectors The fitness of each individual is calculated by averaging the similarities between each image in the set and the centroid image of the set that includes the images from that individual together with the images from the original dataset. The similarity is calculated using the Normalized Cross-Correlation metric. The calculations are as follows.

$$T = I^\frown O \tag{1}$$

$$C = \frac{\sum_{t \in T} t}{length(T)} \tag{2}$$

$$F = \frac{\sum_{i \in I} NCC(i, C)}{length(I)} \tag{3}$$

T is the set resulting of concatenation of the images from the individual (I) with the images from the original dataset(O). C is the centroid of the set T. F is the fitness of the individual calculated through averaging the similarities measured using the Normalized Cross-Correlation(NCC). The Calculation of the similarity metric is as follows:

$$NCC(A, B) = \frac{\sum((A - B) \odot (A - B))}{\sqrt{\sum(A \odot A) \times \sum(B \odot B)}} \tag{4}$$

The $\odot$ corresponds to the Hadamard product between two images.

On a last note, since the fitness function measures similarity instead of diversity, the objective of the algorithm is set to minimization. In the end, we should end up with a set of images that are more diverse, than if we just picked a randomly.

4.5 Experimental Results

In order to test our framework, we performed a comparison between the performances of the classifiers before and after performing Data Augmentation. The evaluation of each classifier in the cross-validation was performed in the test dataset, where several metrics were measured, namely Accuracy, Precision, Recall, F1 score, Area Under Receiving Operator Characteristic Curve (AUROC) and Average Precision. Each test was performed 5 times with different seeds. In the following results we will be presenting the mean across these 5 repetitions. Note that the initialization of the weights is the same between Model-X and Seed-X (e.g. Model-0 and Seed-0), Seed-0 differs on the seed used to generate the augmented dataset and, of course, the existance of augmented instances.

The first tests were performed with the original dataset. By looking at Figure 3 we can observe that at 250 epoch the training has already reached a plateau in terms of accuracy, which means that it will most probably not get any benefits from further training since it will tend to overfit. We were also able to find the performance that our solution should be able to overcome [Table 4.5].

The next step was building the sets of images of images to be added to the original dataset and train with the augmented dataset. For this experiment it was decided to test a dataset composed of 50% original images and 50% synthetic images. As such, for each class we generated and evolved sets of 40 images. The selection of these sets was repeated for every repetition of the test with the classifier which was performed 5 times with different seed, similarly to what was done with the baseline test.

First, by analysing the line of evolution in Figure 4 we can see that the values of fitness of the best individuals do not have a great variation between the first and the last generations. All values, for every class, sit on an interval of 0.01 between, 0.99 and 1. This means that the similarity between the images in this dataset, for this metric, is really high and that it promote a good evolution. However, if we take look at Figure 5 which puts side-by-side the best individual of the first generation (top) against the best individual of the last generation (bottom) in the evolution of a set of the class "Amorphous", we can

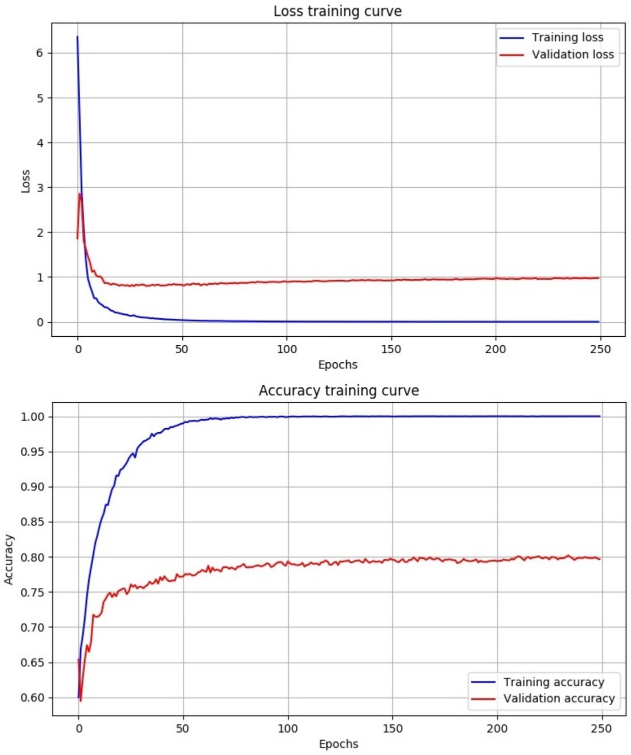

Figure 3. Average of the learning curve of the trainings with the original dataset across the all repetitions

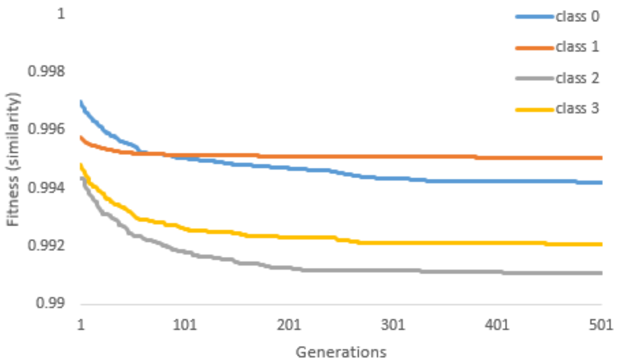

Figure 4. Average of the fitness of the best individual during the evolution process across all repetitions

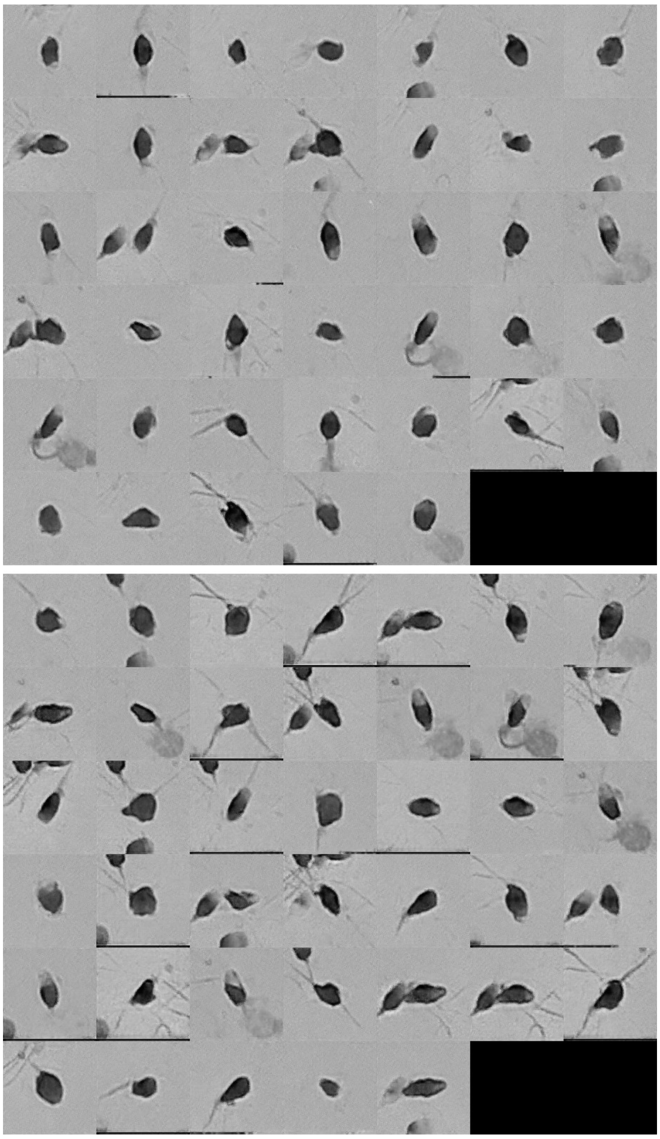

Figure 5. Best individuals at the end of generations 0 and 500, top and bottom respectively, from the process of evolution of individuals from the class ”Amorphous”

argue that the latter has in fact, from a subjective perspective, more visual diversity than the first.

The last step is the training of the classifiers with the augmented datasets. By analysing the training curve in Figure 6 we can see that at epoch 250 the training also reaches a plateau accuracy-wise, meaning that further training would not help better the performance.

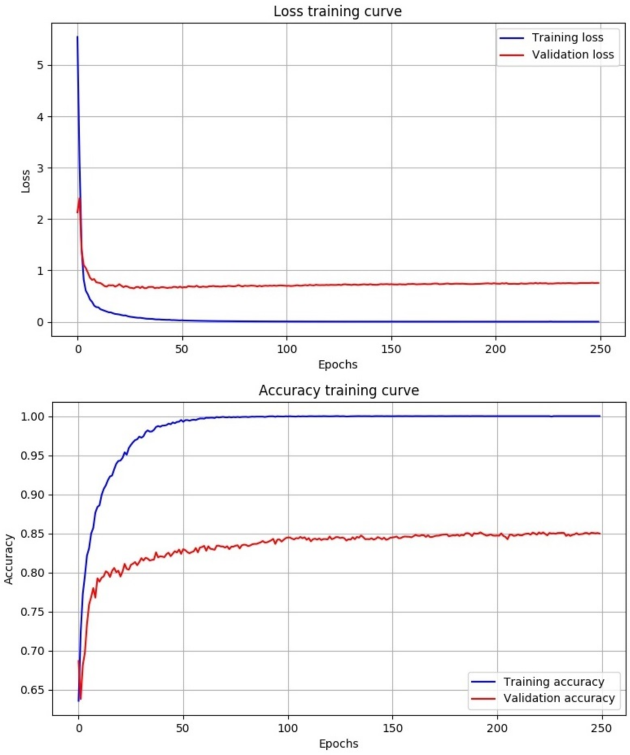

Figure 6. Average of the learning curve of the training with the augmented dataset across the all repetitions

Looking at the test results [Table 4.5], we can see that the performance of the classifiers trained with the augmented dataset was, on average, better for every metric. Although it would be necessary to perform more tests to verify the benefit of our solution, this shows that our approach might indeed be a way to improve datasets and consequently the performance of Classifiers. Each *Seed-X* represents a classifier that was trained by a subset from the Evolutionary Computation process using a different random generator seed. We can observe that in 4/5 seeds we are able to improve beyond the average performance of the original baseline classifier. We have one Seed that improves up to 4% over the baseline average. One of the seeds hindered the performance of the classifier but on average we have improvements over all the metrics when compared with the trained models with different initialization weights.

5 CONCLUSION

We explored an approach that uses Generative Adversarial Networks for Data Augmentation to improve the performance of a supervised classifier applied in a real-world problem. The underlying idea is to explore the latent space of the generative model using Evolutionary Computation to generate sets of instances to be used in the training dataset of a supervised classifier. Since arbitrarily adding instances to the dataset could hinder the performance of the classifier that is

Table 4. The classifier test results for each metric. Each model trained is denoted as *Model-X* and each model trained with the augmented dataset from the Evolutionary process is denoted as *Seed-X*. Augmented is the average of the 5 seeds and Original is the average of 5 models trained.

	Metrics					
Data	Acc	Prec	Rec	F1	AUROC	Avg-Prec
Original	0.578	0.623	0.580	0.582	0.719	0.487
Model-0	0.593	0.610	0.598	0.592	0.731	0.508
Model-1	0.575	0.648	0.580	0.588	0.719	0.497
Model-2	0.571	0.599	0.569	0.563	0.712	0.469
Model-3	0.571	0.621	0.575	0.581	0.715	0.472
Model-4	0.579	0.641	0.580	0.587	0.718	0.486
Augmented	**0.586**	**0.633**	**0.590**	**0.594**	**0.725**	**0.496**
Seed-0	0.589	0.645	0.587	0.594	0.724	0.498
Seed-1	0.589	0.632	0.596	0.596	0.729	0.495
Seed-2	0.539	0.586	0.534	0.537	0.689	0.446
Seed-3	0.596	0.642	0.604	0.609	0.734	0.508
Seed-4	0.618	0.663	0.629	0.633	0.750	0.536

being trained, we rely on a Supervisor module that selects the best set based on different criteria.

We instantiate this framework in a real-world application problem the Human Sperm Head Morphology dataset has a proof of concept. Due to the small number of instances we can categorize it as small data dataset, which presents an opportunity to explore Data Augmentation approaches. We created a baseline classifier with the provided data for comparison with the classifiers created by our framework. We used a Genetic Algorithm to evolve sets of latent space vectors that generated sets of images. We used the normalized cross-correlation similarity metric to calculate the dissimilarity among the sets and used the average value to assign fitness to each one. Overall we were able to guide evolution and generated dissimilarity subsets. The best subset from the last population of the evolutionary algorithm was used to augment the training dataset of the classifier. The classifier was then trained with the synthetic and with a base subset of instances. We used cross-validation to compute performance metrics. Overall the results show that we can increase the performance of the classifier. For example, we were able to raise accuracy by 0.8% and the f1-score by 1.2%. Although more tests are needed to verify this conclusion and even to improve the quality of the solution, it is a first step and a proof of concept of the potential of such an approach.

Future work may include testing with different proportions between original images and generated images in augmented datasets, and even testing on training sets composed of generated images only. Also, we may even test different datasets, use different similarity metrics or even improve the supervision algorithm with the inclusion of other techniques. Finally, we may also look into comparing this approach to other data augmentation approaches.

REFERENCES

[1] Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. Optimizing the latent space of generative networks, 2017.

[2] Philip Bontrager, Wending Lin, Julian Togelius, and Sebastian Risi, 'Deep interactive evolution', in *Computational Intelligence in Music, Sound, Art and Design*, eds., Antonios Liapis, Juan Jesús Romero Cardalda, and Anikó Ekárt, pp. 267–282, Cham, (2018). Springer International Publishing.

[3] João Correia, *Evolutionary Computation for Classifier Assessment and Improvement*, Ph.D. dissertation, University of Coimbra, 2018.

[4] Paulo Fernades, João Correia, and Penousal Machado, 'Evolutionary latent space exploration of generative adversarial networks', in *EvoApps 20' Proceedings of the 23rd European Conference on the Applications of Evolutionary and bio-inspired Computation - Evolutionary Machine Learning*, p. to appear, (2020).

[5] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio, 'Generative adversarial nets', in *NIPS*, pp. 2672–2680, (2014).

[6] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2015.

[7] A. Roy, N. Memon, J. Togelius, and A. Ross, 'Evolutionary methods for generating synthetic masterprint templates: Dictionary attack in fingerprint recognition', in *2018 International Conference on Biometrics (ICB)*, pp. 39–46, (Feb 2018).

[8] Fariba Shaker. Human sperm head morphology dataset (hushem), 2018.

[9] Vanessa Volz, Jacob Schrum, Jialin Liu, Simon M. Lucas, Adam Smith, and Sebastian Risi. Evolving mario levels in the latent space of a deep convolutional generative adversarial network, 2018.

[10] Gregory P. Way and Casey S. Greene, *Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders*, 80–91, 2018.

[11] D Winant, Joachim Schreurs, and J Suykens, 'Latent space exploration using generative kernel pca', in *Proc. of the 28th Belgian Dutch Conference on Machine Learning (Benelearn2019)*. BNAIC/Benelearn, (2019).