



Honors College Theses

2021

Deepfakes Generated by Generative Adversarial Networks

Olympia A. Paul
Georgia Southern University

Follow this and additional works at: <https://digitalcommons.georgiasouthern.edu/honors-theses>



Part of the [Computer Engineering Commons](#), and the [Technology and Innovation Commons](#)

Recommended Citation

Paul, Olympia A., "Deepfakes Generated by Generative Adversarial Networks" (2021). *Honors College Theses*. 671.

<https://digitalcommons.georgiasouthern.edu/honors-theses/671>

This thesis (open access) is brought to you for free and open access by Digital Commons@Georgia Southern. It has been accepted for inclusion in Honors College Theses by an authorized administrator of Digital Commons@Georgia Southern. For more information, please contact digitalcommons@georgiasouthern.edu.

Deepfakes Generated by Generative Adversarial Networks

By

Olympia Paul

Under the mentorship of Dr. Hayden Wimmer

ABSTRACT

Deep learning is a type of Artificial Intelligence (AI) that mimics the workings of the human brain in processing data such as speech recognition, visual object recognition, object detection, language translation, and making decisions. A Generative adversarial network (GAN) is a special type of deep learning, designed by Goodfellow et al. (2014), which is what we call convolution neural networks (CNN). How a GAN works is that when given a training set, they can generate new data with the same information as the training set, and this is often what we refer to as deep fakes. CNN takes an input image, assigns learnable weights and biases to various aspects of the object and is able to differentiate one from the other. This is similar to what GAN does, it creates two neural networks called discriminator and generator, and they work together to differentiate the sample input from the generated input (deep fakes). Deep fakes is a machine learning technique where a person in an existing image or video is replaced by someone else's likeness. Deep fakes have become a problem in society because it allows anyone's image to be co-opted and calls into question our ability to trust what we see. In this project we develop a GAN to generate deepfakes. Next, we develop a survey to determine if participants are able to identify authentic versus deep fake images. The survey employed a questionnaire asking participants their perception on AI technology based on their overall familiarity of AI, deep fake generation, reliability and trustworthiness of AI, as well as testing to see if subjects can distinguish real versus deep fake images. Results show demographic differences in perceptions of AI and that humans are good at distinguishing real images from deep fakes.

Thesis Mentor: _____

Dr. Hayden Wimmer

Honors Director: _____

Dr. Steven Engel

December 2021

Information Technology

Honors College

Georgia Southern University

Acknowledgments

Without the help of many people, this thesis would not be completed. I would like to give special thanks to my brother for encouraging me to join the honors college that provided me this wonderful research experience. I would also like to dedicate this piece to my parents, I am grateful for all their support throughout this process. I would like to acknowledge all the time and efforts provided by the professor who is also my mentor – Dr. Hayden Wimmer. I am very thankful for all the help he provided me in completing my research, I couldn't have done it without him.

Introduction

Artificial intelligence (AI) is a wide-ranging branch of computer science concerned with building smart machines capable of performing tasks that typically require human intelligence. AI requires a foundation of specialized hardware and software for writing and training machine learning algorithms. No one programming language is synonymous with AI, but a few, including Python, R and Java, are popular. In general, AI systems work by ingesting large amounts of labeled training data, analyzing the data for correlations and patterns, and using these patterns to make predictions about future states. Examples of AI include Siri, chatbots, Alexa, smart assistants, self-driving cars, TV show recommendations etc.

Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns, and make decisions with minimal human intervention. While AI is the broad science of mimicking human abilities, machine learning is a specific subset of AI that trains a machine how to learn. Because of new computing technologies, machine learning today is not like machine learning of the past. It was born from pattern recognition and the theory that computers can learn without being programmed to perform specific tasks; researchers interested in artificial intelligence wanted to see if computers could learn from data. The iterative aspect of machine learning is important because as models are exposed to new data, they are able to independently adapt. They learn from previous computations to produce reliable, repeatable decisions and results. It's a science that's not new – but one that has gained fresh momentum.

Deep learning is a subset of machine learning where artificial neural networks, algorithms inspired by the human brain, learn from large amounts of data. Similarly, to how we learn from experience, the deep learning algorithm would perform a task repeatedly, each time tweaking it a little to improve the outcome. We refer to ‘deep learning’ because the neural networks have various (deep) layers that enable learning. Just about any problem that requires “thought” to figure out is a problem deep learning can learn to solve. The amount of data we generate every day is staggering—currently estimated at 2.6 quintillion bytes—and it’s the resource that makes deep learning possible. Since deep-learning algorithms require a ton of data to learn from, this increase in data creation is one reason that deep learning capabilities have grown in recent years. In addition to more data creation, deep learning algorithms benefit from the stronger computing power that’s available today as well as the proliferation of AI as a Service. AI as a Service has given smaller organizations access to artificial intelligence technology and specifically the AI algorithms required for deep learning without a large initial investment.

The objective of this research is to determine if humans can recognize deepfake technology by using a deep learning method called Generative Adversarial Networks to create deepfakes. Deepfakes are fake images created using machine learning techniques making it harder for humans to distinguish between authentic and non-authentic images. We use python language to program our GAN and create fake images of the public CelebA dataset. Then we create a survey for people to rate the authenticity of 10 different images. Using the data we receive from our survey we perform t-testing to test our hypothesis on people’s ability to tell deepfakes from authentic images. Results show our

data analysis using null and research hypotheses. The purpose of this research is to educate people on the dangers of deepfakes; how it can trick the society into believing false news and destroy the reputation of public figures. People need to pay close attention to how AI is rapidly progressing harmful technology such as deepfakes and shouldn't be too quick to believe what they see in the media.

Literature Review

Goodfellow, et al. [1] proposed a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G . In the proposed adversarial nets framework, the generative model is pitted against an adversary: a discriminative model that learns to determine whether a sample is from the model distribution or the data distribution. The generative model can be thought of as analogous to a team of counterfeiters, trying to produce fake currency and use it without detection, while the discriminative model is analogous to the police, trying to detect the counterfeit currency. Competition in this game drives both teams to improve their methods until the counterfeits are indistinguishable from the genuine articles. They trained the adversarial nets on an array of datasets including MNIST, Toronto Face Database (TFD) and CIFAR-10. The new framework comes with advantages and disadvantages relative to previous modeling frameworks. The disadvantages are primarily that there is no explicit representation of the data z , and that D must be synchronized well with G during training (in particular, G must not be trained too much without updating D , in order to avoid "the Helvetica scenario" in which G collapses too many values of variables to the same value

of x to have enough diversity to model. An advantage of adversarial networks is that they can represent very sharp, even degenerate distributions. This framework admits that a conditional generative model can be obtained by adding c as input to both G and D . Also, learned approximate inference can be performed by training an auxiliary network to predict z given x [1].

Radford, et al. [2] mentions how supervised learning with convolutional networks (CNNs) are more popular than unsupervised learning with CNNs which receive less attention. However, they hope to help bridge the gap by introducing a class of CNNs called Deep Convolutional Neural Networks (DCGANs). In this paper [2] evaluates a set of constraints on DCGAN that make them more stable to train, they use the trained discriminator for image classification tasks, showing competitive performance with other unsupervised algorithms and empirically visualize it to show the filters. Previous work has demonstrated that supervised training of CNNs on large image datasets results in very powerful learning. Additionally, supervised CNNs trained on scene classification learn object detectors. But according to results, authors demonstrate that an unsupervised DCGAN trained on a large image dataset can also learn a hierarchy of features that are interesting. Using guided backpropagation [2].

Liu and Tuzel [3] described Coupled generative adversarial networks (CoGAN) as an extension of GAN for learning joint distribution of multi-domain images. CoGAN is designed for joint image distribution tasks in two different domains. “It consists of a pair of GANs - GAN1 and GAN2; each responsible for synthesizing images in one domain.” During training, they force them to share a subset of parameters, which results in the GANs learning to synthesize pairs of corresponding images without

correspondence supervision. In the experiments, they applied CoGAN on MNIST digits, image faces, Color and Depth images - RGBD dataset and NYU dataset. The learning results for the MNIST showed that without training the CoGAN for corresponding images they learned to render corresponding ones for the images [3].

Kwak and Zhang [4] mentions that image generation remains a fundamental problem of artificial intelligence with so many different proposed models to improve it. In this paper [4] proposes a model called composite generative adversarial network (CGAN), that disentangles complicated factors of images with multiple generators in which each generator generates some part of the image. CGAN is an extension of GAN that consists of multiple generators connected with a recurrent neural network (RNN). The generators in CGANs are different from that of GANs as there are additional alpha channels in the output. The images are then combined sequentially with alpha blending to form a final image. CGANS assigns roles for each generator by factoring the common factors of images and creating realistic samples. CGANs are being trained on CelebA, and Oxford 102 Flowers datasets and pororo cartoon video. All images are resized to 64 x 64 with antialiasing. Using three generators CGANS successfully generated image's part by part, it generated backgrounds, faces, and hair parts, respectively to end up with the final images. The third generator at first failed to generate meaningful images but after applying alpha loss the problem diminishes, so the alpha loss makes the images less blurry [4].

Hitawala [5] Since the discovery of GAN by Goodfellow et al. (2014) many modifications have been proposed. This study shows a comparative analysis of these models over the original model. Some of the modifications reviewed in this project are

CGAN, Laplacian Pyramid of Adversarial Networks (LAPGAN), DCGAN, Generative Recurrent Adversarial Networks (GRAN), AAE, InfoGAN and BiGAN. The initial versions of GANs such as Vanilla GAN and CGAN only supported supervised learning which were later augmented to support semi-supervised and unsupervised learning. The primary objective of any adversarial network remains a 2-player minimax game over all versions. Additionally, some models had secondary objectives such as feature learning and learning of representations through related semantic tasks and then later using these learned features for classification or recognition in unsupervised settings. Also, models such as LAPGAN and GRAN introduced a sequential generation of images by the generator using Laplacian pyramids and recurrent networks. Additionally, earlier models evaluated model performance based on log-likelihood estimates which was discarded in later versions as it was not a good estimate. Instead, accuracy and error rates were used for evaluating the performance of a model. Also, GRAN proposed a new evaluation metric called Generative Adversarial Metric for evaluating the performance of generative adversarial nets Comparative Study on Generative Adversarial Networks although it has not been in use by any other generative model. Conclusively, the later versions of adversarial networks are more robust and have many more applications compared to the original version. Also, these networks can prove to be useful in image classification, recognition, capturing and generation in a variety of ways [5].

Zhao, et al. [6] learning from synthetic faces may not achieve the desired performance due to discrepancy between distributions of the synthetic and real face images. To narrow this gap, they propose a DA-GAN model which can improve the realism of face simulators' output using unlabeled real faces, while preserving the identity

information during the realism refinement. The proposed DA-GAN effectively combines prior knowledge from data distribution and domain knowledge of faces to exactly recover the lost information inherent in projecting a 3D face into the 2D image space. Using the benchmark dataset DA-GAN presents a good photo realistic quality. This method won 1st place on verification and identification tracks in NIST IJB-A 2017 face recognition competitions and authors encourage the application of DA-GAN for other transfer learning applications in the future [6].

Yang, et al. [7] discusses Age progression which is the process of aesthetically rendering a given face image to present the effects of aging. It is often used in the entertainment industry and forensics. However, the intrinsic complexity of physical aging, the interferences caused by other factors and shortage of labeled aging data collectively make face age progression a rather difficult problem. The last few years have witnessed significant efforts tackling this issue, where aging accuracy and identity permanence are commonly regarded as the two-underlying premises of its success. In this work a novel based GAN method is proposed. This method involves the techniques on face verification and age estimation and exploits a compound training critic that integrates the simple pixel-level penalty, the age-related GAN loss achieving age transformation, and the individual-dependent critic keeping the identity information stable. For generating detailed signs of aging, a pyramidal discriminator is designed to estimate high-level face representations in a finer way. Extensive experiments are conducted, and both the achieved aging images and the quantitative evaluations clearly confirm the effectiveness and robustness of the proposed method [7].

Song, et al. [8] Proposes dual conditional GANs (Dual cGANs) for face aging progression and regression when using training sets of unlabeled face images. Face aging and rejuvenation is predicting how a person would look at different ages. While prior work has made great progress in this topic, there are two major problems that have remained largely unsolved which are: the majority of prior work requires sequential training data, which is very rare in real scenarios, and how to simultaneously render aging face and preserve personality. So, to tackle these problems Dual cGANs are proposed. Dual cGANs consist of the primal cGAN and the dual cGAN. Each of them consists of three components: target generator, source generator and their discriminators. The target generator generates the face of a person at different ages based on the input image and target age, the input and output are both colorful face images with shape $256 \times 256 \times 3$. The source generator reconstructs the input face of a person based on the synthesized image and source age, while the discriminator aims to distinguish between the generated image and its ground truth. Authors experiment on the UTKFace dataset which has a long age span ranging from 0 to 116 years of age and the results demonstrate that the generated images are photo-realistic, the details in the skin, muscles and wrinkles are very clear, the process shows the colors change from light to dark and the skin from smooth to wrinkles. Lastly, the generated images in each age group have specific subtle features. For example, the child tends to have a round face and no teeth, while the elderly people usually have small eyes and gray hair. These results indicate that our Dual cGANs achieve promising results for face aging and rejuvenation [8].

Zhao, et al. [9] introduces unconstrained face recognition as a very important and challenging problem. Labelling huge amounts of data for feeding supervised deep

learning algorithms is expensive and time-consuming. The pose distribution of available face recognition datasets is usually unbalanced, showing a long tail with large pose variations, so this has become a main obstacle for further pushing unconstrained face recognition performance. Several researches have been made to employ synthetic profile face images to balance pose variations, but this may not always achieve desired performance due to the discrepancy between synthetic and real face images. Therefore, this work proposes a novel Dual-Agent Generative adversarial network for photorealistic and identity preserving profile face synthesis even under extreme poses. DA-GAN leverages a fully convolutional network as the generator to generate high-resolution images and an auto-encoder as the discriminator with the dual agents. Besides the novel architecture, we make several key modifications to the standard GAN to preserve pose, texture as well as identity, and stabilize the training process: (i) a pose perception loss; (ii) an identity perception loss; (iii) an adversarial loss with a boundary equilibrium regularization term. Results show that DA-GAN not only achieves outstanding perceptual results but also significantly outperforms state-of-the-arts on the large-scale and challenging NIST IJB-A and CFP unconstrained face recognition benchmarks. Conclusively, DA-GAN is also a promising new approach for solving generic transfer learning problems more effectively [9].

Yi, et al. [10] introduces image stylization using deep learning. Training a computer program with artists' drawings and automatically transforming an input photo into high quality artistic drawings is much desired. With the development of deep learning, neural style transfer (NST), which uses CNNs to perform image style transfer was proposed in previous work, but later GAN based style transfer methods have

achieved good results. Artistic portrait drawings (APDrawings) are substantially different in style from portrait painting styles studied in previous work and existing methods fail to produce high quality artistic portrait drawings. To resolve this issue a hierarchical GAN architecture for APDrawing synthesis is introduced (APDrawingGAN). They can generate high quality and expressive artistic portrait drawings. In particular, the method can learn complex hair styles with delicate white lines. Artists use multiple graphical elements when creating a drawing. In order to best emulate artists, the authors model separates the GAN's rendered output into multiple layers, each of which is controlled by separated loss functions. They also propose a loss function dedicated to APDrawing with four loss terms in their architecture, including a novel DT loss (to promote line-stroke based style in APDrawings) and a local transfer loss (for local networks to preserve facial features). [10] pretrains the model using 6,655 frontal face photos collected from ten face datasets and constructs an APDrawing dataset (containing 140 high-resolution face photos and corresponding portrait drawings by a professional artist) suitable for training and testing. Experimental results and a user study show that our method can achieve successful artistic portrait style transfer and outperforms state-of-the-art methods. Results are still not as clean in hair and lip regions, but this is planned to be addressed in future work [10].

Gu, et al. [11] Portrait editing is a popular subject in photo manipulation. GAN advances the generating of realistic faces and allows more face editing. In this paper, authors argue about three issues in existing techniques: diversity, quality, and controllability for portrait synthesis and editing. To address these issues, they propose a novel end-to-end learning framework that leverages conditional GANs guided by

provided face masks for generating faces. The framework learns feature embeddings for every face component (e.g., mouth, hair, eye), separately, contributing to better correspondences for image translation, and local face editing. With the mask, our network is available to many applications, like face synthesis driven by mask, face Swap (including hair in swapping), and local manipulation. It can also boost the performance of face parsing a bit as an option of data augmentation. [11] uses the Helen Dataset to validate the effectiveness of the proposed method and our results show that using local embedding sub-network helps the generated results to keep the details (e.g., eye's size, skin color, hair color) from the source images and in conclusion generating more realistic faces [11].

Zhang, et al. [12] discusses how to improve the quality of generated face images with generative adversarial networks by replacing MLP with Convolutional neural networks (CNN) and removing pooling layers. For face image generation the original GAN network is not stable in image quality generated by generators during training and cannot get high-quality generators. The main reason for this problem is that generators and discriminators use the same back-propagation network. To address this problem this paper proposes methods to modify the original GAN architecture. They use DCGAN architecture to train the model. First, the full convolutional network uses stride convolution instead of the deterministic space pooling function. They use the method of network learning's own spatial down-sampling to be applied in the generating network, allowing it to learn its own spatial up-sampling in the discriminating network. Then, they remove the fully connected layer. The experiments are performed on the LFW and CelebA face dataset and show the effectiveness of their method [12].

Karras, et al. [13] The resolution and quality of images produced by generative adversarial networks is improving rapidly. The current state-of-art method for high resolution images is StyleGAN, which has been shown to work on a variety of datasets. This work focuses on fixing its characteristic artifacts and improving the result quality further. Many observers have noticed characteristic artifacts in images generated by StyleGAN. The authors identify these artifacts and describe changes in architecture and training methods that eliminate them. First, they investigate the origin of common blob-like artifacts and find that the generator creates them to circumvent a design flaw in its architecture. They redesign the normalization used in the generator, which removes the artifacts. Secondly, they analyze artifacts related to progressive growth that have been highly successful in stabilizing high-resolution GAN training. They propose an alternative design that achieves the same goal — training starts by focusing on low-resolution images and then progressively shifts focus to higher and higher resolutions — without changing the network topology during training. This new design also allows reasoning about the effective resolution of the generated images, which turns out to be lower than expected, motivating a capacity increase. Authors also use Fréchet inception distance (FID) which measures differences in the density of two distributions in high dimensional feature space of an inception V3 classifier to quantify the image improvements of the stylegan. Results show that they identified and fixed several image quality issues in StyleGAN, improving the quality further and considerably advancing the state of the art in several datasets. In some cases, the improvements are more clearly seen in motion, as demonstrated in the accompanying video. Despite the improved quality, StyleGAN2 makes it easier to attribute a generated image to its source. They find that the

projection of images to the latent space works significantly better with the new path length regularized StyleGAN2 [13].

Bayat, et al. [14] GANs synthesize realistic images from a random latent vector. While many studies have explored various training configurations and architectures for GANs, the problem of inverting a generative model to extract latent vectors of given input images has been inadequately investigated. Although there is exactly one generated image per given random vector, the mapping from an image to its recovered latent vector can have more than one solution. In this work, [14] trains a residual neural network (ResNet18) in order to map an input image to its corresponding latent vector using a combination of a reconstruction loss and a perceptual loss. Mean Absolute Error (MAE) is used as the reconstruction loss. Authors introduce two frameworks: the first architecture trains the network on generated faces for which we have the ground truth latent vectors. The second architecture deals with natural human faces using a pixel loss and a perceptual loss between the reconstructed face and the target as well as the z-loss. The results show that adding perceptual loss improves visual quality and results in faces indistinguishable from the target. Therefore, their latent vector reconstructs better face features and also performs better in identification tasks [14].

Tang [15] GANs have made great progress in synthesizing realistic images in recent years. However, they are often trained on image datasets with either too few samples or too many classes belonging to different data distributions. Because of these they are prone to overfitting and underfitting, mode collapse and performance degrading. To cope with these challenges the author trains variants of GAN on artificial datasets (mixtures of Gaussians in high dimensional space) that have many samples and simple

real data distributions. The author uses Wasserstein GAN (WGAN) and MIX+GANs on the 3-Gaussians dataset and results show that increasing the size of the training set can improve the performance of GANs, even when the training set is already large. Also, training a mixture of GANs is more beneficial than simply increasing the complexity of standalone networks for modeling multi-modal data. Moreover, the results show that current datasets might not be large enough to make GANs learn the real data distribution [15].

In this paper Tripathy, et al. [16] proposed a generic face animator that is able to control the pose and expression of a given face image. The animation was controlled using human interpretable attributes consisting of head pose angles and action unit activations. The selected attributes enabled selective manual editing as well as mixing the control signal from several different sources (e.g., multiple driving frames). One of the key ideas in our approach was to transform the source face into a canonical presentation that acts as a template for the subsequent animation steps. Our model was demonstrated in numerous face animation tasks including face reenactment, selective expression manipulation, 3D face rotation, and face frontalization. In the experiments, the proposed ICface model was able to produce high quality results for a variety of different source and driving identities. The future work includes further increasing the resolution of the output images and further improving the performance with extreme poses having a few training samples [16].

Zhang and Zhao [17] explain face image generation based on GAN is a hot research topic in computer vision. Existing GAN-based algorithms are constrained by training instability and mode collapse, but in order to further explore methods to improve

the stability and quality of image generation, this paper constructs a training method of GAN based on particle swarm optimization algorithm. The particle swarm optimization (PSO) is utilized to optimize the parameters of the generator network, where two indicators of generating quality and generating diversity are constructed to evaluate the performance of the generator. The optimal solution of the population and the optimal solution of a single particle are found in the population, and the iterative training is carried out. The experiment is done on the CelebA dataset, they use PSO to optimize the parameters of the generator network and improve the inertia weight of PSO [17].

Background

Machine learning is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence. They build models based on samples and make predictions or decisions about that data without being programmed to do so. They are used to identify objects in images, transcribe speech into text, match news items, posts, or services on user's interests, and select relevant results of a search. And these all make use of techniques in deep learning. Deep learning is an AI function that mimics the workings of the human brain in processing data such as speech recognition, visual object recognition, object detection, language translation, and making decisions. Deep learning AI can function without human supervision, drawing from data that is both unstructured and unlabeled.

A GAN is a special type of deep learning which is what we call convolution neural networks (CNN). GAN is a class of machine learning designed by Goodfellow et al. (2014). How GAN works is that when given a training set, they can generate new data

with the same information as the training set. For instance, a GAN prepared on photos can produce new photos that take a gander in any event hastily true to human observers. This new generated data is what is often referred to as Deepfakes. CNN takes an input image, assigns learnable weights and biases to various aspects of the object and is able to differentiate one from the other. This is similar to what GAN does, it creates two neural networks called discriminator and generator, and they work together to differentiate the sample input from the generated input (deepfakes).

These Deepfakes are images generated by deep learning AI to create synthetic media in which a person in an existing image or video is swapped with another person's likeness. This has become a societal challenge as they are difficult to impossible to distinguish from an authentic image. They have been negatively used to trick the society by creating fake news and misleading pictures and videos. Because machines are generating perfect images these days, it has become difficult to distinguish the machine-generated images from the originals. Examples of deepfakes that exist are the video of Barack Obama cursing out Donald Trump, Mark Zuckerberg bragging about having control of billions of people's stolen data and Jon Snow's moving apology for the dismal ending to Game of Thrones. Also, many deepfakes are pornographic images of celebrities. The AI firm Deeprtrace found 15,000 deepfake videos online in September 2019, a near doubling over nine months. A staggering 96% were pornographic and 99% of those mapped faces from female celebrities onto porn stars. As new techniques allow unskilled people to make deepfakes with a handful of photos, fake videos are likely to spread beyond the celebrity world to fuel revenge porn. As Danielle Citron, a professor of law at Boston University, puts it: "Deepfake technology is being weaponized against

women.” Beyond the porn there’s plenty of spoof, satire and mischief. Deepfake technology can create convincing but entirely fictional photos from scratch. A non-existent Bloomberg journalist, “Maisy Kinsley”, who had a profile on LinkedIn and Twitter, was probably a deepfake. Another LinkedIn fake, “Katie Jones”, claimed to work at the Center for Strategic and International Studies, but is thought to be a deepfake created for a foreign spying operation. Audio can be deepfaked too, to create “voice skins” or “voice clones” of public figures. Last March, the chief of a UK subsidiary of a German energy firm paid nearly £200,000 into a Hungarian bank account after being phoned by a fraudster who mimicked the German CEO’s voice. The company’s insurers believe the voice was a deepfake, but the evidence is unclear. Similar scams have reportedly used recorded WhatsApp voice messages. University researchers and special effects studios have long pushed the boundaries of what’s possible with video and image manipulation. But deepfakes themselves were born in 2017 when a Reddit user of the same name posted doctored porn clips on the site. The videos swapped the faces of celebrities – Gal Gadot, Taylor Swift, Scarlett Johansson and others – on to porn performers. It takes a few steps to make a face-swap video. First, you run thousands of face shots of the two people through an AI algorithm called an encoder. The encoder finds and learns similarities between the two faces, and reduces them to their shared common features, compressing the images in the process. A second AI algorithm called a decoder is then taught to recover the faces from the compressed images. Because the faces are different, you train one decoder to recover the first person’s face, and another decoder to recover the second person’s face. To perform the face swap, you simply feed encoded images into the “wrong” decoder. For example, a compressed image of person

A's face is fed into the decoder trained on person B. The decoder then reconstructs the face of person B with the expressions and orientation of face A. For a convincing video, this has to be done on every frame. It is hard to make a good deepfake on a standard computer. Most are created on high-end desktops with powerful graphics cards or better still with computing power in the cloud. This reduces the processing time from days and weeks to hours. But it takes expertise, too, not least to touch up completed videos to reduce flicker and other visual defects. That said, plenty of tools are now available to help people make deepfakes. Several companies will make them for you and do all the processing in the cloud. There's even a mobile phone app, Zao, that lets users add their faces to a list of TV and movie characters on which the system has trained. Deepfakes are not illegal per se, but producers and distributors can easily fall foul of the law. Depending on the content, a deepfake may infringe copyright, breach data protection law, and be defamatory if it exposes the victim to ridicule. There is also the specific criminal offence of sharing sexual and private images without consent, i.e. revenge porn, for which offenders can receive up to two years in jail. In Britain the law is split on this. In Scotland, revenge porn law includes deepfakes by making it an offence to disclose, or threaten to disclose, a photo or film which shows or appears to show another person in an intimate situation. But in England, the statute carefully excludes images that have been created solely by altering an existing image.

For this project the technology we will be using to create deepfakes is GAN. GAN is the popular method that is being used to create deepfakes from a given dataset. GANs consist of two networks, a Generator $G(x)$ the encoder, and a Discriminator $D(x)$ the decoder. They both play an adversarial game where the generator tries to fool the

Convolutional GANs (DCGANs), Conditional GANs (cGANs), StackGAN, InfoGANs, Wasserstein GANs (WGAN) and Disco GANs.

a. Deep Convolutional GANs (DCGANs):

DCGANs are an improvement of GANs. They are more stable and generate higher quality images. In DCGAN, batch normalization is done in both networks, i.e the generator network and the discriminator network. They can be used for style transfer. For example, you can use a dataset of handbags to generate shoes in the same style as the handbags.

b. Conditional GANs (cGANs):

These GANs use extra label information and result in better quality images and are able to control how generated images will look. cGANs learn to produce better images by exploiting the information fed to the model.

c. StackGAN:

Using a StackGAN, one can generate images from a text description, and they also perform image to image translation by producing a real image of an object using sketches. For example, a StackGAN can generate an image of a flying bird from a sentence describing the image and action.

d. InfoGANs:

InfoGAN is an information-theoretic extension to the GAN that is able to learn disentangled representations in an unsupervised manner. InfoGANs are used when your dataset is very complex, when you would like to train a cGAN and the

dataset is not labelled, and when you'd like to see the most important features of your images.

e. Wasserstein GANs (WGAN):

WGANs change the loss function to include a Wasserstein distance. They have loss functions that correlate to image quality.

f. Discover Cross-Domain Relations with Generative Adversarial Networks (Disco GANs):

Disco GANs are basically used for style transfer by using the network transfer style from one domain to another.

History of GAN

In the late 90s and early 2000s face detection was a major area of research because of its possible implications for military and security use. Almost twenty years later, this problem is basically solved, and face detection technology is available freely as open-source libraries in most programming languages. Python's most popular face detection library may be OpenCV or face-recognition. From here various apps have evolved to be able to swap faces of two people in images. Friends have used these apps to see how they would look with the other friend's body and even switched faces with celebrities and politicians. An example of an app that does this is called FaceApp.

GAN takes a different approach to learning than other types of neural networks. GANs algorithmic architectures that use two neural networks called a Generator and a Discriminator, which "compete" against one another to create the desired result. The

Generator's job is to create realistic-looking fake images, while the Discriminator's job is to distinguish between real images and fake images. If both are functioning at high levels, the result is images that are seemingly identical real-life photos. Generative Adversarial Networks have had a huge success since they were introduced in 2014 by Ian J.

Goodfellow. They were developed in the first place because it has been noticed most of the mainstream neural nets can be easily fooled into misclassifying things by adding only a small amount of noise into the original data. Surprisingly, the model after adding noise has higher confidence in the wrong prediction than when it predicted correctly. The reason for such an adversary is that most machine learning models learn from a limited amount of data, which is a huge drawback, as it is prone to overfitting. Also, the mapping between the input and the output is almost linear. Although it may seem that the boundaries of separation between the various classes are linear, in reality, they are composed of linearities and even a small change in a point in the feature space might lead to misclassification of data.

GAN Implementation

Our objective is to create the GAN model capable of generating realistic human images that do not exist in reality. We used the Deep Convolutional Adversarial Network (DCGAN) to generate deepfakes of the public CelebA dataset. DCGAN is the most popular network design for GAN, it is one of the models that demonstrated how to build a practical GAN that can learn by itself how to synthesize new images. DCGAN is very similar to GANs but specifically focuses on using deep convolutional networks in place of fully connected networks used in Vanilla GANs. Convolutional networks help in finding deep correlation within an image, that is they look for spatial correlation. This

means DCGAN would be a better option for image/video data, whereas GANs can be considered as a general idea on which DCGAN and many other architectures (CGAN, CycleGAN, StarGAN and many others) have been developed.

CelebFaces Attributes Dataset (CelebA) is a large-scale face attributes dataset with more than 200K celebrity images, each with 40 attribute annotations. The images in this dataset cover large pose variations and background clutter. CelebA has large diversities, large quantities, and rich annotations, including - 10,177 number of identities, - 202,599 number of face images, and - 5 landmark locations, 40 binary attributes annotations per image. The dataset can be employed as the training and test sets for the following computer vision tasks: face attribute recognition, face detection, and landmark (or facial part) localization.

Using python, we coded our GAN in Google colab which is a free open source application provided by Google specifically for deep learning tasks. It runs completely in the cloud, enables you to share your work, save your files directly to your google drive and offers resources for compute power which is what we specifically needed for our GAN. Training a GAN needs a lot of computing power because it takes a long time to train and Colab offers GPU and TPU runtime for accelerated training. A code that would run for days can run for 10 minutes using Colab.

To begin implementing our GAN I mounted our drive to upload files from it to Colab using `“from google.colab import drive drive.mount("/content/drive")”`. After that we downloaded our CelebA dataset from Kaggle and unzipped it on colab into our drive. Secondly, we imported the necessary libraries we needed to build our GAN such as

numpy, pandas, image and pyplot to train our dataset. Then, we loaded our dataset to see how our input images look like:

```
images = []
for pic_file in tqdm(os.listdir(PIC_DIR)[:IMAGES_COUNT]):
    pic = Image.open(PIC_DIR + pic_file).crop(crop_rect)
    pic.thumbnail((WIDTH, HEIGHT), Image.ANTIALIAS)
    images.append(np.uint8(pic))
```

```
100%|██████████| 10000/10000 [33:26<00:00, 4.98it/s]
```

```
images = np.array(images) / 255
print(images.shape)
```

```
from matplotlib import pyplot as plt
```

```
(10000, 128, 128, 3)
```

```
plt.figure(1, figsize=(10, 10))
for i in range(25):
    plt.subplot(5, 5, i+1)
    plt.imshow(images[i])
    plt.axis('off')
plt.show()
```

Figure 2: Loading our dataset

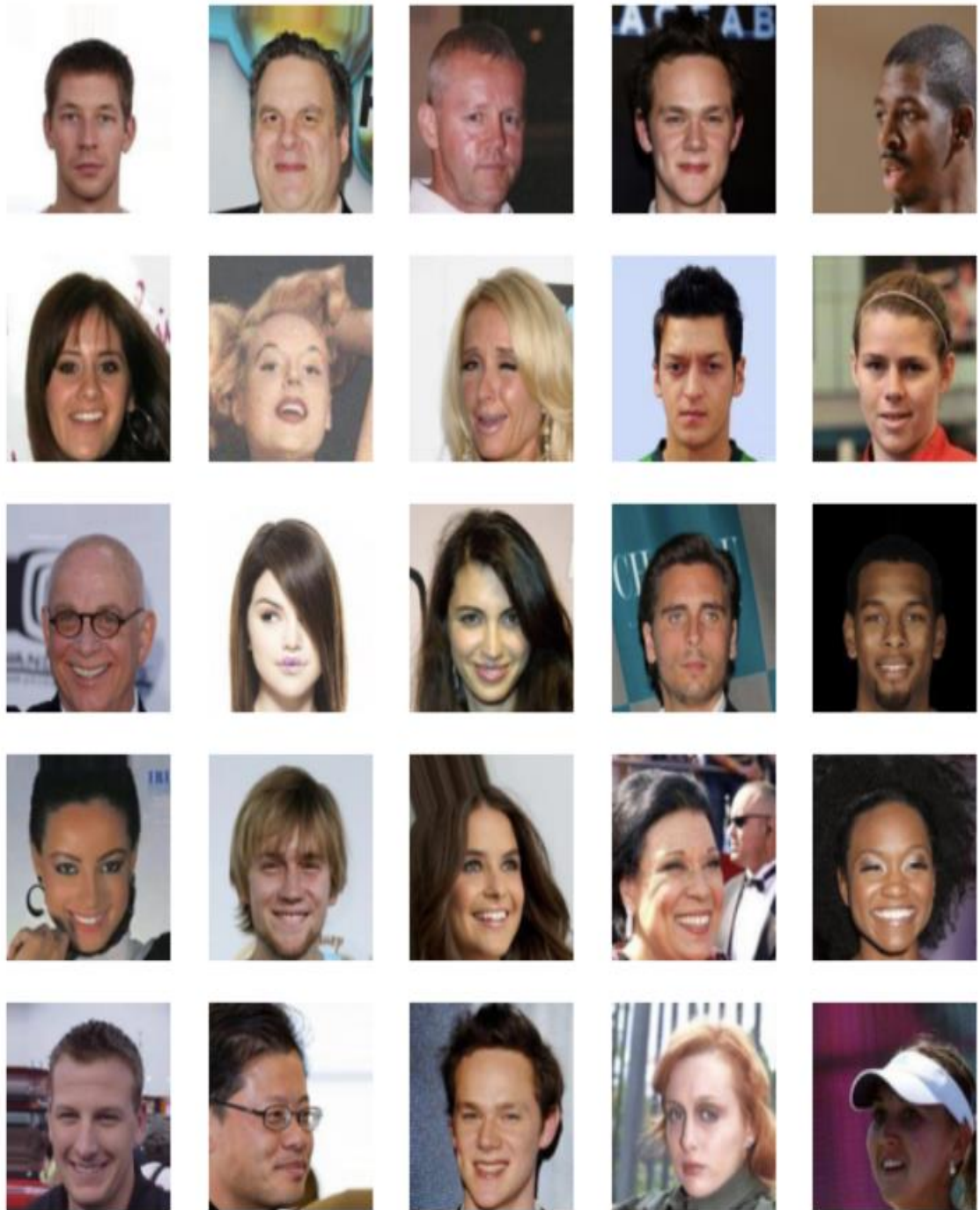


Figure 3: Sample CelebA images

Thirdly, we created our generator and discriminator. The generator network consists of 8 convolutional layers. Each convolutional layer performs a convolution and

then performs batch normalization and a leaky ReLU as well. Then, we return the tanh activation function:

```
[ ] # Create generator model
def create_generator():
    gen_input = Input(shape=(LATENT_DIM, ))

    x = Dense(128 * 16 * 16)(gen_input)
    x = LeakyReLU()(x)
    x = Reshape((16, 16, 128))(x)

    x = Conv2D(256, 5, padding='same')(x)
    x = LeakyReLU()(x)

    x = Conv2DTranspose(256, 4, strides=2, padding='same')(x)
    x = LeakyReLU()(x)

    x = Conv2DTranspose(256, 4, strides=2, padding='same')(x)
    x = LeakyReLU()(x)

    x = Conv2DTranspose(256, 4, strides=2, padding='same')(x)
    x = LeakyReLU()(x)

    x = Conv2D(512, 5, padding='same')(x)
    x = LeakyReLU()(x)
    x = Conv2D(512, 5, padding='same')(x)
    x = LeakyReLU()(x)
    x = Conv2D(CHANNELS, 7, activation='tanh', padding='same')(x)
```

Figure 4: Creating our generator model

The discriminator network consists of convolutional layers the same as the generator. For every layer of the network, we perform a convolution, then we perform

batch normalization to make the network faster and more accurate and finally, we perform a Leaky ReLu:

```
[ ] # Create discriminator
def create_discriminator():
    disc_input = Input(shape=(HEIGHT, WIDTH, CHANNELS))

    x = Conv2D(256, 3)(disc_input)
    x = LeakyReLU()(x)

    x = Conv2D(256, 4, strides=2)(x)
    x = LeakyReLU()(x)

    x = Conv2D(256, 4, strides=2)(x)
    x = LeakyReLU()(x)

    x = Conv2D(256, 4, strides=2)(x)
    x = LeakyReLU()(x)

    x = Conv2D(256, 4, strides=2)(x)
    x = LeakyReLU()(x)

    x = Flatten()(x)
    x = Dropout(0.4)(x)

    x = Dense(1, activation='sigmoid')(x)
    discriminator = Model(disc_input, x)

    optimizer = RMSprop(
        lr=.0001,
        clipvalue=1.0,
        decay=1e-8
    )

    discriminator.compile(
        optimizer=optimizer,
        loss='binary_crossentropy'
    )
```

Figure 5: Creating our discriminator model

Next, the GAN model combines both the generator model and the discriminator model into one larger model. This larger model will be used to train the model weights in the generator, using the output and error calculated by the discriminator model. The discriminator model is trained separately, and as such, the model weights are marked as not trainable in this larger GAN model to ensure that only the weights of the generator model are updated. This change to the trainability of the discriminator weights only affects when training the combined GAN model, not when training the discriminator standalone. This larger GAN model takes as input a point in the latent space, uses the generator model to generate an image, which is fed as input to the discriminator model, then output or classified as real or fake.

Since the output of the Discriminator is sigmoid, we use binary cross-entropy for the loss. RMSProp as an optimizer generates more realistic fake images compared to Adam for this case. The learning rate is 0.0001. Weight decay and clip value stabilize learning during the latter part of the training. GANs try to replicate a probability distribution. Therefore, we used loss functions that reflect the distance between the distribution of the data generated by the GAN and the distribution of the real data:


```
generator = create_generator()  
discriminator = create_discriminator()  
discriminator.trainable = False
```

```
▶ gan_input = Input(shape=(LATENT_DIM, ))  
gan_output = discriminator(generator(gan_input))  
gan = Model(gan_input, gan_output)  
  
optimizer = RMSprop(lr=.0001, clipvalue=1.0, decay=1e-8)  
gan.compile(optimizer=optimizer, loss='binary_crossentropy')
```

Figure 6: Loss function and optimizer

Rather than just having a single loss function, we need to define three: The loss of the generator, the loss of the discriminator when using real images and the loss of the discriminator when using fake images. The sum of the fake image and real image loss is the overall discriminator loss.

Finally, we train the GAN. Training is the hardest part and since a GAN contains two separately trained networks, its training algorithm must address two complications: They must juggle two different kinds of training (generator and discriminator) and their convergence is hard to identify. As the generator improves with training, the discriminator performance gets worse because the discriminator can't easily tell the difference between real and fake. If the generator succeeds perfectly, then the discriminator has a 50% accuracy. In effect, the discriminator flips a coin to make its prediction. This progression poses a problem for convergence of the GAN as a whole: the discriminator feedback gets less meaningful over time. If the GAN continues training

past the point when the discriminator is giving completely random feedback, then the generator starts to train on junk feedback, and its quality may collapse:

```

start = 0
d_losses = []
a_losses = []
images_saved = 0
for step in range(iters):
    start_time = time.time()
    latent_vectors = np.random.normal(size=(batch_size, LATENT_DIM))
    generated = generator.predict(latent_vectors)

    real = images[start:start + batch_size]
    combined_images = np.concatenate([generated, real])

    labels = np.concatenate([np.ones((batch_size, 1)), np.zeros((batch_size, 1))])
    labels += .05 * np.random.random(labels.shape)

    d_loss = discriminator.train_on_batch(combined_images, labels)
    d_losses.append(d_loss)

    latent_vectors = np.random.normal(size=(batch_size, LATENT_DIM))
    misleading_targets = np.zeros((batch_size, 1))

    a_loss = gan.train_on_batch(latent_vectors, misleading_targets)
    a_losses.append(a_loss)

    start += batch_size
    if start > images.shape[0] - batch_size:
        start = 0

    if step % 50 == 49:
        gan.save_weights('./drive/MyDrive/Colab/gan.h5')

    print('%d/%d: d_loss: %.4f, a_loss: %.4f. (%.1f sec)' % (step + 1, iters, d_loss, a_loss, time.time() - start_time))

    control_image = np.zeros((WIDTH * CONTROL_SIZE_SQRT, HEIGHT * CONTROL_SIZE_SQRT, CHANNELS))
    control_generated = generator.predict(control_vectors)
    for i in range(CONTROL_SIZE_SQRT ** 2):
        x_off = i % CONTROL_SIZE_SQRT

```

Figure 7: Training our GAN model

This training took about a day to complete, we let it run overnight and as it trains the images it is being automatically saved to our drive and it shows the progression of our images from random noise to increasingly real images.

Methods

Real vs Deepfakes

GAN-generated images can be very convincing. Neural networks have gotten alarmingly good at creating realistic human faces. This can be dangerous, since GANs can be used to create fake dating profiles, catfish people, and spread fake information. It is very important for us to be able to distinguish between fake and real images and educate people about this because it can cause societal disruption. The reason GANs are so good is that they test themselves. One part of the network generates faces, and the other compares them to the training data. If it can tell the difference, the generator is sent back to the drawing board to improve its work. There is a possibility that deepfakes can be used to create misinformation on terrorist attacks, generate fake culprits that could circulate online and on social networks which can be very damaging to the society. Therefore, Researchers are developing tools that can spot deepfakes. When looking out for deepfakes there are presently a few things you can spot if you really take a close look and pay attention to. Examples are a surreal background; when GANs are focused on training faces the backgrounds can contain anything. Asymmetry is also a major problem in deepfakes; Ornaments such as earrings may not match in generated images, eyes may be crossed or not looking the same direction or may have different colors or sizes, as well

as ears too. Misaligned teeth can also be another issue you can spot, GANs sometimes shrink or stretch out each tooth in unusual ways. Messy hair or weird hair texture is one of the quickest ways to identify deepfakes. GANs may create random wisps around the shoulders and throw thick stray hairs on foreheads. Hair styles have a lot of variability, but also a lot of detail, making it one of the most difficult things for a GAN to capture. Things that aren't hair can sometimes turn into hair-like textures, too. However, these AI techniques are becoming better every year. 2 to 3 years from now deepfakes would most likely become indistinguishable.

We believe that humans are currently able to detect real versus deepfakes, therefore we advance to state our hypothesis that deepfakes can be identified by humans.

H_0 : There is no difference in rating between Real and deepfakes.

H_a : Deepfakes rate higher on authenticity scale.

Hypothesis Testing

The purpose of statistical inference is to draw conclusions about a population based on data obtained from a sample of that population. Hypothesis testing is the process used to evaluate the strength of evidence from the sample and provides a framework for making determinations related to the population, i.e., it provides a method for understanding how reliably one can extrapolate observed findings in a sample under study to the larger population from which the sample was drawn. The investigator formulates a specific hypothesis, evaluates data from the sample, and uses these data to decide whether they support the specific hypothesis. The first step in testing hypotheses is the transformation of the research question into a null hypothesis, H_0 , and an alternative

hypothesis, H_A . The null and alternative hypotheses are concise statements, usually in mathematical form, of 2 possible versions of “truth” about the relationship between the predictor of interest and the outcome in the population. These 2 possible versions of truth must be exhaustive (ie, cover all possible truths) and mutually exclusive (ie, not overlapping). The null hypothesis is conventionally used to describe a lack of association between the predictor and the outcome; the alternative hypothesis describes the existence of an association and is typically what the investigator would like to show. The goal of statistical testing is to decide whether there is sufficient evidence from the sample under study to conclude that the alternative hypothesis should be believed.

One-tailed vs Two-tailed testing

Two-tailed is appropriate to use if the estimated value is greater or less than a certain range of values, this method is better used for null hypothesis testing. A one-tailed test on the other hand is used if the estimated value may depart from the reference value in only one direction, left or right, but not both. Alternative hypothesis testing is used on one-tailed over null hypotheses.

Null hypothesis: $H_0: \mu = \mu_0$

Alternative hypothesis: $H_a = \mu > \mu_0, < \mu \neq \mu_0$

Equation 1: Null vs Alternative hypothesis

T-test

A t-test is a statistical test that is used to compare the means of two groups. It is often used in hypothesis testing to determine whether a process or treatment influences

the population of interest, or whether two groups are different from one another. When students are choosing the t-test to use, they will need to consider two things: whether the groups being compared come from a single population or two different populations, and whether you want to test the difference in a specific direction. There's one-sample, two-sample and paired t-test. In a One Sample t-test, the test variable's mean is compared against a "test value", which is a known or hypothesized value of the mean in the population. Test values may come from a literature review, a trusted research organization, legal requirements, or industry standards. The two-sample t-test (also known as the independent samples t-test) is a method used to test whether the unknown population means of two groups are equal or not. While the paired t-test is a method used to test whether the mean difference between pairs of measurements is zero or not.

t-test formula:

$$t = \frac{m - \mu}{\frac{s}{\sqrt{n}}}$$

n: number of samples

Equation 2: t-test

Two-sample t-test formula:

$$t = \frac{(m_1 - m_2) - (\mu_1 - \mu_2)}{sp \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

Equation 3: two-sample t-test

Paired t-test Formula:

$$t = \frac{\sum d}{s_p \sqrt{n}}$$

Equation 4: Paired t-test

m = mean

μ = population mean

n = sample size (number of observations)

s = standard deviation

sp = pooled standard deviation

d = differences between all pairs

Analysis of Variance (ANOVA test)

Analysis of variance (ANOVA) is an analysis tool used in statistics that splits an observed aggregate variability found inside a data set into two parts: systematic factors and random factors. The systematic factors have a statistical influence on the given data set, while the random factors do not. Analysts use the ANOVA test to determine the influence that independent variables have on the dependent variable in a regression study. ANOVA is also called the Fisher analysis of variance, and it is the extension of the t- and z-tests. A one-way ANOVA is used for three or more groups of data, to gain information about the relationship between the dependent and independent variables. If no true variance exists between the groups, the ANOVA's F-ratio should equal close to 1.

One-way ANOVA versus two-way ANOVA

There are two main types of ANOVA: one-way (or unidirectional) and two-way. There are also variations of ANOVA. For example, MANOVA (multivariate ANOVA) differs from ANOVA as the former tests for multiple dependent variables simultaneously while the latter assesses only one dependent variable at a time. One-way or two-way refers to the number of independent variables in your analysis of variance tests. A one-way ANOVA evaluates the impact of a sole factor on a sole response variable. It determines whether all the samples are the same. The one-way ANOVA is used to determine whether there are any statistically significant differences between the means of three or more independent (unrelated) groups.

A two-way ANOVA is an extension of the one-way ANOVA. With a one-way, you have one independent variable affecting a dependent variable. With a two-way ANOVA, there are two independents. For example, a two-way ANOVA allows a company to compare worker productivity based on two independent variables, such as salary and skill set. It is utilized to observe the interaction between the two factors and tests the effect of two factors at the same time.

ANOVA formula:

$$F = \text{MSE} / \text{MST}$$

where:

F=ANOVA coefficient

MST=Mean sum of squares due to treatment

MSE=Mean sum of squares due to error



Figure 8: Survey face image sample

We also employed a questionnaire asking participants their perception on AI technology based on their overall familiarity of AI, deep fake generation, reliability and trustworthiness of AI. Below are the 13 questions we asked participants:

Q1. Overall, I am Familiar with AI	Q2. I am familiar with AI being used for Deepfake generation	Q3. I am familiar with using AI technology
Q4. I generally trust AI	Q5. I generally have faith in AI technology	Q6. I feel that AI are generally reliable

Q7. AI is trustworthy	Q8. I believe that AI has my best interest in mind	Q9. Utilizing AI for deepfake generation would possess some risks
Q10. Utilizing AI would involve financial risk	Q11. How would you rate your overall perception of risks from AI (AI is risky)	Q12. I think using AI technology for deepfake generation is convenient
Q13. I can save time by using AI technology		

Table 1: Survey pre-questions on AI technology

Using ANOVA testing we performed an analysis to find if there were demographic differences in participants' perception on AI. ANOVA test results are found in table 4, 5 and 6 below in our results.

Before creating the survey, we had to request Institutional Review Board (IRB) approval since we were performing human subject testing for our research project. The process took about three weeks to be approved after submitting a request.

Results

After receiving our survey results, we performed our data analysis on excel. We created two columns; real and fake images, consisting of the data we received for each image, then we enabled the data analysis add-in on excel and ran our t-test. We chose t-test because we are comparing two groups, real and deepfakes. This test assumes that the different data came from distributions with unequal variances, and it is used to determine whether the samples are likely to have come from distributions with equal population means.

H_0 : There is no difference in rating between Real and deepfakes

H_a : Deepfakes rate higher on authenticity scale

Since $p < 0.05$ we must reject the null hypothesis and accept the alternate hypothesis that states that deepfakes rate higher on authenticity scale. Furthermore, humans are currently able to detect real from deepfakes.

T-test Results:

The data of our real and fake images are highlighted in green and yellow respectively

t-Test: Paired Two Sample for Means		
T-test	Real	Fake
Mean	2.95	3.62
Variance	4.42	4.51
Observations	440.00	440.00
Pearson Correlation	0.16	
Hypothesized Mean Difference	0.00	
df	439.00	
t Stat	-5.16	
P(T<=t) one-tail	0.00000	
t Critical one-tail	1.65	
P(T<=t) two-tail	0.00000	
t Critical two-tail	1.97	

Table 2: T-test results

Demographic Analysis:

When differences among demographic groups were tested, results showed differences in age, gender and race groups. In table 1 we can see that 37% of people that took the survey were within the age group of 28 – 27. Table 2 shows that almost 55% of most people were females and the predominant race of participants were black - 62%.

Age	Choice
18 - 27	34
28 - 37	37
38 - 47	12
48 - 57	8
58 or older	8

Table 3: Age group



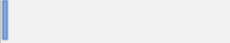
Gender	Choice
Male	 44
Female	 54
Prefer not to say	 1

Table 4: Gender group


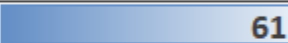



Race	Choice
White	 21
Black or African American	 61
Latino/ Hispanic	 4
Asian	 8
Other	 5

Table 5: Race group

ANOVA test Results:

ANOVA results show that there is significance statistical difference in our pre-questions between demographic groups:

ANOVA						
		Sum of Squares	df	Mean Square	F	Sig.
Q3	Between Groups	17.873	4	4.468	3.253	0.016
	Within Groups	114.025	83	1.374		
	Total	131.898	87			
Q1_8	Between Groups	14.731	4	3.683	3.336	0.014
	Within Groups	91.633	83	1.104		
	Total	106.364	87			
Q1_11	Between Groups	6.782	4	1.696	2.575	0.044
	Within Groups	54.661	83	0.659		
	Total	61.443	87			

Table 6: We tested specific demographics for age

ANOVA						
		Sum of Squares	df	Mean Square	F	Sig.
Q1_2	Between Groups	11.086	2	5.543	4.030	0.021
	Within Groups	116.902	85	1.375		
	Total	127.989	87			
Q1_3	Between Groups	15.649	2	7.825	5.721	0.005
	Within Groups	116.248	85	1.368		
	Total	131.898	87			
Q1_11	Between Groups	7.341	2	3.670	5.766	0.004
	Within Groups	54.103	85	0.637		
	Total	61.443	87			

Table 7: We tested specific demographics for gender

ANOVA					
Q1_13					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	7.205	4	1.801	2.918	0.026
Within Groups	51.238	83	0.617		
Total	58.443	87			

Table 8: We tested specific demographics for race

Discussion

Deepfakes can be identified now because of how unstable GANs are. GANs have a number of common failure modes and while all of these common problems are areas of active research none of them have been completely solved. Some of the major problems scientists are trying to tackle are vanishing gradients and mode collapse. Research has suggested that if your discriminator is too good, then generator training can fail due to vanishing gradients. In effect, an optimal discriminator doesn't provide enough information for the generator to make progress. Mode collapse happens when the generator can only produce a single type of output or a small set of outputs. This may happen due to problems in training, such as the generator finds a type of data that is easily able to fool the discriminator and thus keeps generating that one type. These issues are receiving great attention in GAN research now but as it stands, humans are good at detecting images generated by GAN tech. However, as AI progresses these deepfakes may be impossible to detect. That is why awareness on deepfakes is important now before AI completely takes over and fools us all.

Finally, this work benefits the Human Computer Interaction (HCI) field because of how AI and HCI intersect. HCI is research in the design and the use of computer technology, which focuses on how humans interact with computers and design technologies in novel ways. AI brings to the picture - collaboration between the user and the computer. So many advanced technologies such as Natural language progression as well as speech recognition which brought rise to smart assistants, AI-powered chatbots, Siri, Alexa have improved how humans interact with computers through speech. Other advanced technology including computer vision and neurotechnology has bridged the gap between humans and machines. Scientists and researchers are constantly developing new ways to use machine learning to provide insights into the human mind and improve the interaction between computers, robots, and people.

Conclusion

Conclusively, AI is smarter than humans in so many ways, but the real question is how can we leverage this intelligence for good? The advancement of artificial intelligence has taken an exponential curve for the past few years. Merely 10 years ago, things like Siri and Alexa didn't even exist. Today, we are able to leverage AI to detect cancer from medical images, Google Assistant can book appointments for you over the phone by mimicking human-voice, and developing fake images that are almost flawlessly similar to real images has never been easier before. The widespread concerns regarding privacy and misinformation have shunned the spotlight on deepfakes. In the hands of the wrong person, this technology can be used for fraud. For instance, recently, a deepfaked voice was used to scam the CEO of a UK firm for an amount of \$244,000. The emergence of GANs disrupted the development of fake images. Previously, people have

been using manual methods like photoshop, but with Generative Adversarial Networks, this process is being automated and the results are generally significantly better. Since it is a relatively new neural network first introduced in 2014 by Ian Goodfellow, there are still a lot of concerning issues regarding it that ongoing research are attempting remedies for.

To sum up our paper, we used this research to create awareness of deepfakes and GAN technology. We built a GAN to generate deepfakes of the CelebA dataset, we tested humans on how well they can detect real versus fake images, and using the t-test method we found that people can currently tell the difference between real and fake, which brings us to state our hypothesis that deepfakes can be detected by humans. However, 2 to 3 years from now as deep learning quickly progresses, further research would show how non-detectable deepfakes will become.

References

- [1] I. Goodfellow et al., "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.
- [2] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.
- [3] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," *Advances in neural information processing systems*, vol. 29, pp. 469-477, 2016.
- [4] H. Kwak and B.-T. Zhang, "Generating images part by part with composite generative adversarial networks," *arXiv preprint arXiv:1607.05387*, 2016.
- [5] S. Hitawala, "Comparative study on generative adversarial networks," *arXiv preprint arXiv:1801.04271*, 2018.
- [6] J. Zhao et al., "Dual-Agent GANs for Photorealistic and Identity Preserving Profile Face Synthesis," in *NIPS*, 2017, vol. 2, p. 3.
- [7] H. Yang, D. Huang, Y. Wang, and A. K. Jain, "Learning face age progression: A pyramid architecture of gans," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 31-39.
- [8] J. Song, J. Zhang, L. Gao, X. Liu, and H. T. Shen, "Dual Conditional GANs for Face Aging and Rejuvenation," in *IJCAI*, 2018, pp. 899-905.

-
- [9] J. Zhao, L. Xiong, J. Li, J. Xing, S. Yan, and J. Feng, "3d-aided dual-agent gans for unconstrained face recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 10, pp. 2380-2394, 2018.
- [10] R. Yi, Y.-J. Liu, Y.-K. Lai, and P. L. Rosin, "Apdrawinggan: Generating artistic portrait drawings from face photos with hierarchical gans," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10743-10752.
- [11] S. Gu, J. Bao, H. Yang, D. Chen, F. Wen, and L. Yuan, "Mask-guided portrait editing with conditional gans," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3436-3445.
- [12] Z. Zhang, X. Pan, S. Jiang, and P. Zhao, "High-quality face image generation based on generative adversarial networks," *Journal of Visual Communication and Image Representation*, vol. 71, p. 102719, 2020.
- [13] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8110-8119.
- [14] N. Bayat, V. R. Khazaie, and Y. Mohsenzadeh, "Inverse mapping of face GANs," *arXiv preprint arXiv:2009.05671*, 2020.
- [15] S. Tang, "Lessons learned from the training of gans on artificial datasets," *IEEE Access*, vol. 8, pp. 165044-165055, 2020.

[16] S. Tripathy, J. Kannala, and E. Rahtu, "Icface: Interpretable and controllable face reenactment using gans," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 3385-3394.

[17] L. Zhang and L. Zhao, "High-quality face image generation using particle swarm optimization-based generative adversarial networks," Future Generation Computer Systems, vol. 122, pp. 98-104, 2021.