Facial Composite Generation from Natural Language Text Descriptions using Stacked Generative Adversarial Networks

Michael Ryan
michael_ryan_2000@yahoo.com

I. Introduction

Sketch artistry has been a part of law enforcement since the late 1800s with documented use in wanted posters from the American West and Great Britain [1]. Over the past hundred years, there have been many developments in the generation of facial composites. In the 1950s, a hand-assembled system of designing graphical representations of an eyewitness' memory of a face, known as facial composites, from printed features called Identikit was released [1]. More recently, computerized forms of facial composite generators have taken a presence in law enforcement by building faces from individually described features [2]. The latest technology in the field uses machine learning and evolutionary models to generate facial composites [3].

Despite these advances in facial composite generation, the success rate in using facial composites in criminal investigations has remained staggeringly low. A case study of the Humberside Police revealed that using traditional computer-based face generation software led to suspect identification in 14% of cases [4]. This number is sharply different from the suspect identification rate and conviction rate when using newer technologies such as EvoFIT with 60% and 17% respectively [4]. Still, neither technology nor sketch artistry itself has provided a high success rate. Additionally, at least 80% of wrongful convictions are due to mistaken eyewitness identification [2], which according to criminology research, can be partially attributed to inadequate practices by law enforcement agents [2]. Clearly there is a problem in the way that facial composites are generated and used currently.

This paper proposes an alternative model to generating facial composites using some of the newest developments in machine learning models - generative adversarial networks. This solution focuses on minimizing difficulty for an operator to utilize the model, and maximizing the potential for the user to engage in the psychological process of recognition rather than recall.

II. Related Work

A. Problems with Existing Facial Composite Generation Techniques

Researchers have multiple ideas as to the source of issues in existing facial composite generation technologies. There are two main perspectives that explain the biggest problems in facial composite generation: the psychological perspective and the logistical perspective.

From the human psychology perspective, the problem with existing facial composite generation technologies comes from their reliance on the mental process of facial recall rather than facial recognition. Facial recognition occurs when connecting the ideas of facial features to their spatial representation on the face [5]. This is useful when a person encounters a face and evaluates whether or not they remember this face. Facial recall is used when trying to remember specific aspects of a face [6], such as when describing facial features to an officer constructing a facial composite. Facial recall is generally more difficult than facial recognition [6]. Perhaps the ability of the sketch artist is not the issue, but the difficulty involved in the mental process of recalling specific facial features and putting those ideas into words.

Conversely, the logistical perspective suggests the skill of the people operating the generation technologies is the issue. In a survey of 163 police agencies in 2006, researchers found that 26% of police officers learn how to use facial composite generation tools by trial and error [2]. Additionally, only 68% of officers receive any professional training [2]. This would not be an issue if these technologies were simple enough that both trained and untrained users could generate equally effective composites, but this is not the case. In a study at the University of Aberdeen, a 19% higher success rate was observed in matching composites to pictures when an experienced composite generator operated the facial generation software [7]. Clearly, skill level and lack of training for officers is a contributor to the low success rate of composite generators.

B. Potential Solutions to Issues in Facial Composite Generation Technology

One proposed solution to the issue of lower effectiveness of facial recall compared to facial recognition is to implement facial recognition into composite generation systems. Such a model could operate by showing multiple faces to a witness to find a target which invokes the greatest familiarity [6]. These recognition-based technologies permit a more natural thought process in creating facial composites, because they utilize a holistic approach through showing the witness completed face constructions [4]. The EvoFIT composite generator was designed with these principles in mind. This model shows

completed faces to the user and asks them to choose which best resemble the offender [4] [3].

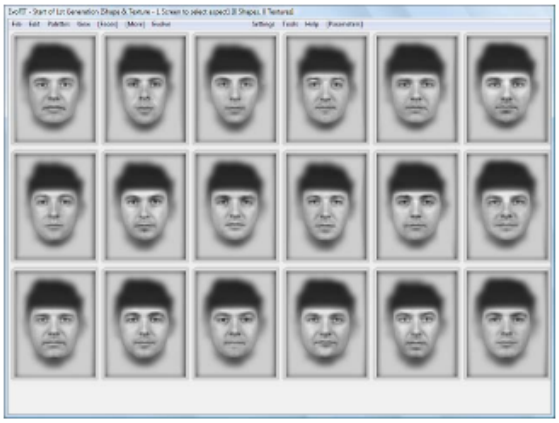

Fig. 1. EvoFIT shows the user multiple completed faces to allow them to select which one most closely resembles the culprit. [4]

EvoFIT utilizes an evolutionary architecture to mix the faces with the greatest resemblance together to create a new set of images [3]. By only asking witnesses to identify features of the criminal to generate the preliminary images, EvoFIT limits the use of facial recall and maximizes the use of facial recognition in the user.

This solution has shown promise. In a laboratory trial, composites generated with EvoFIT were correctly matched to the original face in 25% of trials, as opposed to recall reliant, feature based systems which scored about 5% [5]. Still, there is room for improvement upon this technology. EvoFIT generates faces using an algorithm called principal component analysis (PCA), and it was originally created on a dataset of 72 images [3]. The use of PCA in generating faces with a focus on law enforcement applications was originally proposed in 2000 [8], and PCA has been utilized in conjunction with face manipulation since the early nineties [8]. This does not inherently mean that the technology is bad or completely outdated, however, the field of computer science moves quickly and there have been great strides in facial generation technology over the past eighteen years. Perhaps by looking at developments in facial generation algorithms, it will be possible to improve upon the success of existing proposed solutions.

C. Analysis of New Image Generation Technologies

One recent breakthrough in the field of image generation technology was the invention of Generative Adversarial Networks (GANs) in 2014. GANs set two neural networks in competition with each other, a generator network and a discriminator network [9]. The generator network tries to convert an input of completely random noise to appear similar to data from the training dataset, and the discriminator network tries to determine if the sample came from the generator or the original dataset [9]. This puts the networks in direct competition until the generator output becomes nearly indistinguishable from the original dataset and the discriminator has a fifty-fifty shot of predicting correctly [9]. For instance, in an example where the generator is learning to produce images of faces, this end state would occur when the generated faces look life-like and cannot be distinguished from real pictures (at least not by the discriminator network). See Figure 2.

Fig. 2. Examples of faces generated by a GAN in the original paper proposing the concept from 2014. The yellow boxed images were not generated, rather those were the most similar images to their neighboring image within the training dataset. They were included to prove that the GAN did not merely memorize the dataset of images. [9]

This architecture has shown promise, especially in image generation, often creating very realistic images. Additionally, analysis into the output of GANs has shown that the output is typically represented in a latent vector space logical to human observers [10]. This means that vector arithmetic can be performed on generated images to create images with desired properties [10]. For example, using vector arithmetic, an image of a man with glasses can have a man without glasses subtracted, and a woman without glasses added to produce an image of a woman with glasses [10]. See Figure 3.

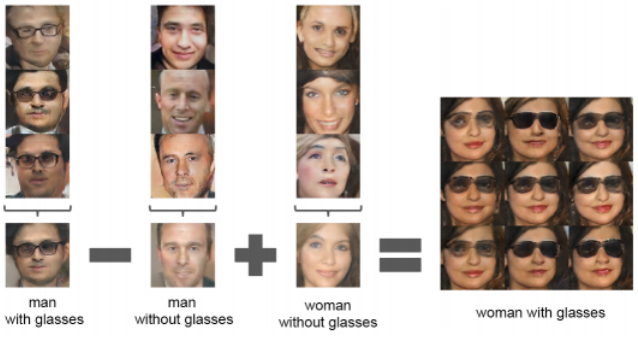

Fig. 3. Vector Arithmetic using Generative Adversarial Networks [10]

This vector arithmetic operation can act as the basis for the facial recognition portion of this model as the user can select the images with the greatest and least resemblance to the culprit for addition and subtraction in vector space.

Additionally, further research using GANs has demonstrated that by providing a separate conditional input to both the generator and discriminator network, specific classes of output can be generated [11]. For instance, the same GAN could generate images of a “hand drawn” number three or a “hand

drawn" number six depending on the input to the network [11]. See Figure 4.

Fig. 4. "Handwritten" digits generated by a conditional GAN. Each row represents a different conditional input. [11]

Parallel research in natural language processing has led to the development of algorithms to convert words and phrases into a latent vector space [12] and representing the text as a series of numbers. This representation of text as a series of numbers is termed a "Text Embedding" or "Text Encoding". By combining these concepts together, computer scientists have been able to vectorize strings of text, feed them as conditions into a GAN, and generate images based on that input text [13]. Using this model researchers were able to generate images of birds and flowers based solely on a few sentences describing those objects [13]. See Figure 5.

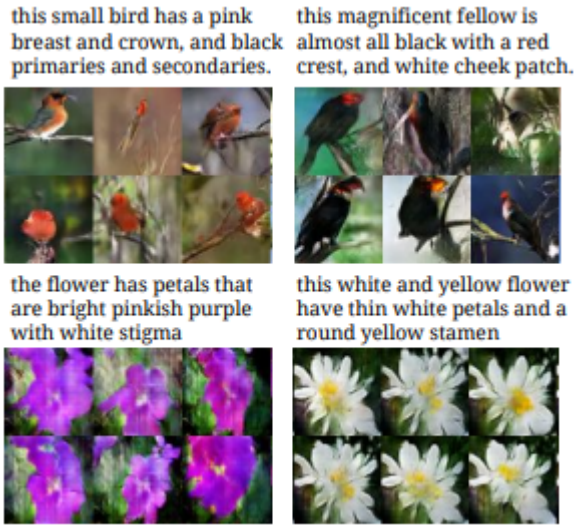

Fig. 5. Images Generated by a GAN conditioned on Natural Language Text Input [13]

It follows that this could be extended to generate faces from text-based descriptions. Perhaps this could help resolve the issue of lack of training with composite generators as an operator would simply transcribe witness observations. A similar idea was implemented in a 2014 paper which generated text descriptions from images of criminals to help match witness descriptions to existing images [14]. This model had a similar concept, but it performed the process in the reverse order, generating text from images rather than images from text. Using these highly effective modern machine learning technologies, it may be possible to improve on the effectiveness of facial composite generators by emphasizing facial recognition and ease of use.

D. Identifying the Gap

The gap in the professional knowledge in facial composite generation remains in analyzing the utility of modern machine learning methods in improving the effectiveness and ease of use over current facial composite generators. These modern machine learning technologies have been developed mostly over the past five years and focus on image generation. As such, it makes sense that these technologies could be useful for generating accurate and high quality images of faces. Specifically, this paper will focus on the effectiveness of using conditional generative adversarial networks and text input, along with producing various outputs to emphasize the mental process of facial recognition as well as ease of use of the model. This will differ from much of the existing knowledge in the field, which is centered around either facial recall methods by building faces only from described features, or facial recognition using older computer facial generation technologies such as EvoFIT.

E. Research Question

After analyzing the professional conversation surrounding this issue in facial composite generation as well as modern developments in machine learning algorithms, the question arises "Would a new conditional GAN-based facial composite generation model, using natural language processing for ease of use and focus on facial recognition, generate verifiably and quantifiably convincing facial composites? Further, would this model improve in effectiveness over existing composite generation models which utilize facial recall methodologies or older facial generation algorithms?"

III. Background

A. Text Embeddings

A text embedding or encoding method is any way to convert a string of text into a numerical representation. Typically this numerical representation is a multidimensional vector. This numerical representation of the input text can later be used as an input condition for a conditional GAN.

One particularly effective method of encoding text in relation to images is the Char-CNN-RNN model, short for Character-Convolutional Neural Network-Recurrent Neural Network [15]. This architecture involves iterating over the input text in fixed length chunks of a certain number of characters. These segments are then compared sequentially. This permits the conservation of temporal information in the vectorized output [15]. See Figure 6. For instance in the description "the man was friendly, but not fast" it is important that the "not" lies before "fast" and not "friendly" as in "the man was not friendly, but fast" as this changes the meaning

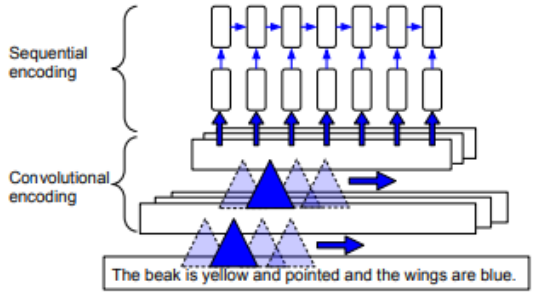

Fig. 6. Graphical Representation of the Processing of Text String into Vector Output. First the text is broken into chunks in the Convolutional Encoding Layer. The Chunks are compressed into preliminary numerical representations and these numerical representations are compared sequentially in the sequential encoding layer. [15]

of the sentence. The Char-CNN-RNN model is considered a high capacity text encoder, because it has many layers of text processing, allowing the network to find trends and patterns in the text [15].

B. Stacked Generative Adversarial Networks

Maintaining stability in training Generative Adversarial Networks is an area of active research in the Computer Science community [16]. One common issue found when training GANs is mode collapse, where a generator model will begin creating images of one color and texture and basing all other images upon this set format [16]. Other issues involve the discriminator network growing too strong at determining the difference between real and generated images and preventing training within the generator model. In an effort to solve some of these issues, especially in higher resolution image generation, Han Zhang et al. proposed StackGAN, a stacked architecture of Generative Adversarial Networks [16].

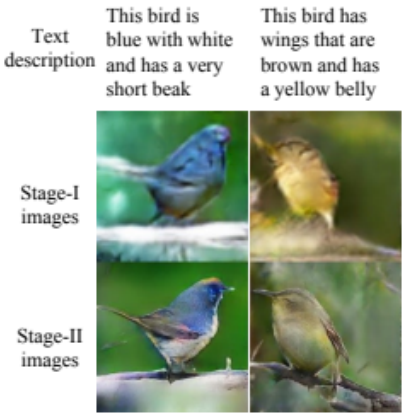

Fig. 7. Examples of Images Generated in Multiple Layers by the StackGAN architecture. The first stage outlines the general shape and colors for the image, while the second stage adds in the details. [16]

In the StackGAN architecture, there are layers of generator and discriminator networks at increasingly higher resolutions. The first generator might output images at 64x64 pixels, the second 128x128, and the third 256x256 for instance. The model would feed the output image of one layer in as the input to the next. The researchers realized that this increased the stability of the network, because some of the lower resolution layers could focus on the shape and basic color of the image, while the higher resolution layers could focus on the finer details [16]. This parallels research from NVIDIA and Aalto University which started training a GAN on low resolution images and progressively scaled up the actual network itself as it grew stronger [17]. Both methods of stacking or growing GANs have strong results for increasing the stability of the model.

C. Evaluation Metrics

In order to assess the quality and diversity of the images produced by a GAN it is common to use an inception score [18]. The inception score makes use of the powerful Inception Model - a neural network previously trained in recognizing and labelling images. In order to calculate the inception score many generated images are fed to the Inception Model of similar initial conditions in order to find the conditional label distribution $p(y|x)$ of the model [18]. This ideally will have low variation as meaningful objects with the same conditions should be labelled similarly [18]. At the same time the overall label distribution of the entire model $p(y)$ would ideally have high variation as a strong model should have a high diversity of generated images [18]. To evaluate the final score these values are fed into equation (1) where the Kullback Leibler divergence (KL) is calculated for both the distributions and the final result is exponentiated for ease of comparison [18].

$$exp(\mathbb{E}_x KL(p(y|x)||p(y))) \tag{1}$$

A higher inception score generally indicates higher quality images with a greater diversity of generated images. This has been shown to correlate well with human evaluations of GANs [18].

IV. Methods

This project had the goal of designing a new machine learning model to generate images of faces from text descriptions. This goal best relates to a "create" approach in undertaking a project, wherein a novel solution to an existing problem is designed and tested. Since every step of this project involves writing and running programs that can take multiple days to reach completion, every stage in the design process requires careful thought and planning. As such, the creation of the model was divided into four stages - constructing the dataset, constructing the text-embedding algorithm, constructing the GAN, and evaluating the model. Evaluating the model will be covered in the Data section of the paper.

A. Collecting and Preparing the Dataset

When collecting my dataset of images with associated text descriptions I needed a large dataset. While Generative Adversarial Networks have been shown to operate remarkably well on a limited dataset of images [19], most high capacity

text embedding algorithms, such as Char-CNN-RNN, have improved performance on large datasets of text and image pairs [15]. I first found the Face2Text dataset, a dataset of 400 images and 1400 descriptions [20]. See Figure 8. While this dataset had a wide variety of quality descriptions of faces, the Face2Text dataset alone did not contain a large number of text and image pairs.

- blonde hair, round face, thin long nose
- While female , American stylish blonde hair and blue or green eyes wearing a suit , public speaks person

Fig. 8. Sample image and matching descriptions from the Face2Text Dataset (This comes from a subset of the dataset that was already publicly released) [20]

To supplement the Face2Text dataset, I found the Large-scale CelebFaces Attributes (CelebA) Dataset, with 202,599 images of celebrities with 40 binary attributes describing each image [21]. See Figure 9. Unfortunately, the binary attributes only described whether or not the image contained a specific aspect, for instance, "Eyeglasses-TRUE or "Wavy hair-FALSE [21]. While this dataset provided a great deal of images, it did not provide the necessary associated text. To solve this I wrote a program in Python that converted the binary attributes into text descriptions. The program included more than 205 unique adjectives and nouns to add variation to the descriptions. The program randomized the order of each attribute in the description, and also randomly chose to write the description in the past or present tense. I made these decisions to maximize the variation in the dataset, because this would allow the text embedding algorithm to better generalize to other descriptions. Although these generated descriptions could not match the variation found in human generated descriptions, I selected this method for creating the text descriptions because I could not write descriptions for 202,599 images in a reasonable time frame. I generated 3 descriptions for every image within the CelebA dataset for a total of 607,797 text descriptions. With so much data, I decided that I would use the Face2Text data as a supplement during the evaluation phase of the project instead.

In order to perform a quality control check on this large dataset, I decided to run a facial recognition Python script over my dataset. This program had a 99.38% accuracy on the Labeled Faces in the Wild dataset [22] [23], so it was proven to recognize faces well. Any image where a face could not be found by the script was discarded from the dataset. This removed images where the subject was not looking at the camera or had their face obstructed by objects. See Figure 10. Using this strategy, I removed 5583 images from the dataset leaving the final dataset at 197016 images and 591048 descriptions. I divided the dataset into 100 sets (16 with 1971 images and 84 with 1970 images) so that I could run the text embedding algorithm and GAN on subsets of the data at a given time.

Fig. 9. Examples of images with different associated attributes sampled from the CelebA dataset [21]

Fig. 10. Examples of images removed from the dataset since the facial detection algorithm could not find a face. (a) not looking at camera. (b) face obstructed. (c) face obstructed. (d) egregious image reshape artifacts.

B. Designing the Text Embedding Algorithm

I selected the Char-CNN-RNN text embedding algorithm because it showed promising results when compared to traditional text embedding algorithms, such as Bag-of-Words or Word2Vec [15] without needing a predefined vocabulary such as Word LSTM. See Table I. This architecture was

especially useful for relating text to images since the model could train a joint encoding between a string of characters and a vector representation of an image [15] unlike models such as Word2Vec which learn to relate words by predicting the following word in a sentence [12]. I decided to use the model proposed in Scott Reed et al. [15], the paper which initially proposed the Char-CNN-RNN model. This meant using the Lua programming language alongside the Torch machine learning library.

TABLE I
ACCURACY OF VARIOUS TEXT EMBEDDING ALGORITHMS [15]

Embedding	Accuracy (%)
Word2Vec	54.6
Bag-of-Words	56.7
Char CNN	51.1
Char LSTM	29.1
Char CNN-RNN	61.7
Word CNN	60.2
Word LSTM	**62.3**
Word CNN-RNN	60.9

Minor modifications needed to be made to my text dataset after selecting this particular model to run the algorithm. The initial Char-CNN-RNN model limited text descriptions to 201 characters long [15], and the input could only be a fixed number of characters long. In order to decide the maximum length for my text I decided to find some statistics on a sample of 19000 generated descriptions. I chose this number because it was a large sample while remaining under 10% of the population, so independence could be assumed. The average length, $\bar{x}$, was 215.433 characters and the sample standard deviation, s_x, was 28.097 characters. I wanted the vast majority of my samples to remain uncut, so I decided to allow the maximum length to be 300 characters. I found the sample proportion $\hat{p}$ of descriptions with fewer than 300 characters and it was 99.86% which made sense since 300 was a bit above 3 standard deviations. In order to ensure a constant length in my text descriptions, I also needed to extend some descriptions. To accomplish this, I zero padded the descriptions (added zeroes until the length was 300) [15].

The Char-CNN-RNN model also required that the images be encoded into a latent vector space prior to evaluating, as the model based the text embedding off of the given image embedding. For this task I again used the face recognition Python library [22]. The library had the ability to generate 128 dimensional vector representations of faces (128 numbers it used to identify key features in a face). This was intended for comparing two faces to determine if the face belonged to the same person [22], but it also worked well as an image embedding. I wrote a Python script to generate these 128 dimensional vectors for each image in the dataset.

With the preparation completed I could begin running the model. I ran the model on a Google Cloud Virtual Machine running Linux, because the Torch machine learning library was designed for Linux and OSX. I started with a learning rate of 0.0007 (the learning rate indicates how quickly the model adjusts to try to improve). This learning rate matched the rate from Scott Reed et al. [15]. I set the learning rate to decay to 98% for every iteration through the dataset. This allowed the model to narrow in on a particular local maximum performance. If the learning rate remains too high the model can continuously overshoot with changes and stop improving too early. I ran the model for about 250 iterations of the the training set, and then I locked the learning rate to 0.00005 and allowed the model to run for 100 more iterations. At this point the loss function, a function which measures how far off the model is from theoretical perfection, had dropped from initial values around 2.0 to around 0.25.

C. Designing the Generative Adversarial Network

I selected StackGAN version two for the generative adversarial network for this project, because it showed marked improvements over other GANs including the original StackGAN [16]. I split the dataset into training and testing data comprised of 80% and 20% respectively of the dataset, because research has shown that GANs perform well on limited training data [19]. With this division of training and testing data, I had text samples that my model had never seen to evaluate the model on. This provided a fairer evaluation, because otherwise it would be possible that the model had just memorized the training data, and could not generalize to unseen data.

I decided to run the model on another Google Cloud Virtual Machine, this time with a Nvidia Tesla T4 GPU to improve run time. I used Python with the Pytorch machine learning library since these were the technologies used to develop StackGAN v2. Using Tensorboard, I was able to log the progress of the model and graphically monitor the training process. Unfortunately my desktop lost connection to the Google VM on the first run and after restarting from the last checkpoint it appeared this hindered progress, so I restarted training from scratch. See Figure 11. On the second run I trained the model on 140,000 images, just under one iteration through the entire training dataset. For the final product, I used the version of the model trained on 76000 images, because it performed better than the more trained versions based both on quantitative analysis (inception score) and a qualitative assessment of image quality from a random sample.

D. Limitations and Future Methods

While these methods did achieve the goal of determining if a conditional GAN model built on text descriptions would generate verifiably and quantifiably convincing facial composites, these methods did not allow for the direct comparison with existing facial composite generation technologies. This was due to limitations in time for the study. With more time I would have tested my model using human participants by following the methods outlined in [5], which have been considered the "gold standard" for determining the success rate of facial generation models for use in criminal identification [5]. The plan was to show subjects a face they were unfamiliar with for 60 seconds. After two days passed, the subject would be administered a cognitive interview and asked to reconstruct

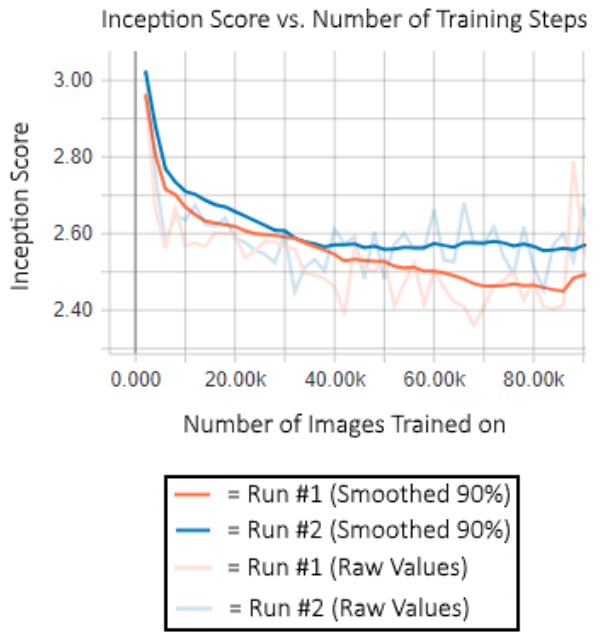

Fig. 11. This graph shows the inception score of each model as a function of the number of images it has trained on (it decreases from the beginning because at the beginning the model has a high variability, but it still hasn't balanced that with realism). This graph highlights the difference between the first and second run of the model. In the first run, the virtual machine lost connection around step 38,000 and needed to load up a checkpoint there. This resulted in a drop in performance which can be seen in the drop in inception score directly after that point. The second run did not suffer this same dip in performance.

the face using my model. Next a second group of subjects would try to identify the faces generated by the initial group. This group would be familiar with the faces that the first group was trying to create, for instance fans of a sports team [5]. They would first need to pass a quick identification test of real images to assess their familiarity with the group of faces that they were trying to identify (i.e. do they really know this sports team?). Then they would be shown generated samples and asked to identify them. This methodology has been used in the past to evaluate various other facial composite generation models [5], so following these methods in the future would allow me to compare my model with existing composite generation techniques. This procedure was presented to and approved by the Institutional Review Board at my high school.

V. Results and Analysis

A. Quantitative Analysis

The quantitative metric selected to analyze this model was inception score. The inception score metric is the most widely used metric for assessing GAN performance [24]. Some researchers criticize inception scores for having misleading representations of image quality [24], however it was the most accessible and common score used for quantitatively assessing GAN performance, so it was selected for this project. The final inception score of the final model was 2.494 after training on 76000 images. This suggests that the model did effectively learn a varied distribution of faces while remaining realistic. A lower inception score could suggest that a total nonsensical mode collapse occurred, where the generator begins generating one type of image that does not resemble the target and has little variation. This was seen after training the model on 128000 images when the inception score dropped significantly to 1.162 by 140000 images. Although little analysis can be drawn from comparing inception scores of different models trained on different datasets, it is worth noting that the original text to image GAN had an inception score of 2.88 on the bird text-to-image dataset [16] [13]. This similar score indicates that this level of performance is not out of the ordinary.

B. Qualitative Analysis

By viewing some of the images generated by the model it appears that there may have been a few nonsensical partial mode collapses, especially while generating images of blond haired females and an older gentlemen. See Figure 13. These images appear blurry and have the outlines of features, but without completing the details. This may have been caused by the discriminator loss dropping too low and thus causing the generator to fail to learn better representations of faces due to the vanishing gradients problem [25]

C. Mixed Analysis

In order to gain a bit more insight into how well the model was converting the original attributes from the text into images of faces, I annotated a random sample of 122 generated images that had recognizable features (the full sample contained 201 randomly selected images, but 79 did not have discernible features). 121 was selected as the minimum sample size, so that the maximum margin of error in each attributes proportion of correctly transferred features would remain under 7.5% at a 90% confidence level. After annotating 122 images, I compared my 40 attributes describing the faces (ie. Blond_Hair=TRUE, Bald=FALSE) to the original binary attributes, before they were converted into a text description and generated as a face. For each attribute I calculated the percentage of attributes from the generated images which matched the values from the original dataset. The top 5 most accurate attributes that appeared at least once in the sample are shown in Table II.

TABLE II
Top 5 Most Accurate Attributes which Appeared in Sample

Attribute	Percent Correct	Images with Attribute in Sample
Male	99.2% ± 1.3%	38.5%
Wearing Hat	99.2% ± 1.3%	4.9%
Double Chin	98.4% ± 1.9%	0.8%
Eyeglasses	95.9% ± 3.0%	4.1%
Chubby	93.4% ± 3.9%	3.3%

However, all but the male attribute scored so highly because it barely showed up in the sample at all. This suggests that for many of these attributes, the model simply did not learn how to generate an image with those attributes at all (ie. could not

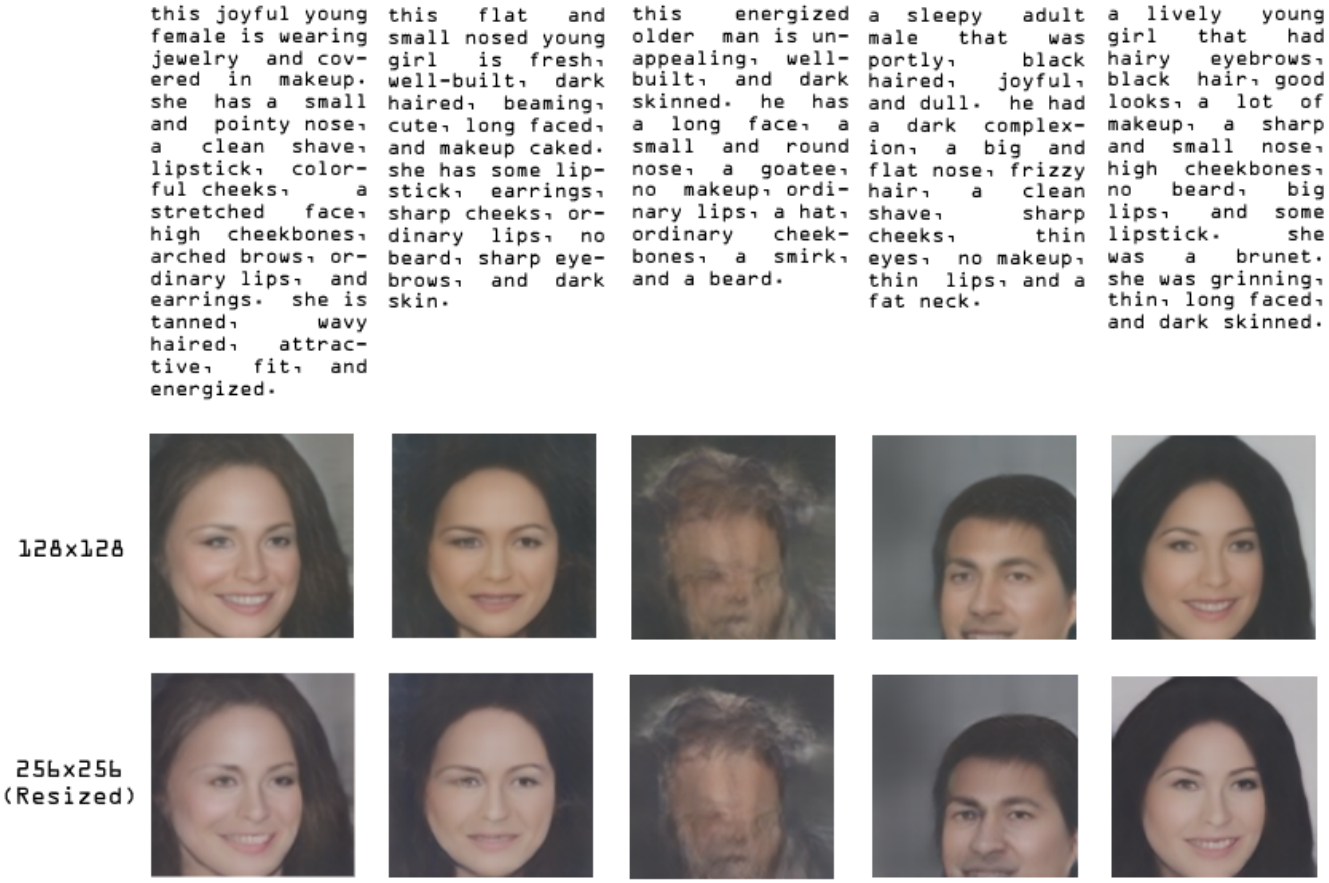

Fig. 12. Random sample of images generated by the final model. Qualitatively it appears that the generator always gets the gender of the person correct, however some of the other features can be a bit random. Additionally the 3rd image is an example of a nonsensical image generated, which make up around 40% of generated images. This may be the result of a mode collapse.

learn how to construct glasses). This explains the high success rate, because if no images were constructed with glasses and very few images were meant to have glasses then the model would receive a high percent correct. As such, a more accurate measure of how well the model learned each attribute would be to look at how the percent correct varied from the amount which would be predicted by always guessing the more likely value. This was calculated using equation (2). The top 5 best attributes as sorted by this metric are shown in Table III.

$$score(x) = \begin{cases} 1 - \hat{p} - \hat{p}_{correct} & \text{if } \hat{p} < 0.5 \\ \hat{p} - \hat{p}_{correct} & \text{if } \hat{p} \geq 0.5 \end{cases} \quad (2)$$

Since the proportion of the dataset which has these attributes is less skewed it is more likely that the model genuinely learned these attributes. As such, it appears that the model really learned how to interpret text to determine the sex of the subject and translate that into a resulting image. It also appears that the model learned how to translate lipstick and smiles well from text to an image. Some of the other attributes such as "Attractive" or "Pointy Nose" have a percentage correct near 50% (59.0% and 46.7% respectively). I believe that this is because some of these attributes are subjective and thus labelling these attributes in the image is akin to random guessing. From this test sample, it appears as though the model only learned to repeatedly capture a few of the attributes from the text description. I theorize that the Char-CNN-RNN did not fully represent these features in the 128 dimensional vector produced. This vector size was much smaller than the size from Scott Reed et al. [15] which was a 1024 dimensional vector. This was due to limitations in the

TABLE III
TOP 5 BEST ATTRIBUTES SORTED BY ESTIMATED MODEL LEARNING STRENGTH

Attribute	Proportion Correct	Images with Attribute in Sample
Male	99.2% ± 1.3%	38.5%
Wearing Lipstick	73.8% ± 6.6%	49.2%
Smiling	71.3% ± 6.7%	48.3%
Heavy Makeup	73.8% ± 6.6%	38.5%
Mouth Slightly Open	59.0% ± 7.3%	50.0%

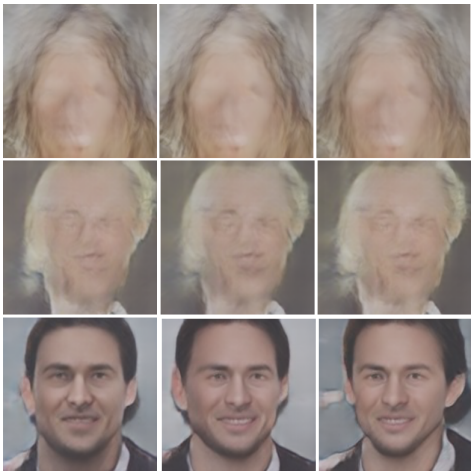

Fig. 13. These are some examples of potential partial mode collapses within the final model. All of these images were independently generated, but they look very similar.

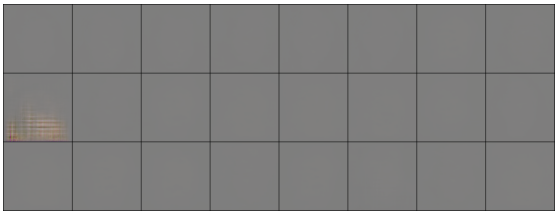

Fig. 14. This is a random sample of images generated by the model after training on 140000 images. Almost all of the images are just gray. It is clear that by this point the model underwent a total mode collapse as no faces can be discerned in any image.

face encoder model from the Python Face Recognition library [22]. Perhaps with a higher dimensional representation of the model, more attributes could be transferred from the text to the image. Additionally, using this same test, I found that the proportion of good, recognizable faces with distinctive features that the model generates is 60.7% $\pm$ 5.7%.

VI. Conclusion

A. Discussion

The goal of this research was to design and implement a generative adversarial and natural language processing based facial composite generator, and to evaluate this generator to determine if the faces produced were verifiably and quantifiably convincing. Another secondary goal was to compare the performance of the model to the performance of existing facial composite generation software. The primary goal of this study was achieved by evaluating the final model using the inception score which quantifiably assesses the realism and variation in GAN output. Furthermore this goal was also pursued through annotating a random sample of generated images and comparing them with the target attributes.

Unfortunately the second goal to this project could not be fully realized due to limitations in the methods of this project. The initial plan for this project would have allowed for the comparison of this model to existing composite generation software based on the "gold standard for facial composite generation models outlined in [5], however this goal was not fully realized due to time constraints after completing the final model. Through a qualitative comparison to existing facial generation techniques it appears that this model did not perform as well as technologies such as EvoFIT (See Figure 15), however such subjective observations can hardly yield solid conclusions in regards to this goal.

Fig. 15. Image generated by a commonly used existing technology called EvoFIT (left) [3], and a face generated by this new model (right).

Although the model produced in this project appears inferior to existing technologies in facial generation, this research can serve as a proof of concept that GANs along with natural language processing show promise as a tool to generate facial composites in an easy to use manner. Additionally the high proportion of correctly generated attributes such as correct sex demonstrate convincing evidence that the model actually learned to convert certain attributes from text to the final image. Furthermore, the ability to generate a wide variety of images by simply altering the input noise to the model demonstrated how simple it is to use a GAN to generate many images and utilize facial recognition in selecting the face rather than just facial recall. All of these findings suggest that with more refinement and tweaking of the model, it would be possible to create a GAN based facial composite generator with far stronger results and with perhaps even comparable or improved performance to existing technologies. Thus, the greater implications of this research are proving the effectiveness of utilizing GANs for facial composite generation and laying the foundation for future refinement of this concept which could revolutionize facial composite generation technologies used by law enforcement agencies.

B. Future work

There are multiple steps researchers could take to further study this topic. One direction is to investigate tweaking this model to attempt to produce a stronger facial composite generator. For instance, I believe that by changing the text embedding from a 128 dimensional vector to a 1024 dimensional vector, much more information would be transferred from the original text description to the final image. This was one of the main drawbacks to the existing model. Similarly, I believe that through tweaking some of the hyperparameters (ie. learning rate) to the GAN, it would produce a wider variety of images because it might be possible to avoid a

mode collapse. This would require running the model multiple times with different settings, and was not possible for this study due to monetary limitations with running the model on the Google Cloud Platform. Finally one other area of future research would be following through with the "gold standard" for comparing facial composite generation models [5] after improving the model in the other manners listed. Overall this project acted as a compelling proof of concept, and future work would mainly involve refining and testing the model.

REFERENCES

[1] K. Taylor, D. Glassman, and B. Gatliff, *Forensic Art and Illustration*. CRC Press, 2000.

[2] D. McQuiston-Surrett, L. D. Topp, and R. S. Malpass, "Use of Facial Composite Systems in US Law Enforcement Agencies," *Psychology, Crime & Law*, vol. 12, no. 5, pp. 505–517, 2006.

[3] C. D. Frowd, "Varieties of Biometric Facial Techniques for Detecting Offenders," *Procedia Computer Science*, vol. 2, pp. 3–10, 2010.

[4] C. D. Frowd, M. Pitchford, F. Skelton, A. Petkovic, C. Prosser, and B. Coates, "Catching Even More Offenders with EvoFIT Facial Composites," in *2012 Third International Conference on Emerging Security Technologies*, pp. 20–26, Sept 2012.

[5] C. D. Frowd, M. Pitchford, V. Bruce, S. Jackson, G. Hepton, M. Greenall, A. H. McIntyre, and P. J. B. Hancock, "The Psychology of Face Construction: Giving Evolution a Helping Hand," *Applied Cognitive Psychology*, vol. 25, no. 2, pp. 195–203.

[6] S. L. Sporer, R. S. Malpass, and G. Koehnken, "Psychological Issues in Eyewitness Identification," pp. viii, 318–viii, 318, 1996.

[7] G. Davies, A. Milne, and J. Shepherd, "Searching for Operator Skills in Face Composite Reproduction," *Journal of Police Science & Administration*, 1983.

[8] P. J. Hancock, "Evolving Faces from Principal Components," *Behavior Research Methods, Instruments, & Computers*, vol. 32, no. 2, pp. 327–333, 2000.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 2672–2680, Curran Associates, Inc., 2014.

[20] A. Gatt, M. Tanti, A. Muscat, P. Paggio, R. A. Farrugia, C. Borg, K. P. Camilleri, M. Rosner, and L. van der Plas, "Face2Text: Collecting an Annotated Image Description Corpus for the Generation of Rich Face Descriptions," *ArXiv e-prints*, Mar. 2018.

[10] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *ArXiv e-prints*, Nov. 2015.

[11] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," *ArXiv e-prints*, Nov. 2014.

[12] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," *ArXiv e-prints*, Oct. 2013.

[13] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative Adversarial Text to Image Synthesis," *ArXiv e-prints*, May 2016.

[14] B. F. Klare, S. Klum, J. C. Klontz, E. Taborsky, T. Akgul, and A. K. Jain, "Suspect Identification Based on Descriptive Facial Attributes," in *IEEE International Joint Conference on Biometrics*, pp. 1–8, Sept 2014.

[15] S. Reed, Z. Akata, B. Schiele, and H. Lee, "Learning Deep Representations of Fine-grained Visual Descriptions," *ArXiv e-prints*, May 2016.

[16] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas, "StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks," *arXiv e-prints*, Oct. 2017.

[17] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive Growing of GANs for Improved Quality, Stability, and Variation," *ArXiv e-prints*, Oct. 2017.

[18] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved Techniques for Training GANs," *ArXiv e-prints*, Jun. 2016.

[19] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep Image Prior," *ArXiv e-prints*, Nov. 2017.

[21] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep Learning Face Attributes in the Wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[22] A. Geitgey, "face recognition." Github.com, Mar. 2017.

[23] D. E. King, "Dlib-ml: A Machine Learning Toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[24] D. Jiwoong Im, H. Ma, G. Taylor, and K. Branson, "Quantitatively Evaluating GANs With Divergences Proposed for Training," *ArXiv e-prints*, Mar. 2018.

[25] M. Arjovsky and L. Bottou, "Towards Principled Methods for Training Generative Adversarial Networks," *arXiv e-prints*, Jan. 2017.

[26] S. Li, T. Xiao, H. Li, B. Zhou, D. Yue, and X. Wang, "Person Search with Natural Language Description," *ArXiv e-prints*, Feb. 2017.