

Interactive Neuroevolution: Human Driven GANs-based Data Augmentation using Neuroevolution

Usman Kaisser Moghal

19i-1245

Master of Science (Computer Science)



Department of Computer Science

National University of Computer & Emerging Sciences

Islamabad, Pakistan.

2021

Interactive Neuroevolution: Human Driven GANs-based Data Augmentation using Neuroevolution

Submitted by

Usman Kaisser Moghal

19i-1245

Supervised by

Dr. Omer Ishaq

Master of Science (Computer Science)

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science (Computer Science) at National University of Computer & Emerging Sciences



Department of Computer Science

National University of Computer & Emerging Sciences

Islamabad, Pakistan.

15-12-2021

Plagiarism Undertaking

I take full responsibility of the research work conducted during the MS Thesis titled "Interactive Neuroevolution: Human Driven GANs-based Data Augmentation using Neuroevolution". I solemnly declare that the research work presented in the thesis is done solely by me with no significant help from any other person; however, small help wherever taken is duly acknowledged. I have also written the complete thesis by myself. Moreover, I have not presented this thesis (or substantially similar research work) or any part of the thesis previously to any other degree awarding institution within Pakistan or abroad.

I understand that the management of Department of Computer Science of National University of Computer and Emerging Sciences has a zero tolerance policy towards plagiarism. Therefore, I as an author of the above-mentioned thesis, solemnly declare that no portion of my thesis has been plagiarized and any material used in the Thesis from other sources is properly referenced. Moreover, the thesis does not contain any literal citing of more than 70 words (total) even by giving a reference unless I have the written permission of the publisher to do so. Furthermore, the work presented in the thesis is my own original work and I have positively cited the related work of the other researchers by clearly differentiating my work from the irrelevant work.

I further understand that if I am found guilty of any form of plagiarism in my thesis work even after my graduation, the University reserves the right to revoke my MS degree. Moreover, the University will also have the right to publish my name on its website that keeps a record of the students who plagiarized in their thesis work.

Student's name

Dated:

Author's Declaration

I, Usman Kaiser Moghal, hereby state that my MS thesis titled "Interactive Neuroevolution: Human Driven GANs-based Data Augmentation using Neuroevolution" is my own work and it has not been previously submitted by me for taking partial or full credit for the award of any degree at this University or anywhere else in the world. If my statement is found to be incorrect, at any time even after my graduation, the University has the right to revoke my MS degree.

Student's name

Date:

Abstract

Generative Adversarial Networks (GAN) have become very popular as of late in various fields, specifically computer vision. Several new methods have been discussed and implemented. However, despite this rapid development, GANs still have some issues. Specifically, with regards to training the generator. In this thesis, a novel approach into training GANs is discussed utilizing Neuroevolution coupled with human feedback. Neuroevolution is the involvement of a Genetic Algorithm (GA) to improve the training of a GAN. Here, the GA is specifically used for improving the generators training process.

Based off of recent studies, introducing a human observer to interact with and accommodate the training process of neural networks, has shown to produce positive results. In this thesis, we utilize Two-Alternative Forced Choice (2AFC) for improving the training process of the generator. The idea is essentially to inject human interaction into the training process of the generator. In certain domains, a human's ability to discern is greater than typical AI, and adding this to a GANs training process can greatly skew the results towards better results.

Within a network, certain weights influence the results more than others. In this thesis, these weights ("Golden Indices") are calculated and exploited. The resultant framework of combining neuroevolution and 2AFC with a GAN while utilizing "Golden Indices", produces results qualitatively similar to DCGAN, but with greater efficiency. The generator gets trained at a fraction of the time as DCGAN, and with lower computational requirements as well.

Acknowledgements

This thesis could not have been possible without the will and guidance of Allah. At several moments where I felt stuck, He showed me a way, and made things easy for me. Allah be praised.

Amongst us mortals, I would first like to direct attention towards my wife, Huda Inam, who sacrificed much of herself to allow me the time I needed to complete this thesis. She was always there for me, providing me with coffee and food when I needed it, making sure that our daughter was taken care of and not disturbing me while I worked, and most importantly, sacrificing her own need and wants so I could have the space I needed to function. May Allah reward you in this life and the next.

It goes without saying that my parents were a pillar of support during this endeavor. Without their support, I would not have been even capable of attempting a degree in higher education, much less be able to complete it as well as I have.

Next would be my supervisor, colleague, and friend, Omer Ishaq. He took me under him even though he already had several students waiting on him, taught me, and guided me. I would drop in unannounced in his office multiple times a week, and he was always welcoming. And whenever I messed up or slacked off, he would correct me and push me to work harder. Without his encouragement and guidance, this thesis would have ended before it even began.

Finally, I would like to thank my examiners, Bilal Khalid, Zainab Abaid, and Fahad Tahir. You all took the time to go through my thesis, and critiqued me for my own improvement. You were neither harsh nor unfair. I learned a lot from our interactions, and my thesis has improved as a result.

Table of Contents

Table of Contents

1.	Introduction	10
1.1	Introduction and Background.	10
1.2	Motivation and Research Problem	12
1.3	Overview of the Proposed Solution	13
1.4	Major Contributions.....	13
2.	Literature Review	14
2.1	Neural Networks	14
2.2	Datasets.....	14
2.3	Generative Adversarial Networks (GANs).....	15
2.4	Difficulty with training GANs.....	16
2.5	Human Interaction with Networks.....	16
2.6	Genetic Algorithms (GAs)	19
2.7	Evaluation Metrics	20
2.8	Research Gap	20
2.9	Problem Statement	20
2.10	Research Objectives	22
3.	Methodology.....	23
3.1	Model of the Proposed Solution	23
3.2	Dataset and Implementation Details	24
3.3	Evaluation Metrics	24
3.3.1	MS-SSIM	25
3.3.2	Fréchet Inception Distance (FID).....	25
3.4	Generative Adversarial Networks (GANs).....	26
3.4.1	Discriminator.....	27
3.4.2	Generator	28
3.4.3	Fitness	29
3.5	Genetic Algorithms (GAs)	29
3.5.1	Initial Population	31
3.5.2	Fitness Function	31
3.5.3	Selection.....	31
3.5.4	Crossover.....	32

3.5.5	Mutation.....	33
4.	Implementation.....	34
4.1	Population Initialization	35
4.2	Golden Indices Calculation	36
4.3	Population Sorting.....	37
4.4	Two-Alternative Forced-Choice (2AFC) process.....	38
4.5	Crossover	39
4.6	Mutation.....	39
4.7	Selection	39
4.8	DCGAN training the Golden Network.....	40
4.9	Testing Results.....	40
5.	Results and Discussion	41
5.1	Observations during testing	41
5.2	Qualitative Results.....	44
5.3	Quantitative Results	45
5.3.1	Efficiency	45
5.3.2	Attempting Quality Comparison.....	46
5.4	Key findings	47
5.4.1	Discriminator difficulty	47
5.4.2	Limited Trainings	48
6.	Conclusion	49
6.1	Summary	49
6.2	Future work	49
7.	References.....	50

List of Figures

Figure 1: An example of mode collapse for MNIST dataset. (a) Generated data (b) Landscape around datapoints	11
Figure 2: Comparisons of several "state of the art" GANs using input sketches	12
Figure 3: Framework of the proposed solution	23
Figure 4: Architecture of a Generative Adversarial Network	26
Figure 5: Training the discriminator in a Generative Adversarial Network	27
Figure 6: Training the generator in a Generative Adversarial Network.....	28
Figure 7: Failure of Gradient Based Methods.....	30
Figure 8: Population, Chromosomes and Genes	31
Figure 9: Exchanging genes among parents on crossover point	32
Figure 10: Mutation: Before and After	33
Figure 11: Proposed Framework Pipeline	34
Figure 12: Two-Alternative Forced-Choice (2AFC) Process.....	35
Figure 13: Common Indices of extreme weights	41
Figure 14: Greatest changes in weights across trainings.....	42
Figure 15: Greatest changes in common Indices of extreme weights.....	42
Figure 16: Influencing weights of specialized indices	43
Figure 17: Best results comparison.....	44
Figure 18: Results comparison, small improvement.....	44
Figure 19: Results comparison, large improvement.....	45

List of Tables

Table 1: Comparison of different GANs structures	21
Table 2: Comparison of different GANs Implementations	22
Table 3: System Specifications	45
Table 4: Framework's Operating Time per Iteration	46
Table 5: Quantitative Comparison with Different 'Original' Sources	47

List of Equations

Equation 1: MS-SSIM index calculation	25
Equation 2: FID score calculation.....	25
Equation 3: Loss function in GANs	29

1. Introduction

1.1 Introduction and Background.

Deep neural networks have proved to be state of the art algorithms for a number of different application domains in computer vision and image analysis [1] [2] [3]. Typically, deep neural networks require a large amount of annotated data for effective training. In the past, researchers have struggled with a 42GB dataset [4], and even larger datasets exist [5] where researchers will have to deal with a 52GB dataset. There even exists a 150GB dataset which is frequently used by several researchers [6]. Researchers have attempted to work with such large datasets [7], but have had their work restrained due to the sheer size of these datasets.

Researchers are impeded by lack of appropriate datasets (with problems such as insufficient data, unlabeled data, imbalanced data, etc.). For instance, researchers struggle with medical tasks, as there is almost always a shortage of sufficient data [8]. To mitigate this, various techniques have been developed focusing on data augmentation.

Generative Adversarial Networks (GANs) are extensively being used nowadays for data augmentation. GANs can be used in various domains with regards to dealing with limited data [9].

Researchers are heavily dependent on GANs in almost all types of tasks, such as synthesizing medical images [10], emotion classification [11], machine fault diagnosis [12], and speech recognition [13]. As such the list goes on.

In all cases, despite the obvious benefits of GANs, there is an issue that researchers often sweep under the rug. That issue lies in the difficulty in training the GAN, particularly the generator.

A general understanding regarding the difficulty in training GANs has been noted [14] by several researchers, and can be found in multiple publications in great detail. A common issue many researchers face is mode collapse, where the network can get stuck in a local minima, and continue to produce repetitive inaccurate results, such as detailed in Figure 1 [15]. This issue in particular, has been researched in great detail [16].

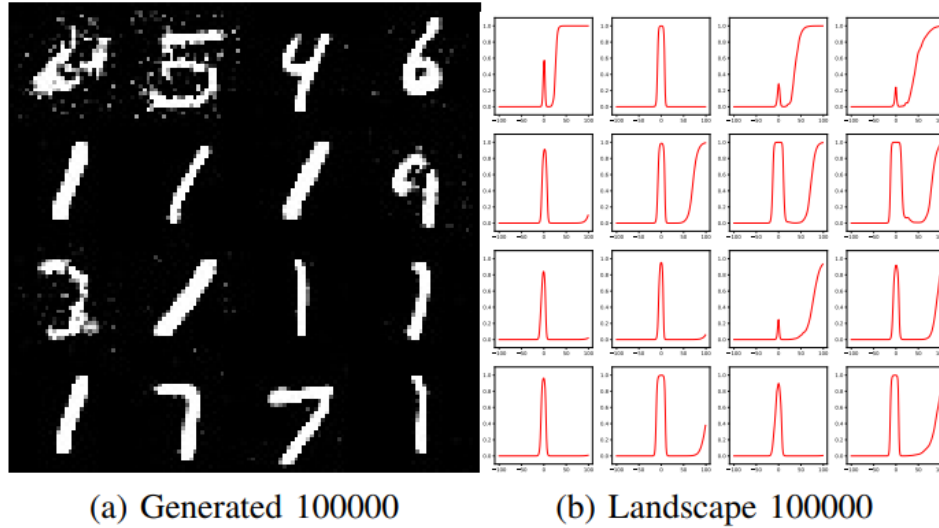


Figure 1: An example of mode collapse for MNIST dataset. (a) Generated data (b) Landscape around datapoints

In comparison to a poorly trained GAN, a human is far more accurate in discriminating between real and fake data. This concept of including a human for assisting a network's training is not alien, and has been implemented to a degree by some researchers [7] [17] [18].

In an attempt to improve the networks output, researchers [7] have included humans for the networks training process by choosing pixels from the image and assigning colors to them, rather than taking the colors from the ground truth. Some researchers have decided to take this human input as soft constraints [17] that the network will consider when generating images, while others [18] have taken the human input as a sample theme image which the network will take as a template drawing to generate an image.

These papers showcase the benefits of including a human element, which in doing so, enables the researchers to create a framework that provides superior results to existing standards. Unfortunately, despite the great success these implementations have achieved, they are limited in their scope and domain, and require potentially extensive customization to use in a different task. There is a need for a generic solution, and this thesis aims to achieve just that.

One of the methods to incorporate human feedback into a networks training process is to use 2AFC, a technique that measures the sensitivity of a rater to external stimulus and improves decision making based on behavioral patterns in input data [19] [20]. Using a method like 2AFC, GANs can be further optimized for data augmentation tasks. We are also exploring other avenues for incorporation of human feedback in the GAN training process.

1.2 Motivation and Research Problem

Very little work has been done with techniques involving human interaction in the training loop of deep neural networks, specifically GANs. Introducing human input with GANs may prove to be an effective method to train a learning model.

Despite various attempts, the outputs of GANs can still be bad [17]. Figure 2 [17] shows a comparison of several “state-of-the-art” GANs against DeepFaceDrawing, which uses human input to improve the results. It is clear that this input makes a positive difference.

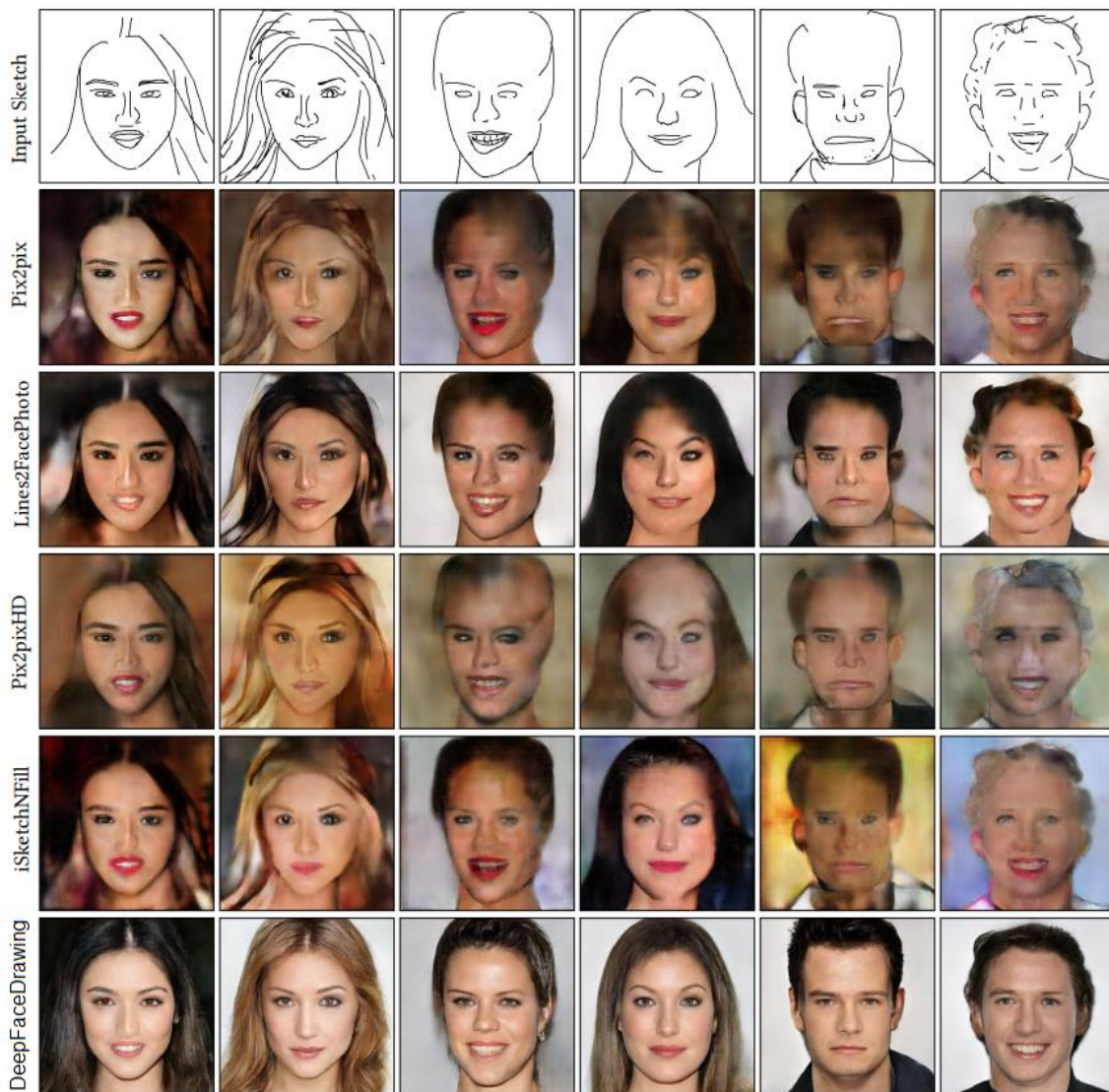


Figure 2: Comparisons of several "state of the art" GANs using input sketches

Similar to the techniques seen in DeepFaceDrawing [17], human interaction has been introduced for improved training [7] [18], however such techniques are restricted to their domains and only show results under constrained specifications of a specific application. As an example, the iDeepColorization [7] will only work for colorizing images, and cannot function outside of the given application setup; the DeepFaceDrawing framework

[17] requires a specified input and can only be used for generating faces using face sketches; and while the framework for SPADE [18] is somewhat generic, the results it produces are more tailored for a specific type of output. Modifying SPADE [18] for a completely generic use case, or even for a specific one, is not a simple task.

The goal behind this thesis is to introduce a generic framework, that can be attached onto potentially any problem, and allow human feedback for the training process. This would lead to improved training, which will allow better synthetic images for potentially any application. One possible method for human interaction is 2AFC.

1.3 Overview of the Proposed Solution

As seen in Figure 3, the proposed solution is similar to a standard GAN, with the inclusion of human interaction and a genetic algorithm.

The architecture utilizes a hybrid approach, where for parts of the dataset, the discriminator will be swapped out with a human operator during the training process. During this time, the human will discern between the input data and generated data, using techniques such as 2AFC.

The genetic algorithm is used to select optimal weights from the network and perform mutations to get potentially better weights, which will then be swapped with the network's weights. The use of a GA also eliminates the issue of mode collapse.

Essentially, the architecture boils down to using a human operator for generating a specialized loss, and improving the network through passing said loss to the genetic algorithm which will optimize the network.

1.4 Major Contributions

- Novel approach into training GANs
- Proving the usefulness of human interaction for network training
- Generic solution for most application areas
- Create a solution free from some typical GAN issues like mode collapse

2. Literature Review

2.1 Neural Networks

Due to the inherent ability of neural networks to overcome drawbacks of traditional algorithms by learning approximations of functions, they have gained an ever-increasing interest in various fields. In a survey conducted in 2017 [1], researchers have detailed some of the most popular deep learning architectures along with their applications. They have also provided a list of active areas of research, as well as future research topics, all with justifications.

A publication from 2015 [2] provides an overview of various deep learning techniques, including supervised learning, unsupervised learning, reinforcement learning and evolutionary computation. It also details approaches into indirect search for short programs encoding deep and large networks.

The better a network's architecture is for a task, the better the network performs. Designing a custom framework for every problem is a painstaking task. Some researchers have discussed in detail [3], an automated method to evolve a network for more optimized learning architectures, while supporting the ability to build real-world applications.

2.2 Datasets

GANs are commonly used for medical datasets. Some researchers [4] have proposed a novel approach for detection of abnormalities in chest x-rays. It uses the inherent properties of GANs to distinguish between two variants, that is normal and abnormal x-rays.

Among medical images, similar to x-rays, CT scans are very popular [5], specifically for the detection pulmonary embolisms.

The YFCC100M dataset [6] is one of the largest image-based datasets publicly available at this time. As networks became more complex and required larger datasets, this dataset was collected in an effort to provide a consistent and concise collection of images to scale up networks and improve learning of concepts at higher layers.

There are instances in certain domains, where datasets are limited. This prevents researchers from actively finding a solution using neural networks, as lack of data leads to overfitting. Researchers have shown how GANs can be used in such scenarios [9] to overcome this limitation.

2.3 Generative Adversarial Networks (GANs)

Medical tasks are often plagued with lack of data. SpeckleGAN [8] is framework showcasing how GANs are used to develop synthetic medical images, with emphasis given to deal with speckle noise in the images. The authors provide a novel framework that outperforms traditional GANs, and creates a crisp image void of typical noise, despite limited dataset availability.

Similar to the dataset SpeckleGAN was originally trained on [8], there is a lack of appropriate data in other areas [10], specifically a case of an imbalanced dataset. Medical datasets often have these issues. Some researchers have proposed a framework [10] to generate synthetic abnormal brain MRIs, which will eventually balance the dataset. The results indicate improved segmentation of tumors.

Another domain where availability of appropriate datasets is scarce, is emotion classification [11]. This time the problem is with unlabeled data, along with a degree of imbalanced data. Emotion classification is a task where such issues with data are inherent. The paper discusses these issues, and proposes a GAN that can deal with missing labels and imbalanced datasets. Results show that the GAN increased the network's prediction by a significant margin.

Given the success GANs have at generating artificial data that is alike real data, certain researchers have proposed a framework [12] to use GANs to learn from mechanical sensors and generate realistic one-dimensional raw data. The method is designed to produce synthetic signals with labels, allowing the generated data to be used as augmented data for future applications to detect machine faults.

GANs are even used to produce a more accurate method for speech recognition [13], by generating new speech samples based on a spectrum of features. Traditional methods such as adding noise to the original waveform, are compared with the proposed methods. The results indicate that despite using an unimpressive unsupervised network for acoustic modeling, there was a significant decrease in WER.

GANs were first introduced by goodfellow in 2014 [21]. It was showcased as a novel framework for estimating generative models, making use of the adversarial process. The framework, corresponding to a two-player min-max game, required two models being trained simultaneously, where one model (G) would generate data based on a random distribution, while the other model (D) would discern the probability of the data being generated or from the original dataset. G's task is to maximize probability of D making a mistake, while D's task is to minimize the probability of G generating accurate predictions. The training for these models continues until D's probability stagnates at 0.5. This framework has yielded massive acclaim, and has paved a path for future research, specifically where

appropriately labeled data is limited, due to its potential to learn high-dimensional complex real data distributions.

Since the inception of GANs, they have received a high level of attention by the community, and have spawned numerous variants. A study conducted in 2019 [22] showcased some of the most popular variants of GANs, and how these variants function. The researchers also discuss the different applications of GANs, and ideal architectures for said applications. Moreover, they thoroughly discuss the various objective functions suggested by other researchers.

A year after said study was conducted, some researchers published an article [23] discussing different GAN architectures, with an emphasis on newer developments and evaluation metrics. They discuss various applicable evaluation metrics, and showcase the metrics by evaluating two of the more popular GAN architectures. The researchers also discuss noteworthy areas for future research into GANs, as well as interesting angles for certain applications.

2.4 Difficulty with training GANs

GANs are notoriously hard to train. Some researchers have discussed various issues with training GANs [14], and explore different strategies for better methods to train them. They propose 3 novel methods, all of which improve convergence and provide superior results.

An in-depth study into two of the most common issues in GANs [15], namely catastrophic forgetting and mode collapse was conducted in 2020. The study described these problems, and why they occur. After investigating the landscape of outputs by various GANs, the researchers had singled out the causes of these issues, and potential solutions.

Certain researchers have considered problems with GANs[16], and even attempted to solve them using manifold training techniques. They compare other solutions for mode collapse proposed in publications, and propose their own solution, which shows better results.

2.5 Human Interaction with Networks

In 2017, researchers introduced iDeepColorization [7], a novel approach into automatic colorization of an image using deep learned priors. They take an approach that is somewhere between the standard AI auto colorization and human guided colorization. Their network takes images for training, and for the first 1%

of the dataset, the network is provided with hints to the ground truth. This is treated as a classification task. Then, for the remainder of the training data, the network is provided with user input. The user selects a pixel to colorize, and the network guesses the most probable colors and provides the user with a palette to choose from. The user can then either choose from this palette, or give a color of their own desire, which is then treated as the ground truth for that pixel, and the training continues. This part of the training is treated as a regression problem. As a result of this mechanism, the final network is not only better able to colorize images in real time, but it can also take user input during the colorization process, thus giving users the ability to customize the final output. Moreover, this method allows unusual colorization, should it be desired by the user. The training process is flexible, and even allows input in the form of a globalized histogram of another image, thus allowing an image to be colorized based on the general colors of a different image.

Several state-of-the-art techniques are used for generating realistic images of faces from sketches. However, the methods tend to overfit to sketches. To improve on these techniques, the researchers developed the DeepFaceDrawing architecture [17], with the idea to implicitly model the shape space of plausible face images and synthesize a face image in this space to approximate an input sketch. The method involves taking input sketches as soft constraints. In the architecture, auto-encoders are used on the input sketch for learning feature embeddings for individual face components. The feature vectors are used for refining the sketch by projecting individual parts to respective corresponding manifolds. The feature vectors are also decoded and sent to the corresponding multi-channel feature maps, which are combined according to spatial locations. The final components are passed to an image synthesizer, which generates the image. This technique is currently the best in class for generating face images from face sketches.

To generate photorealistic images from semantic input, SPADE was introduced [18]. The architecture for conditional image synthesis involves a semantic segmentation mask, which is converted into a photorealistic image. To achieve this, the researchers proposed a spatially adaptive normalization layer (SPADE), which modulates the activations and can effectively propagate the semantic information across the network. SPADE is essentially a generalization of existing normalization layers. It performs better as it can maintain semantic information. The generator architecture consists of a series of SPADE residual blocks, with upsampling layers in between. The discriminator uses the same architecture from pix2pixHD. The objective function is also similar to that from pix2pixHD, but with a Hinge Loss instead of LSGAN loss. Comparing the results from SPADE to other state of the art techniques like pix2pixHD shows that SPADE not only is able to grasp more detailed semantic information, but is also able to show this in the generated images, which are more photorealistic and diverse. The primary goal behind this work is to enhance image generation and open new paths for photo editing.

When a problem consists of complex and rich data, it is often oversimplified into concise labels. This oversimplification can limit the ability of the network, and even skew results. In lieu of this, some researchers describe a method for knowledge elicitation from experts that can lead to more accurate labeling of data, without the risk of oversimplification [24]. This method involves a psychometric test, known as two alternative forced choices (2AFC). In this research, a modified version of 2AFC is used. The researchers prove how this technique involving 2AFC can lead to more accurate data augmentation. In various tasks, the quality of the image output is assessed via humans. However, this may not be possible at all times or for all tasks. Hence, model observers are used; networks that replicate human observers. The proposed method involves improved human observer responses in a two-alternative forced choice task.

A subject's decisions change based on the dynamic nature of their environment. This makes it difficult to understand the timescale of such changes, and strategies adopted by subjects. Recent studies show how subjects can achieve near optimal performance in a two-alternative forced choice task. Researchers have discussed how models compare with such tasks [19], and how their behaviors can be adopted for optimal policy making.

Various models that use two-alternative forced choice for decision making have been discussed [20], and after thorough analysis, an attempt was made to find optimality of such models under given tasks. Various predictions were made regarding the performances of such models.

With the increasing interest in robust autonomous decision-making mechanisms, such as the two-alternative forced choice navigation problem, several such decision-making models were considered [26]. They were studied using a variety of computational and analytical methods, and optimal strategies are proposed.

It can be difficult to define contrast in images with complex and nonlinear structure. GANs can be used for representing these images as a vector of nonlinear latent features. It is theorized [27] that the length of the vector correlates to the contrast of the image. This is confirmed using observers through a two-alternative forced choice task.

2.6 Genetic Algorithms (GAs)

An in-depth understanding on Genetic Algorithms has been detailed by a publication in 1994 [28]. The authors discussed GAs in general, followed by a detail description and tutorial on how GAs work. The authors provided some experimental forms of GAs, including parallel island and parallel cellular models. The publication also discussed genetic search by hyperplane sampling.

For a wider view of Genetic Algorithms, a recent study by Katoch [29] provides a better understanding. The paper describes the more prominent GAs in great detail, along with implementations and their pros and cons. In order to inspire new developments, the genetic operator and their usages are also discussed in length. The study also focuses on recent advances on GAs in the community.

GAs have been successfully used by researchers to search the latent spaces of GANs to provide higher quality images for small and difficult datasets [30]. In particular, the proposed framework improves quality while preserving the original generators diversity. Preliminary results indicate that human observers preferred the new models' outputs.

Neuroevolution is a relatively new idea, which involves using a GA in a GAN for improved generation of data [31]. The researchers proposed a new technique combining neuroevolution and coevolution in the training of a GAN. Training results indicate the large extent of improvement gained by using this new technique. The study also discussed how this new technique deals with the issue of mode collapse.

2.7 Evaluation Metrics

Due to a lack of any consensus on which evaluation metric is the best when comparing GANs, researchers have decided to review and critically discuss more popular metrics in an attempt to find optimal metrics [32]. The review discusses the benefits and issues with using certain metrics with GANs.

Various distance functions between conditional distributional functions are considered, with the intent of finding the ideal evaluation metric for comparing GANs [33]. Specific focus is given to the Fréchet Inception Distance (FID). Various conditional versions of the metrics are developed and analyzed.

Some researchers [34] have proposed a multiscale structural similarity method to assess image quality. Taking influence from the structural similarity image quality paradigm, they developed a method which supplies more flexibility than previous methods in incorporating the variations of viewing conditions.

2.8 Research Gap

Currently, there is only one paper available where human feedback is interactively taken during the network's training process, and directly fed into the network, with the intent of improving the training process. This iDeepColorization framework [7] is unfortunately restricted to its domain and is specific to one application. It cannot be extended to other applications as is.

The motivation for this thesis lies in the lack of work in this area.

2.9 Problem Statement

Data augmentation is one of the solutions to various problems with respect to fulfilling the need for annotated data for training deep neural networks. Specifically, where labelled data is scarce, data generation through GANs is a plausible solution.

In GANs, specifically the discriminator, is difficult to train and keep in balance with the generator. 2AFC, or other techniques for including a human rater in the training process can be used to improve the training process of GANs, and hence produce more accurate and realistic generated data.

Papers	Learning	Architecture/Framework	Objective	Domain
[8]	Supervised		Create a method to synthesize data in lieu of limited dataset	Medical (Ultrasound)
[10]		Conditional GAN (pix2pix)	Create a method to synthesize data in lieu of limited dataset	Medical (MRI)
[11]			Improve emotion classification through data augmentation	Sentiment (Facial Emotion)
[12]	Semi - Supervised	ACGAN	Machine fault diagnosis by managing an unbalanced dataset	Mechanical (Sensors)
[13]	Unsupervised	Conditional GAN	Improved speech recognition by generating noisy samples using GANs	Speech Recognition
[14]	Unsupervised	Custom	Provide techniques for improved GAN Training	Image Generation, Image Translation
[15]			Study common issues in GANs, and discuss potential solutions.	Image Generation
[16]	Unsupervised	DCGAN, LSGAN, DRAGAN, DFM	Discuss mode collapse and effective strategies to prevent it	Image Generation
[21]	Unsupervised	Conditional generative	Introduce novel idea of GAN	Image Generation
[22]	Supervised, Unsupervised, Semi - supervised	DCGAN, Hierarchy, Auto encoder	Describe GAN variants	Image Generation
[23]	Unsupervised, Semi - supervised	Convolutional, Conditional, Autoencoder	Describe GAN variants	Image Generation
[17]	Unsupervised	Custom	Generate photo-realistic face images based on face sketches	Image Generation
[18]	Unsupervised	Multi-modal	Accurate image synthesis through semantic input	Image Generation
[30]	Unsupervised	Multi-modal	Improve GAN training using GA	Image generation
[31]		NEAT	Improve image quality	Image generation

Table 1: Comparison of different GANs structures

Papers	Performance Metrics	Generic Solution	Dataset
[8]	FID, Jensen - Shannon divergence, SSIM	No	IVUS
[10]	Dice Score	No	ADNI, BRATS
[11]	Accuracy	No	FER2013, SFEW, JAFFE
[12]	Accuracy	Partially (Within domain only)	Custom
[13]	WER	No	Aurora4, AMI-SDM
[14]	IS, FID	Yes	CIFAR-10, ImageNet
[15]	Landscape appearance	Yes	CelebA, CIFAR-10, MNIST, 8 Gaussian
[16]	MS-SSIM	Yes	CelebA, CIFAR-10
[21]	Log - likelihood	Yes	MNIST, CIFAR-10, TFD
[22]	Human Observation	Yes	
[23]	MS, FID, MS-SSIM, SS-SSIM	Yes	
[17]	FID, IS	No	CelebAMask-HQ
[18]	mIoU, Accuracy, FID	Partially	Cityscapes, ADE20K, COCO-Stuff, Flickr Landscape
[30]	Koncept512	Partially (Within domain only)	Pokemon, KonIQ-10k, StyleGAN2
[31]	FID	Yes	MNIST

Table 2: Comparison of different GANs Implementations

2.10 Research Objectives

- Training GAN with human feedback
- Using a GA to optimize weights based on human feedback loss
- Create generic framework

3. Methodology

3.1 Model of the Proposed Solution

The general flow of the proposed solution is outlined below, in the Figure 3.

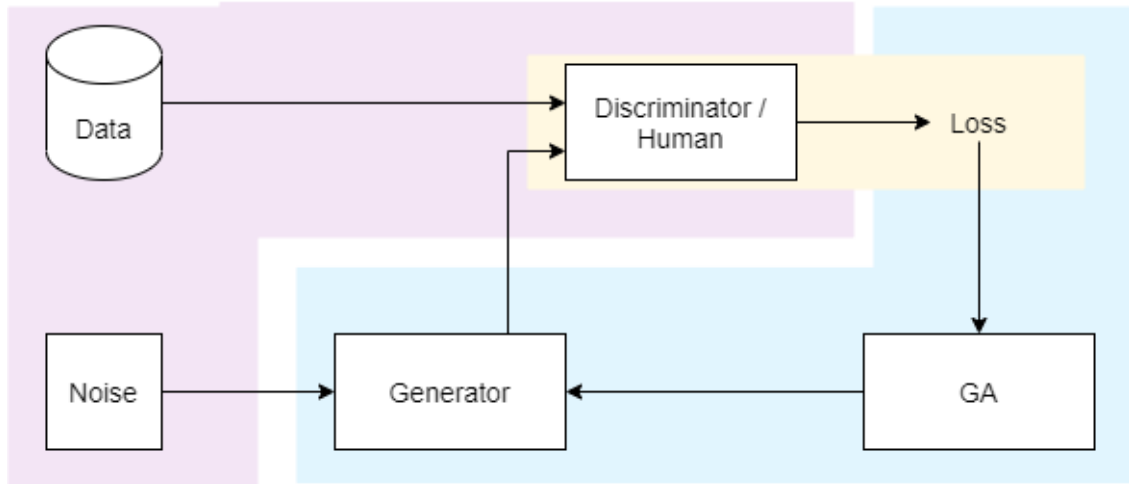


Figure 3: Framework of the proposed solution

As seen in the Figure 3, the architecture can be split into three basic components. The purple segment represents the standard GAN architecture, focused on adversarial training. The blue segment represents the neuroevolution portion to train the generator. The yellow segment represents the human involvement, which will have the switch between discriminator and human, and will provide the specialized loss to the GA segment.

As seen in the purple segment, the discriminator will be fed images from the dataset along with one generated by the generator (using gaussian noise for image generation). The discriminator will determine how similar the generated data is to the dataset, and provide the adversarial loss. This loss will be used to train the discriminator, similar to standard GANs.

As seen in the yellow segment, a human element is involved. For this framework, a hybrid approach will be practiced, where for half of the dataset, instead of the discriminator a human will play the role of determining the realism of the generated datapoints. This can be in the form of a 2AFC task. Multiple data points will be selected, of which some are generated and some are from the dataset, and the human will be provided with sets of two datapoints and asked to select the datapoint that appears to be more likely from the original dataset. This will result in the data being heuristically arranged as a scaled result, which can be used to generate a specialized loss. This new loss will be fed into the GA.

As seen in the blue segment, the GA will take this specialized loss, and based on a fixed population size, select network weights, then converge and mutate the network weights based on the loss. The networks weights will be adjusted based on the result of the GA. This should push the generator to produce more realistic datapoints.

This way, the network will continuously be trained. As the learning rate continues to drop, the system will take fewer inputs from the user. But when the learning rate begins to stagnate, the prompts for user input will increase. If the learning rate does not significantly change despite user input, then the training is concluded.

3.2 Dataset and Implementation Details

GANs are being used in a number of domains for various tasks. Among them, two in particular stand out. That is medical imaging, and facial imaging.

The techniques discussed in this thesis will be evaluated in both domains, with “Pulmonary Chest X-Ray Abnormalities” dataset from the medical domain, and “Fakefaces” dataset from the facial imaging domain. Both are available at Kaggle. Beyond these datasets, there are other even more important datasets. However, for the sake of this thesis, evaluation will be limited to popular open databases.

The proposed framework is being implemented in python using pytorch, and will be run using NVIDIA GPUs. A version of the framework compatible with colab, is also being tested.

For evaluation purposes, the proposed framework will be tested against DCGAN using MS-SSIM and FID as metrics [21] [22] [23]. Human observers will also be scoring the results of the output [22].

3.3 Evaluation Metrics

Based on exhaustive research, two of the most optimal evaluation metrics for GANs include FID and MS-SSIM [32].

3.3.1 MS-SSIM

Among the various methods for perceiving the quality of a digital image, the structural similarity index measure (SSIM) is one of the most popular. It is used for measuring the similarity between two images.

Multiscale-SSIM (MS-SSIM) index is an extension of SSIM, first presented in [34]. MS-SSIM supplies more flexibility than SSIM in various viewing conditions. Since its inception, it has hence become one of the standards to evaluate image quality.

The formula to calculate MS-SSIM index is shown below, in the Equation 1.

$$MSSIM(x, y) = \frac{1}{M} \sum_{j=1}^M \left(\frac{(2\mu_x \mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \right)$$

Equation 1: MS-SSIM index calculation

3.3.2 Fréchet Inception Distance (FID)

The Fréchet Inception Distance (FID) is a metric specifically devised to assess images generated by a GAN, and was conceived as an improvement to the inception score (IS).

The FID evaluated the distribution of generated images and compares the distribution with the distribution of real images which were used to train the generator [33].

The FID in essence measures the diversity of the images generated by the GAN, as well estimating how meaningful is the object in the generated image.

The formula to calculate FID score is shown below, in the Equation 2.

$$FID = \|\mu_r - \mu_g\|^2 + \text{Tr} \left(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2} \right)$$

Equation 2: FID score calculation

3.4 Generative Adversarial Networks (GANs)

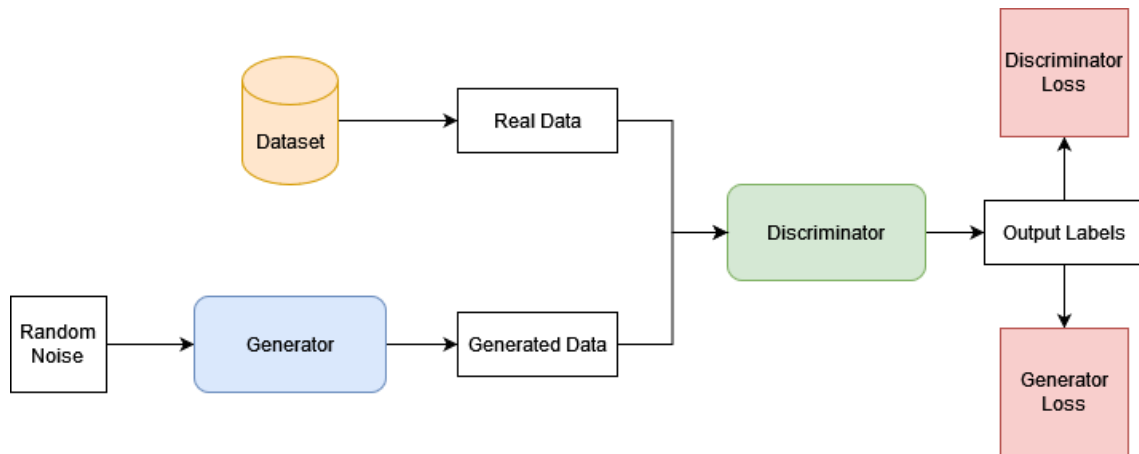


Figure 4: Architecture of a Generative Adversarial Network

Generative Adversarial Networks (GANs), are essentially a framework of a pair of neural networks, that specialize in generating data. They can generate almost any type of data, from digits to photo-realistic images of landscapes or people. Their ability to approximate real data is what has made them so popular over the last few years.

GANs were first introduced by Goodfellow [21], which has formed a basis for numerous variants of GANs, each specialized for different tasks.

The framework is designed such that one network will learn to generate the data (G), and the other network will learn to discriminate between the generated and real data (D). This follows a two-sum game, where the generator continuously tries to generate better data, and the discriminator continuously tries to discern the real from the fake data.

Overtime, both networks are trained till the discriminator can only differentiate between the real and fake data about half the time.

Figure 4 shows the workings of a standard GAN.

3.4.1 Discriminator

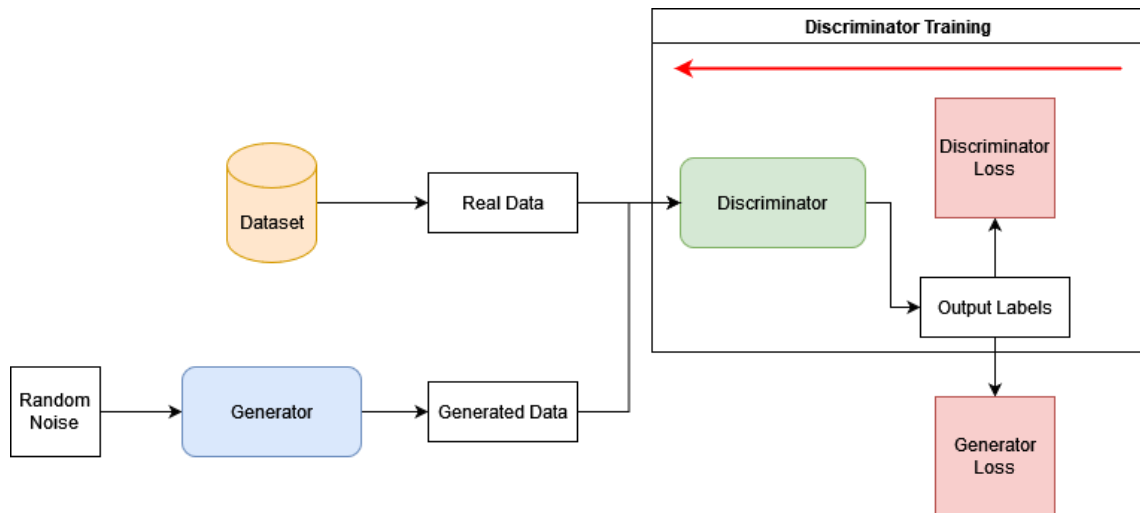


Figure 5: Training the discriminator in a Generative Adversarial Network

Within a GAN, the discriminator attempts to discern generated data from real data. Essentially, it acts as a binary classifier. It is passed with data points along with a label. The label provides a value of 1 or 0, where 1 implies that the datapoint is real, and 0 that the datapoint is fake (generated).

For proper training, the discriminator is trained against both the real dataset as well as the generated one. The contrast between the classifications of these sets is what allows the discriminator to improve itself.

While the discriminator is trained, the generator is made to wait.

Backpropagation of the discriminator is done such that the generator is not allowed to hold onto the loss values generated. After the discriminators loss has been backpropagated, the generator beings its training and loss backpropagation.

Please see Figure 5 for a diagrammatic description of the process.

3.4.2 Generator

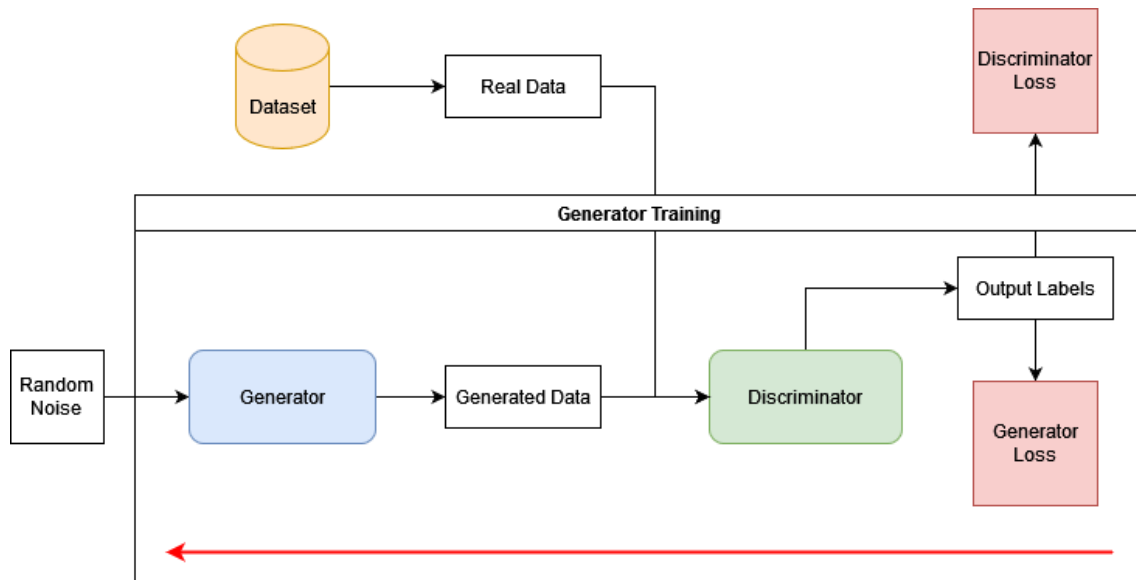


Figure 6: Training the generator in a Generative Adversarial Network

After the discriminators loss has been backpropagated, the generator can begin.

The generator produces fake data (approximate to the real data) from an input of random noise. Its goal is to continuously produce more realistic data. To do this, it requires the discriminator to ascertain how accurately the fake data can approximate the real one.

The generator and discriminator are linked in a two-sum game, and continue to operate in such a manner. The generator continues to get feedback from the discriminator and continues to tweak its results. The loss the discriminator notes regarding the generated data is backpropagated by the generator, and thus the iterations continue.

Please see Figure 6 for a diagrammatic description of the process.

This is one of the more important steps in GAN training. This is also the reason why GAN training is difficult. This means that it is difficult for a GAN to converge to an optimal solution. We will get into those details as well later in the article.

3.4.3 Fitness

$$L(\theta^{(G)}, \theta^{(D)}) = E_x \log D(x) + E_z (1 - D(G(z)))$$

Equation 3: Loss function in GANs

Equation 3 describes the loss function used in traditional GANs, where the networks attempt to optimize the function in a two-sum game. It follows a min-max process, where the discriminator attempts to maximize the $\log D(x)$ term, while the generator attempts to minimize the $\log(1 - D(G(z)))$ term.

3.5 Genetic Algorithms (GAs)

Genetic Algorithms (GAs) are a branch of optimization techniques originating from the influence of genetics and natural selection in modern computing. They are used to find approximate solutions to complex tasks, and are commonly used where the standard solutions would be inefficient or ineffective.

GAs involve applying evolutionary concepts to data. Multiple datapoints collect to form a population of samples, where each sample is a unique datapoint. The members of the population undergo crossbreeding and mutation, producing offspring samples which though similar to their originating parents, are somewhat different. The process is designed such that after the crossbreeding is done, the child sample maintains the best genes of the parents, and that the mutation allows variation among the child samples.

A fitness function is used to determine which sample are better than the others, and as the saying goes, “survival of the fittest”, the fitter samples are kept while the unfit samples can be discarded. This way, multiple iterations are run, till the fittest samples are near perfect for the task.

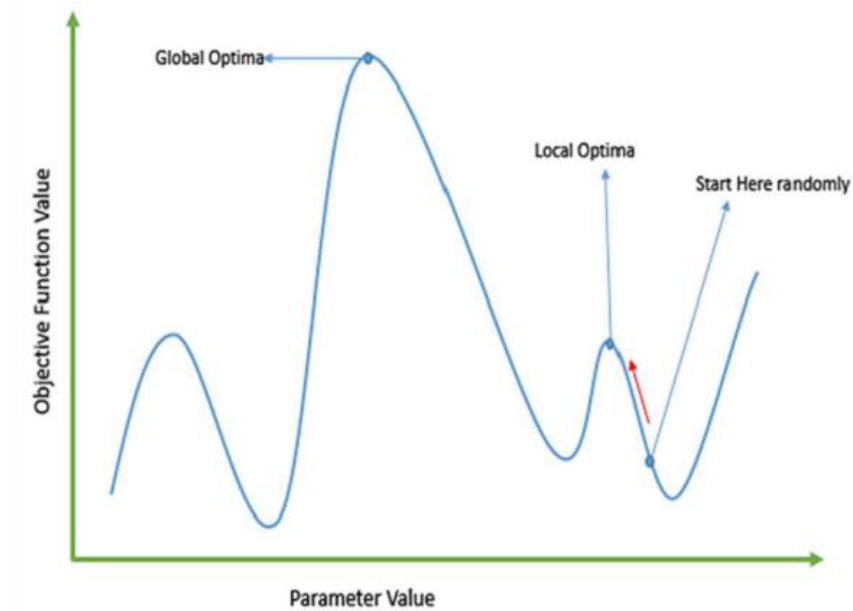


Figure 7: Failure of Gradient Based Methods

The best feature of GAs that allow them to tackle these complex tasks, is their random nature. Samples are crossbred randomly; their genes swapping for breeding is random; their gene mutation is random. Randomness, within defined parameters, allows a broader search of the optimal solution in a feature space.

GAs can often perform even better than traditional methods in certain tasks, primarily thanks to their random nature. This is because, given a certain task, the traditional method (such as gradient descent) can often get stuck in a local optima. The solution being produced is sub-optimal, but the system cannot register that it is stuck in a local optima. GAs can jump around in the feature space, and hence potentially recognize a local optima from a global optima. Figure 7 shows how traditional methods can fail to recognize local optimas.

3.5.1 Initial Population

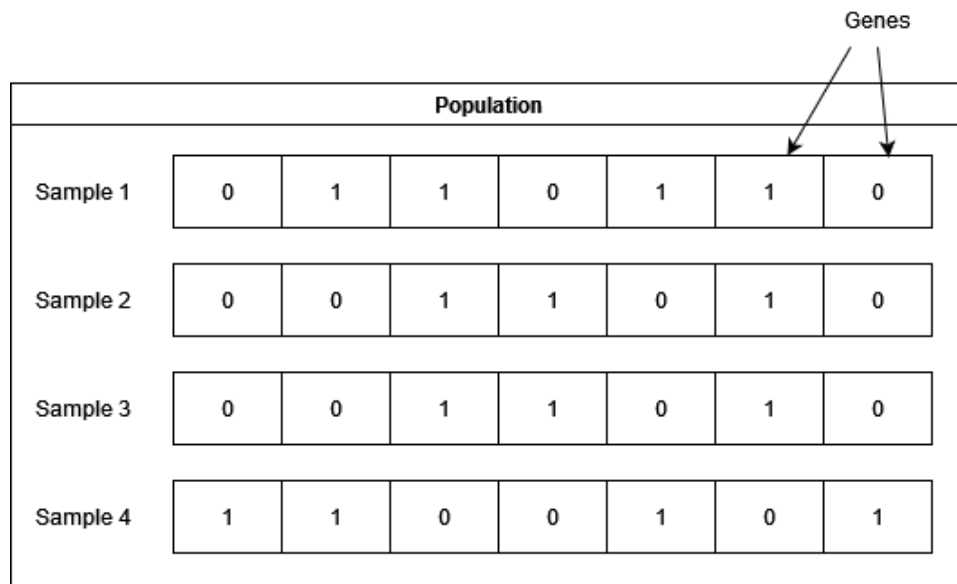


Figure 8: Population, Chromosomes and Genes

To start working with a GA, we first need an initial population. This is a collection of multiple individuals, each with multiple genes. An individual represents a possible solution to a given task, and the genes represent the parameters of the solution of the given task (see Figure 8).

Each individual in a population must represent a possible solution to the problem, and each individual must be unique. This means that the variation between the individuals would be a result of the variation of their gene pool. The number of parameters (genes) of each individual must be the same.

3.5.2 Fitness Function

A method needs to be put in place to ascertain the quality of each individual, with respect to the given task. The fitness function, is designed to take an individual and provide a metric which would define the “fitness” of the given individual. The fitness of each individual is essential, as it would determine its survival, as well as its likelihood to crossbreed with other individuals.

3.5.3 Selection

The GA begins after an initial population is created. The first step is typically to find the fitness of each individual, and select the best individuals for further processing. Pairs are created based on their fitness values, and then passed on for crossbreeding.

3.5.4 Crossover

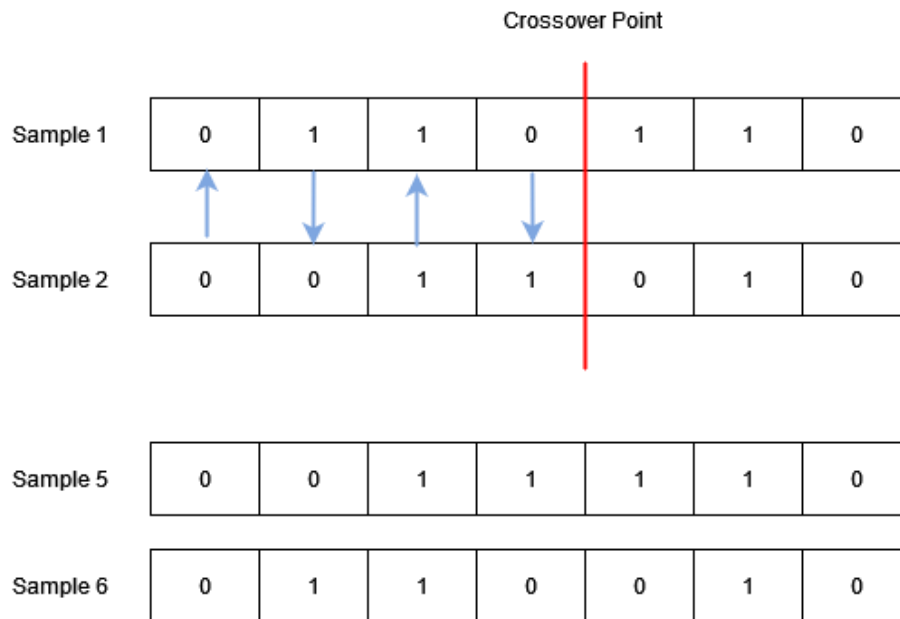


Figure 9: Exchanging genes among parents on crossover point

Crossover is one of the primary functions of a GA, and hence the quality of the crossover has a huge impact on its success.

The pairs of individuals provided to the crossover method after selection, are made to crossbreed. A crossover point is selected, typically at random, and the genes from one individual are swapped with the genes of the other individual. This swapping is done from the start, till the crossover point. Beyond the crossover point, the individuals maintain their original genes, as seen in Figure 9.

This method allows individuals to share their features with other individuals, and by the end, produce new variants of themselves.

The idea is that based on random chance, crossover can allow a unique set of genes to be created from a set of fit individuals, that would be better than the original individuals.

There can be numerous ways to crossbreed individuals, and often a GAs success is tied to how the crossover function operates in dealing with the individuals and crossing specific genes.

As an example, a two-point crossover can also be performed, where two crossover points are set. Here, the genes between the two crossover points remain the same in the individuals, while the other parts are swapped between individuals. This is essentially to allow a more varied crossing of genes.

3.5.5 Mutation

Sample 5	0	0	1	1	1	1	0
Mutated Sample 5	0	0	1	1	1	1	0

Figure 10: Mutation: Before and After

Crossover alone is not enough to solve a given task. With only crossover, the solutions accessible in the feature space are limited, as we are only considering variants of existing solutions. Mutations allow enough randomized variation to explore a larger field in the feature space, while remaining within certain confines to allow a directed search for the appropriate solution.

Mutations are created by changing some of the genes of an individual randomly. A random number of genes, and changed to a random value. Given simpler tasks where genes are either 1s or 0s, they can be flipped, as seen in Figure 10.

Mutations allow diversity in the possible solutions, and prevent premature convergence.

4. Implementation

The proposed framework consists of using a genetic algorithm (GA) to improve the results of a standard generative adversarial network (GAN). Two-alternative forced-choice (2AFC) technique will be used in tandem to assist the selection process of the GA as a pseudo-discriminator. A generalized step-by-step process can be seen in Figure 11 and Figure 12.

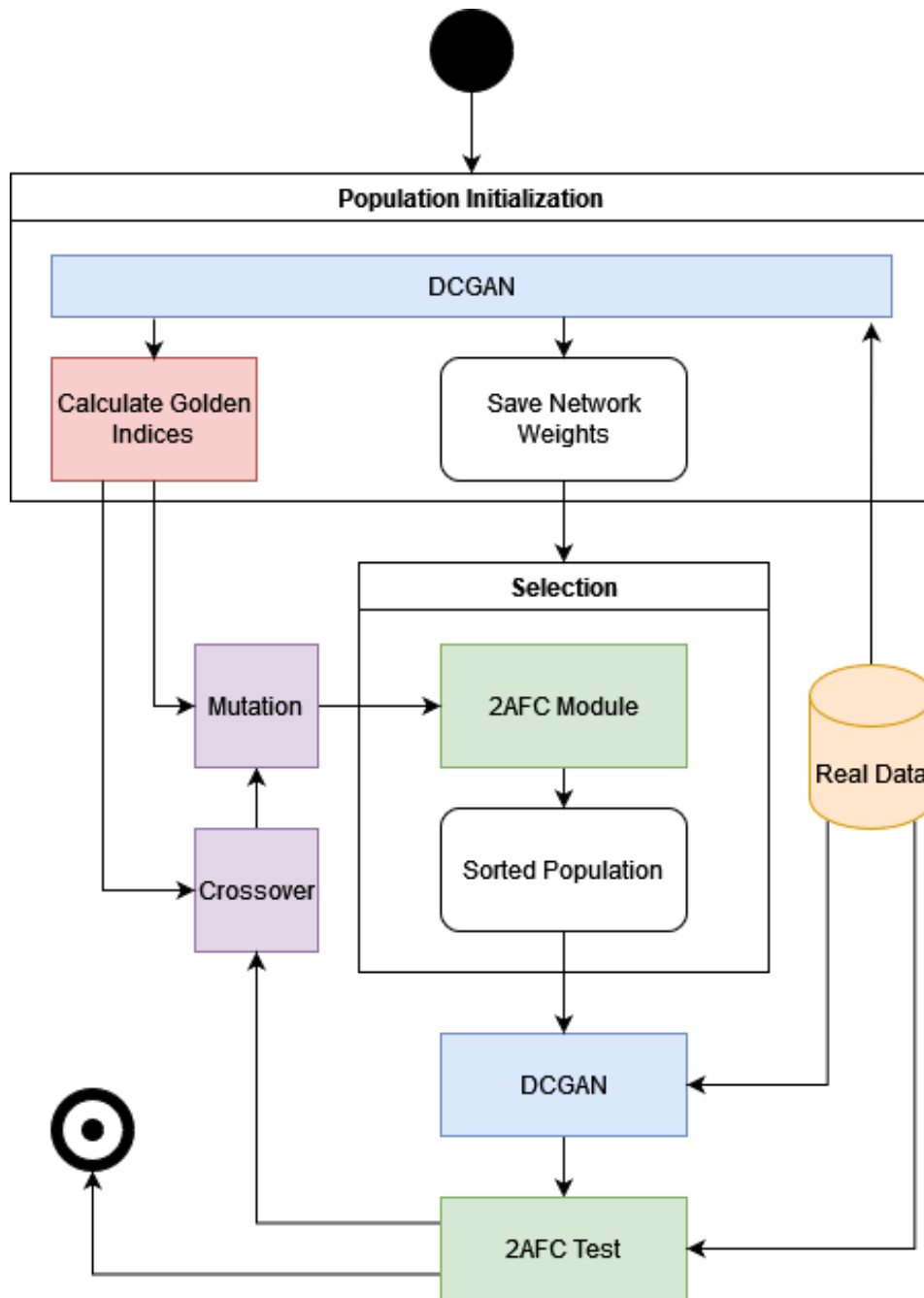


Figure 11: Proposed Framework Pipeline

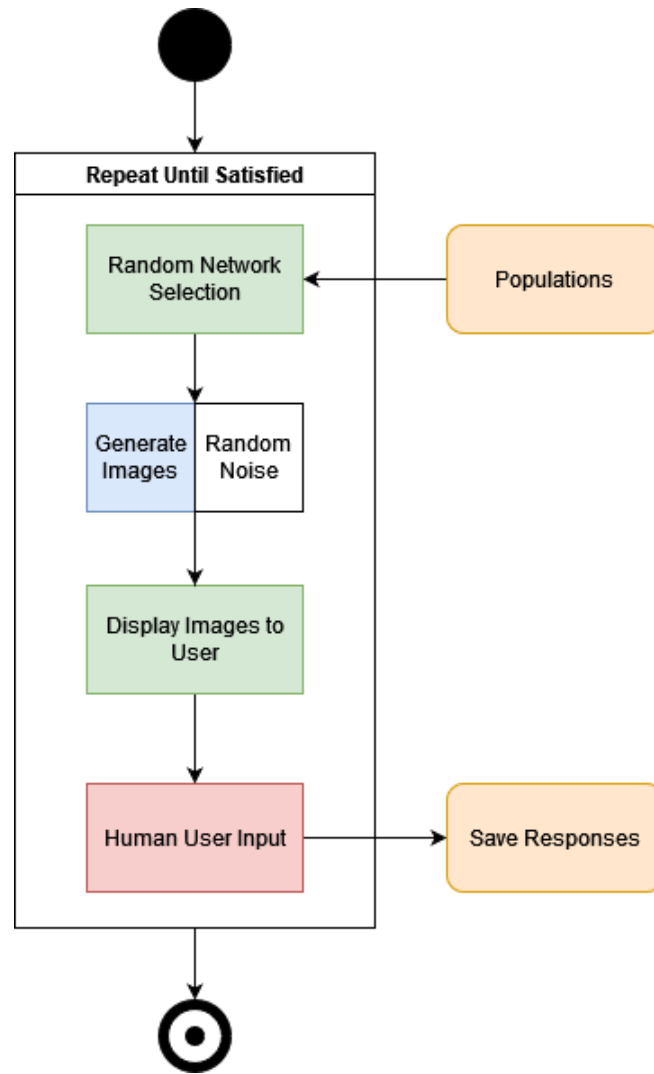


Figure 12: Two-Alternative Forced-Choice (2AFC) Process

4.1 Population Initialization

As seen in Figure 11, this is the first part of the pipeline. An initial population needs to be created. To do this, a standard DCGAN is run on the CelebA dataset. This process results in the training of a single network for the dataset.

As the training progresses across multiple epochs, for each epoch, a copy of the network's parameters is saved. Each of these epoch-level network parameters are samples of our initial population. So, if the network is trained for 5 epochs, 5 samples will be generated for the initial population.

This process is repeated multiple times, all up till the same epoch. The reason for running multiple different epochs is for diversity. As the network backpropagates per epoch, small changes are seen across the various weights of the network, all eventually fitting into the appropriate space to provide the best results. However, this means that the changes between two epochs are

negligible. Hence, for a more diverse sample size with respect to the latest feature vectors, the training is repeated multiple times.

We don't want to run too many epochs per training session, as this training is a slow and resource intensive task. We wish the GA to be able to make up for the networks lack of training.

During our testing, the training was run 6 different times, each up till the 4th epoch.

After running the DCGAN for several sessions, we end up with a population of 24 samples. Each being a 1D vector for roughly 2.5 million floating point values, which represent the weights of the network. This will be the population going forth.

4.2 Golden Indices Calculation

After the population initialization phase, one other important step is carried out, that is the calculation of the “Golden indices”, as see in Figure 11.

During our testing, the following unique observations of the samples were made¹:

- There is a small percent of weights (less than 5%) that have the most extreme values (greater than 80% or less than 80%) across epochs. Let's refer to these as “heavy weights”.
- The indices of some of the “heavy weights” remains consistent across multiple epochs of training. Let's refer to these as “consistent heavy weights”.
- There is a small set of weights (less than 10%) that are having the most extreme changes (increased by 60% or decreased by 60%) across epochs. Let's refer to these as “changing weights”.
- The indices of some of the “changing weights” remains consistent across multiple epochs of training. Let's refer to these as “consistent changing weights”.
- The intersection of the “consistent heavy weights” and the “consistent changing weights” provides us with what we refer to as the “Golden Indices”. These are those weights that have both persistently remained as dominantly large or small, and been persistently been changing heavily across epochs.
- The average absolute change in weights across all weights was recorded to be around 0.02
- The average absolute change in weights across all weights of the

¹Specific to the CelebA dataset, when running DCGAN. We expect a similar pattern for other datasets and GANs.

“Golden Indices” was recorded to be around 0.2

Our belief is that these weights of the network are significant in terms of our current task. Since the resulting “Golden Indices” is a latent vector of less than 10,000 values, our process is significantly sped up, as now we can ignore several million weights.

Hence, our pipeline analyses all samples during the population initialization phase, and silently calculates the “Golden Indices” for all samples, and then takes their union. Moving forward, several process will heavily involve these “Golden Indices”.

Please see Figure 13, Figure 14 and Figure 15.

4.3 Population Sorting

After we have created our population consisting of 24 unique samples, we need to sort them, as seen in Figure 11.

A traditional GA has the concept of “selection”, where the best n-percent of the population are selected based on some heuristic value, and used for future processes. This process is exceedingly difficult for certain tasks, for instance when working with images.

As a test case, we are working with the CelebA dataset, on creating realistic human faces. Which means, the measure of our works success is tied with how realistic our generated images look. This notion of “realistic” is difficult to pin to a numeric value. To begin with, what exactly is “realistic”, and what factors a part of what makes something “realistic”?

In such a task, various standards such as MS-SSIM and PSNR would be calculated to measure the “quality” of the resulting image, which would be part of what makes up a suitable heuristic for how “realistic” an image is. In simple terms, this is a difficult task.

The benefit of using a GAN, is that all these factors that are tied to the “realism” of the generated image, are all automatically part of the numerous hidden numerical values of the neural network. The issue lies in the complexity in finding those ideal set of numbers, and in identifying their exact relation to the required task. Using a GAN allows us to get an approximate answer in a relatively shorter time.

This is where we introduce the two-alternative forced-choice (2AFC) technique. Since the purpose of the GAs selection is to use some fitness heuristic to compare different samples of the population and select the best, we can simplify this entire process using 2AFC.

We have devised a 2AFC process for comparing the various samples, described in chapter 4.4.

For our pipeline, at this stage all the samples are sorted from best to worst using the 2AFC process. After sorting, the best sample is referred to as the “Golden Network”

4.4 Two-Alternative Forced-Choice (2AFC) process

The nature of 2AFC is to show two different samples, and to select one of the two. This task makes the process simpler, particularly when a human operator is concerned. A human doesn't need different heuristic or metrics. A single glance at two images, and they can instantly tell which is more realistic, often more accurately than machines can. Humans do this automatically, and instinctively. Even when both samples are realistic enough to fool even state-of-the-art systems, a human can accurately differentiate them.

As seen in Figure 12, the proposed framework will take two of the samples randomly from our population, generate an image per network from the same input random noise, and display them side by side to the human operator. The human operator needs to select the image that they believe to be more realistic.

Based on a tournament style sorting algorithm, all samples will be tested against each other to rank them from most realistic to least realistic.

Here we make an assumption that if the same human operator is comparing images generated by multiple samples, then the samples have a transitive relation with respect to their rank. For instance, if we have images generated from samples A, B and C, and a human deems A's images to be more realistic than B's, then sample A is better than B. Now when comparing B's and C's images, if B is considered to be more realistic than C, then logically the same human operator would deem A to be more realistic than C, and hence A and C do not need to be actively compared.

Keeping this transitive relation in mind, a smarter comparative algorithm can be used. This would allow fewer comparisons, and thus not only compute faster, but put less work on the human operator.

To assure that the samples are accurately compared, during each sample comparison, they are compared more than once, each time using a different noise vector as input. This needs to be carried out an odd number of times. For our pipeline, samples were compared 7 times.

4.5 Crossover

After the population has been sorted, the crossover process begins, as seen in Figure 11.

Unlike a traditional GA, where pairs are randomly selected from the population for crossover, our method involves keeping the “Golden Network” as one of the samples and crossing it with every other sample in the network. Also, the genes/weights of the “Golden Network” are copied to the other sample, but its own genes/weights will remain unchanged.

For the current pipeline, a one-point crossover was performed, however a multi-point crossover can also be performed for better results.

However, the crossover of genes/weights is only performed for the “Golden Indices” of the samples. The rest of the genes/weights are not touched.

The resultant samples are saved separately, referred to as children. The original samples, referred to as parents, are kept intact.

4.6 Mutation

As seen in Figure 11, after performing crossover, mutation is carried out. This is done on the resultant samples generated after the crossover operations. The parents are not affected.

For this process, a different level of mutation is performed depending on which genes/weights are being considered, based on our earlier observations.

For the genes/weights of the “Golden Indices”, an absolute change between 0.1 and 0.5 was randomly made. This was carried out 200 times, each time at a randomly selected location from the “Golden Indices”. Whereas for the rest of the genes/weights, an absolute change between 0.01 and 0.05 was randomly made. This was carried out 10,000 times, each time at a randomly selected location from all the genes/weights.

4.7 Selection

After both crossover and mutation has been carried out, selection is performed, as seen in Figure 11. By this point in the pipeline, we have 24 samples which were of our original population, labeled as parents, and 24 new samples which have undergone a process of crossover and mutation, labeled as children.

Through the 2AFC process, the parents and children will be ranked, and the top half parents and the top half children will be selected for the next batch of

operations. For this, each of the top halves are saved together as the new population, and the rest of the samples are discarded.

4.8 DCGAN training the Golden Network

This is the step after selection, as seen in Figure 11.

The top ranked sample in our current modified population is our new “Golden Network”. These weights of this sample are saved as a generator of a standard DCGAN. We take the discriminator of our previously trained networks (at epoch 4), and use it in tandem with our generator.

We process the DCGAN for one epoch of the dataset. For faster processing, we can limit the epoch to a fraction of the dataset, or even limit the number of times the discriminator is trained (the generator will be trained every iteration irrespective of the discriminator).

After the DCGAN has been successfully trained for one epoch, the generator weights are extracted and replaced with our previous “Golden Network” weights, while the discriminator is kept ready for the next iteration.

4.9 Testing Results

The final step of the pipeline shown in Figure 11, is to test the results produced so far.

The best sample from the current population (“Golden Network”) is selected, and made to generate images. These images are compared with the actual images of the dataset, using the 2AFC process. This is repeated an odd number of times. If the human operator is fooled by the fake images more than half of the time, then the process ends, and the final weights are saved. Otherwise, the process repeats itself from step 4.3, where the now new population needs to be sorted.

Alternatively, a limit to the number of training iterations can be set, in which case the results would have to be manually compared.

5. Results and Discussion

5.1 Observations during testing

As mentioned in section 4, the experiments were carried out using a standard DCGAN as a base. 20 different networks were created using the DCGAN on the CelebA dataset, all of varying training levels. The lowest was trained till epoch 1, while the best was trained till epoch 4. These 20 formed the initial population for our experiments.

As noted in section 4.2, some peculiar behavior was observed regarding the networks, as seen in Figure 13, Figure 14 and Figure 15.

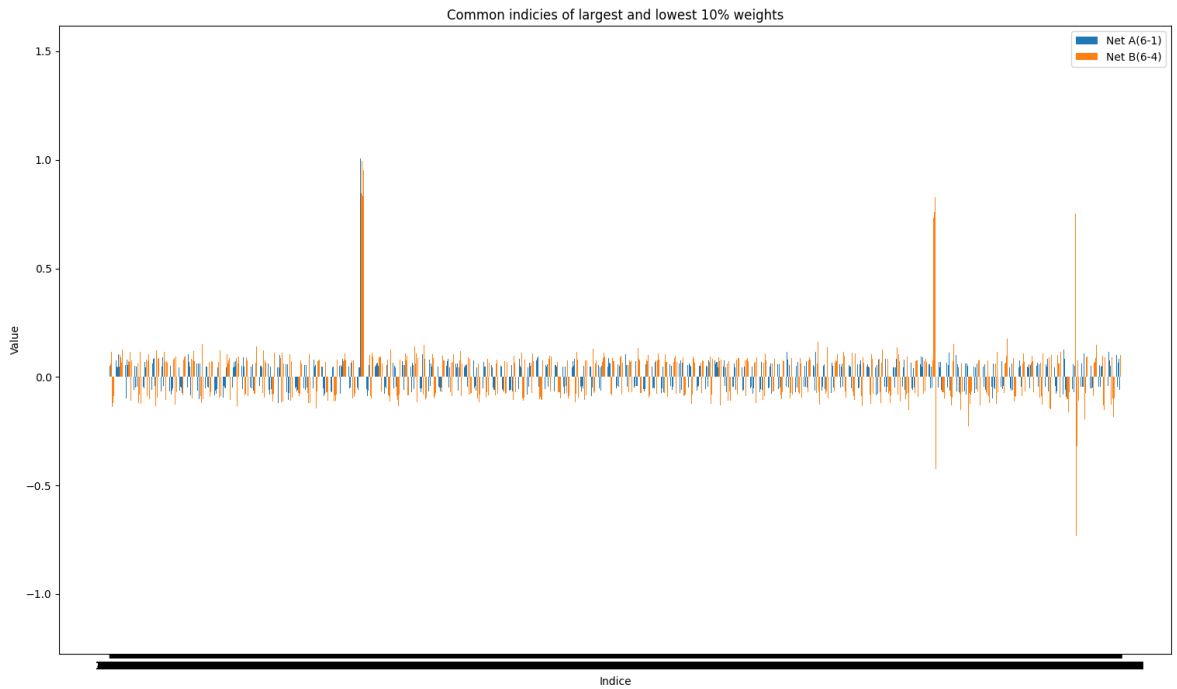


Figure 13: Common Indices of extreme weights

From our analysis of the networks and this behavior, we have concluded that the generator has certain weights that influence our face generation task more greatly. These weights are those that are consistently (across several epochs of training) on the extreme ends (very large or very small as compared to other indices) across several epochs of training. These indices can be found across different initializations of the original network. These indices are referred to as “Golden Indices” in this thesis.

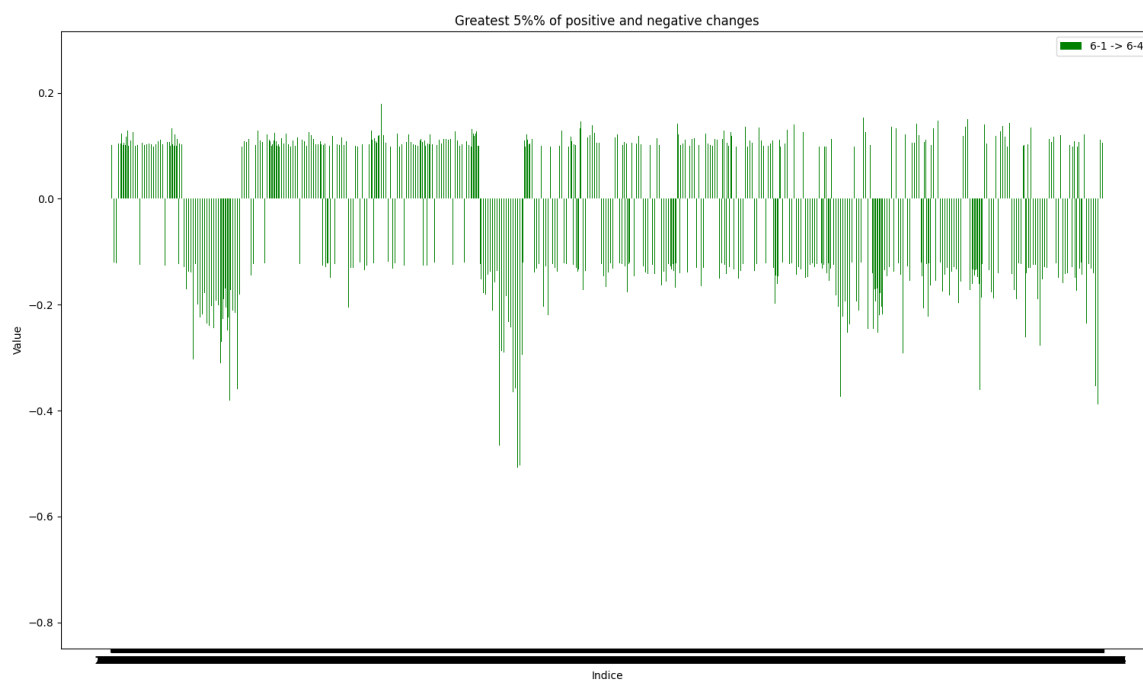


Figure 14: Greatest changes in weights across trainings

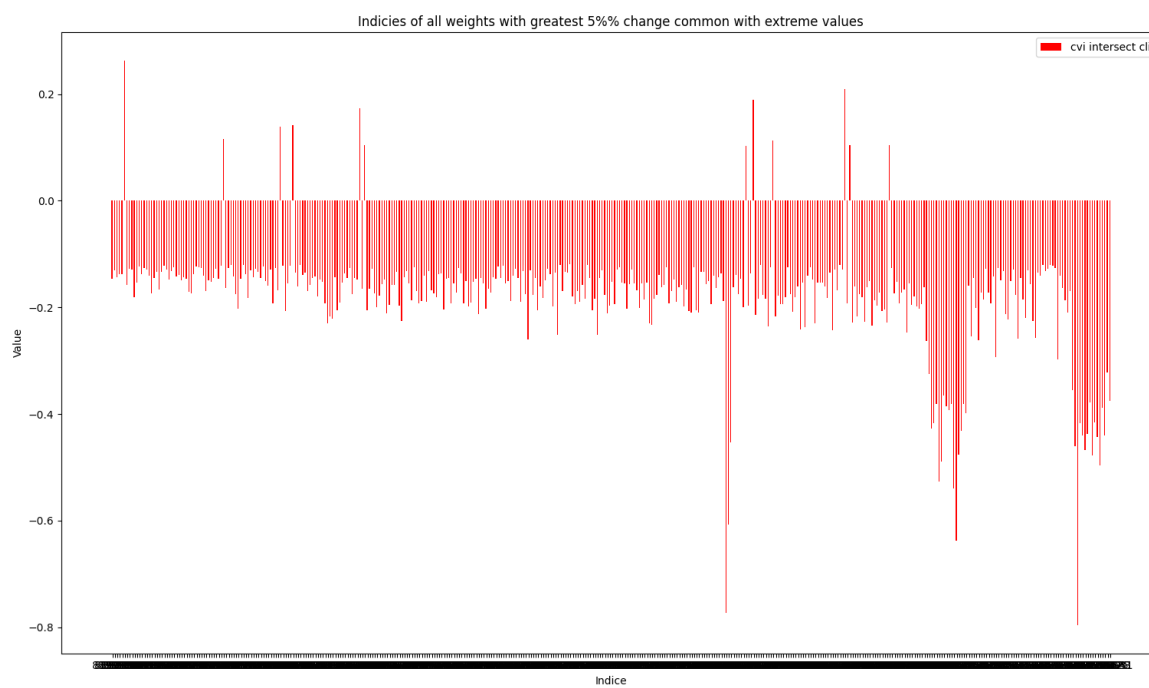


Figure 15: Greatest changes in common Indices of extreme weights

Further testing revealed that if the “Golden Indices” of a network are influenced, then the network begins to show signs of changes, while visually maintaining the appearance of faces. However, if the other indices are influenced in the same manner, then the image deteriorates. This can be seen in Figure 16.



Figure 16: Influencing weights of specialized indices

The figure can be interpreted as follows: left to right, top to bottom; original network, original with crossover and mutation on golden indices, original with crossover and mutation on other indices, original with crossover on other indices and mutation on golden indices.

From here, it is obvious that the major changes need to occur in the “Golden Indices”. However, without some changes in the other weights, there won’t be any significant progress.

Testing revealed that in the “Golden Indices” an absolute average change of ~ 0.2 was noted cross each epoch of training. Meanwhile, the other indices noted an absolute average change of ~ 0.02 across each epoch of training. This gives us a benchmark for mutations in the network.

Accordingly, the appropriate amount of mutation was created in the appropriate indices.

5.2 Qualitative Results

After even 5 iterations, a visible improvement can be seen in the “Golden Network”, as seen below in Figure 17.

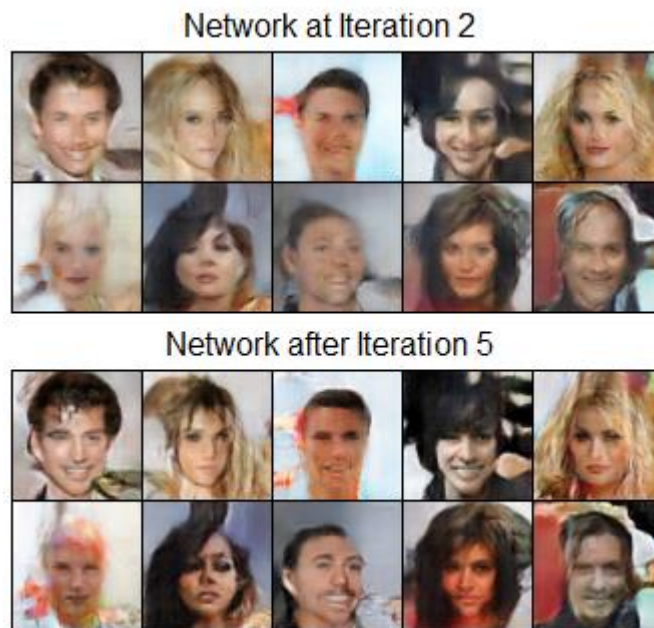


Figure 17: Best results comparison

If we consider the other networks in our population, we can see from Figure 18 and Figure 19, that despite not being considered as the best network, they are fairly better now as compared to before.

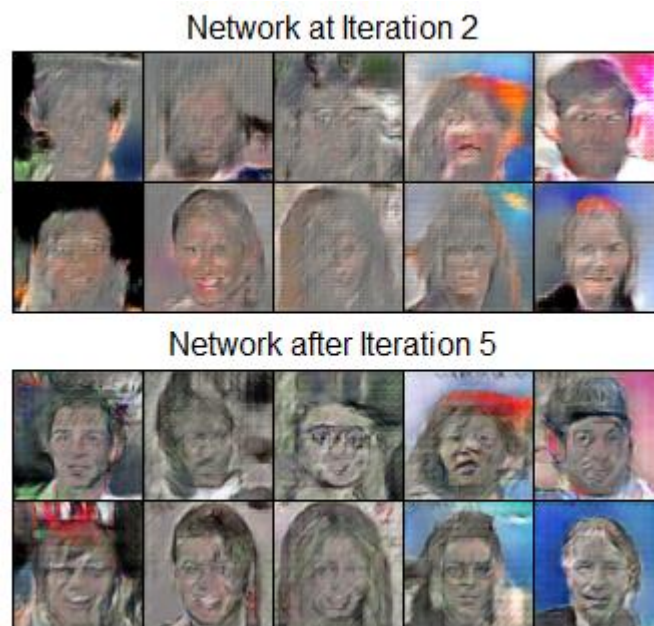


Figure 18: Results comparison, small improvement

Across the training iterations, some networks show improved results while some don't. Figure 17 shows the best results between iteration 2 and 5. Figure 18 and Figure 19 shows small to large improvements across the same iterations.

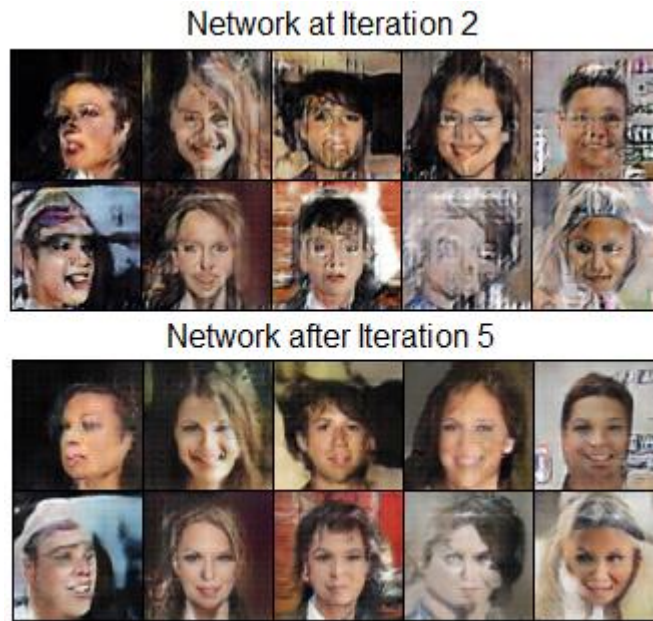


Figure 19: Results comparison, large improvement

5.3 Quantitative Results

5.3.1 Efficiency

Acer Nitro 5	
CPU	Intel i5 9300H
RAM	32GB DDR4-2666 Dual Channel
GPU	NVIDIA GeForce GTX 1650
OS	Windows 10 21H2
Python	3.10.2
Virtualenv	20.4.7
• All hardware was at default settings, no overclocking was performed • Device was only running network training, no other additional applications were running • Training was conducted from within a virtual environment	

Table 3: System Specifications

The network generated by the proposed framework was compared with a standard DCGAN. For consistency, the same hardware was used under similar conditions, shown in Table 3. As the proposed framework does not use

backpropagation for the entire epoch, the network is trained considerably faster than DCGAN. Repeated testing revealed that this method trains the network on average 3.75 times faster than DCGAN. Please consider Table 4 for details.

Operation	Mean Time (min)	Std. Deviation (min)
Selection (2AFC)	7	3
Crossover	0.01	0.005
Mutation	0.01	0.005
Correction	30	2

Table 4: Framework's Operating Time per Iteration

Using the system detailed in Table 3, DCGAN takes approximately 2.5 hrs. $\pm$ 10 min. Whereas the proposed framework takes approximately 37 min $\pm$ 5 min. This is a considerable improvement in training time taken. Moreover, the 2AFC process can be designed such as to use multiple operators instead of one. The framework can be further optimized to take inputs from the user while the network is being trained, so that the 2AFC time overlaps the Correction time.

As the 2AFC process replaces the standard fitness calculation and selection, several previously required computations for calculating comparisons metrics and sorting said metrics have been avoided.

5.3.2 Attempting Quality Comparison

Comparing different GANs with respect to the quality of the produced data can be a difficult process. Such is the case with the current task, when working with GANs producing images of human faces. The standard metrics such as SSIM, MS-SSIM, IS and FID cannot be used as they can be misleading.

As an example, these metrics often require a baseline for what is a good result, often labeled as the original image. Then the results of two different GANs are compared against the original image and a metric is computed to show which result is more alike the original.

The problem is, as in this case, there is no original image. The GANs are producing faces of humans that don't exist. Hence, there can be no original image, and hence these metrics cannot be used.

Some measures can be taken. For instance, we can take a fully trained DCGAN, and pass it a fixed noise vector to produce a certain image and call that the original. Then we can pass the same noise vector to the other GANs which we wish to compare, and attempt to get some inkling as to which GAN is better.

This method might give a decent result, but it would be inconsistent. Moreover, there is a fundamental flaw. Using a fully trained DCGAN for the original will result in a bias, as we are essentially comparing our proposed framework to DCGAN. If we do the inverse, and use a fully trained version of our framework

for the original, then the results would be different from before, and in the favor of the proposed framework. Please see Table 4 for details.

	Metric	Proposed Framwork	DCGAN
DCGAN as Original	SSIM	0.27952	0.37558
	MS-SSIM	0.59523	0.71943
Proposed Framework as Original	SSIM	0.47555	0.37896
	MS-SSIM	0.78836	0.70906

Table 5: Quantitative Comparison with Different 'Original' Sources

Another issue with using such metrics is connected with the notion of having potentially multiple ‘good’ results in our feature space. Essentially, when comparing GANs, a baseline of what is a bad result and what is a good result, is important to establish. Since there can be multiple different good results in the feature space, the ‘path’ each GAN takes in the feature space as it trains, might be different. Hence, for some iterations, a particularly bad result might be generated, whereas in some iterations, a particularly good result might be generated.

Due to this inconsistency, using such metrics would not be advisable for all tasks. Moreover, this shows why human observers are helpful to ascertain the generated samples quality.

5.4 Key findings

During testing, some key observations came into view.

5.4.1 Discriminator difficulty

If the discriminator is trained more than the generator, then the training results in a deterioration of the generator due to vanishing gradients. As our population consists of generator networks which have been all trained to different epochs, the testing becomes more difficult.

To prevent a catastrophic failure, the networks training is monitored. If the discriminators loss approaches zero while the generators loss continues to rise, the iterations training will be restarted with another discriminator whose training is lower (lower epoch reached leads to lower difficulty). If on the other hand, the generators loss approaches zero while the discriminators loss continues to rise, the new discriminator will have a higher level of training (higher epoch reached leads to higher difficulty).

5.4.2 Limited Trainings

During training, if the discriminator is not trained simultaneously with the generator, the generator results slowly deteriorate (depending on how many mini-batches the generator has trained without the discriminator).

As a consequence, both networks have to be trained repeatedly. Considering that the discriminator is not always going to be used in the next iterations, it is a waste.

6. Conclusion

6.1 Summary

In this thesis, a novel approach to improve the training of Generative Adversarial Networks (GANs) has been proposed. This method involves utilizing a Genetic Algorithm (GA) along with Two-Alternative Forced-Choice (2AFC).

The idea is to use the GA for a general exploratory search of the feature space of the GANs generator, and reducing the GAs computational and time complexity using 2AFC.

The results show that the method can successfully produce improvements within the generator in a relatively short amount of time.

This thesis appropriately showcases both the benefits of neuroevolution (using GAs with GANs) and 2AFC.

6.2 Future work

At the end of several experiments, certain ideas originate, which can be the foundations to future research.

2AFC can be used to completely remove the discriminator entirely. This would immensely reduce the processing time and hardware requirements of the task. However, doing so would require an appropriate function to calculate and backpropagate the generators loss. Moreover, this would increase the comparisons required from a human operator. This can be offset by enabling the system to take multiple simultaneously discretized inputs, from multiple human operators. Or as an alternative, split the comparisons required and disseminate them to multiple different systems, each with a different human operator.

An alternative system can also be considered, where instead of using several pre-trained networks as part of the population, only one network is used. The population is instead built up of different variations (mutations) of the original network. This would allow a different look into the feature space of the generator, and thus could provide interesting results.

7. References

- [1] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, p. 11–26, 2017.
- [2] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, p. 85–117, 2015.
- [3] R. Miiikkulainen, J. Liang, E. Meyerson, A. Rawal, D. Fink, O. Francon, B. Raju, H. Shahrzad, A. Navruzyan, N. Duffy and others, "Evolving deep neural networks," in *Artificial intelligence in the age of neural networks and brain computing*, Elsevier, 2019, p. 293–312.
- [4] Y.-X. Tang, Y.-B. Tang, M. Han, J. Xiao and R. M. Summers, "Abnormal chest X-ray identification with generative adversarial one-class classifier," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, 2019.
- [5] E. Colak, F. C. Kitamura, S. B. Hobbs, C. C. Wu, M. P. Lungren, L. M. Prevedello, J. Kalpathy-Cramer, R. L. Ball, G. Shih, A. Stein and others, "The RSNA Pulmonary Embolism CT Dataset," *Radiology: Artificial Intelligence*, vol. 3, p. e200254, 2021.
- [6] K. Ni, R. Pearce, K. Boakye, B. Van Essen, D. Borth, B. Chen and E. Wang, "Large-scale deep learning on the yfcc100m dataset," *arXiv preprint arXiv:1502.03409*, 2015.
- [7] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu and A. A. Efros, "Real-time user-guided image colorization with learned deep priors," *arXiv preprint arXiv:1705.02999*, 2017.
- [8] L. Bargsten and A. Schlaefer, "SpeckleGAN: a generative adversarial network with an adaptive speckle layer to augment limited training data for ultrasound image processing," *International journal of computer assisted radiology and surgery*, vol. 15, p. 1427–1436, 2020.
- [9] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen and T. Aila, "Training generative adversarial networks with limited data," *arXiv preprint arXiv:2006.06676*, 2020.
- [10] H.-C. Shin, N. A. Tenenholtz, J. K. Rogers, C. G. Schwarz, M. L. Senjem, J. L. Gunter, K. P. Andriole and M. Michalski, "Medical image synthesis for data augmentation and anonymization using generative adversarial networks," in *International workshop on simulation and synthesis in medical imaging*, 2018.
- [11] X. Zhu, Y. Liu, Z. Qin and J. Li, "Data augmentation in emotion classification using generative adversarial networks," *arXiv preprint arXiv:1711.00648*, 2017.
- [12] S. Shao, P. Wang and R. Yan, "Generative adversarial networks for data augmentation in machine fault diagnosis," *Computers in Industry*, vol. 106, p. 85–93, 2019.
- [13] Y. Qian, H. Hu and T. Tan, "Data augmentation using generative adversarial networks for robust speech recognition," *Speech Communication*, vol. 114, p. 1–9, 2019.
- [14] P. Soviany, C. Ardei, R. T. Ionescu and M. Leordeanu, "Image difficulty curriculum for generative adversarial networks (CuGAN)," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020.
- [15] H. Thanh-Tung and T. Tran, "Catastrophic forgetting and mode collapse in GANs," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020.
- [16] D. Bang and H. Shim, "Mggan: Solving mode collapse using manifold guided training," *arXiv preprint arXiv:1804.04391*, 2018.
- [17] S.-Y. Chen, W. Su, L. Gao, S. Xia and H. Fu, "DeepFaceDrawing: Deep generation of face images from sketches," *ACM Transactions on Graphics (TOG)*, vol. 39, p. 72–1, 2020.
- [18] T. Park, M.-Y. Liu, T.-C. Wang and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.
- [19] Z. P. Kilpatrick, W. R. Holmes, T. L. Eissa and K. Josić, "Optimal models of decision-making in dynamic environments," *Current opinion in neurobiology*, vol. 58, p. 54–60, 2019.
- [20] R. Bogacz, E. Brown, J. Moehlis, P. Holmes and J. D. Cohen, "The physics of optimal decision making: a formal analysis of models of performance in two-alternative forced-choice tasks," *Psychological review*, vol. 113, p. 700, 2006.
- [21] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and

- Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.
- [22] Y. Hong, U. Hwang, J. Yoo and S. Yoon, "How generative adversarial networks and their variants work: An overview," *ACM Computing Surveys (CSUR)*, vol. 52, p. 1–43, 2019.
- [23] P. Salehi, A. Chalechale and M. Taghizadeh, "Generative Adversarial Networks (GANs): An Overview of Theoretical Model, Evaluation Metrics, and Recent Developments," *arXiv preprint arXiv:2005.13178*, 2020.
- [24] L. Yin, V. Menkovski and M. Pechenizkiy, "Knowledge Elicitation using Deep Metric Learning and Psychometric Testing," *arXiv preprint arXiv:2004.06353*, 2020.
- [25] F. Fan, S. Ahn, B. De Man, K. A. Wangerin, S. D. Wollenweber, C. K. Abbey and P. E. Kinahan, "Deep learning-based model observers that replicate human observers for PET imaging," in *Medical Imaging 2020: Image Perception, Observer Performance, and Technology Assessment*, 2020.
- [26] H. Lei, "Strategies for Two Alternative Forced Choice Navigation Tasks," 2020.
- [27] E. D. Stalker, J. Y. Patel and I. Fruend, "Contrast Sensitivity in Naturalistic Images Measured Using Generative Adversarial Nets," *Journal of Vision*, vol. 19, p. 124c–124c, 2019.
- [28] D. Whitley, "A genetic algorithm tutorial," *Statistics and computing*, vol. 4, p. 65–85, 1994.
- [29] S. Katoch, S. S. Chauhan and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia Tools and Applications*, p. 1–36, 2020.
- [30] B. Roziere, F. Teytaud, V. Hosu, H. Lin, J. Rapin, M. Zameshina and O. Teytaud, "Evolgan: Evolutionary generative adversarial networks," in *Proceedings of the Asian Conference on Computer Vision*, 2020.
- [31] V. Costa, N. Lourenço and P. Machado, "Coevolution of generative adversarial networks," in *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*, 2019.
- [32] A. Borji, "Pros and cons of gan evaluation measures," *Computer Vision and Image Understanding*, vol. 179, p. 41–65, 2019.
- [33] M. Soloveitchik, T. Diskin, E. Morin and A. Wiesel, "Conditional Frechet Inception Distance," *arXiv preprint arXiv:2103.11521*, 2021.
- [34] Z. Wang, E. P. Simoncelli and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers*, 2003, 2003.
- [35] K. Deb, A. Pratap, S. Agarwal and T. A. M. T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE transactions on evolutionary computation*, vol. 6, p. 182–197, 2002.
- [36] Y. Yusoff, M. S. Ngadiman and A. M. Zain, "Overview of NSGA-II for optimizing machining process parameters," *Procedia Engineering*, vol. 15, p. 3978–3983, 2011.