Research Title:

Utilizing Generative Models for generating Face Sketches with Human Judgements for Police Investigations

Name: Jangwon Seo

Supervised by Karan Grewal

Abstract

Police sketch artists often produce an unreliable sketch of criminals, causing hundreds of innocent citizens to be imprisoned in the lack of evidence (DNA). In this project, I suggested a new approach to sketch-production by introducing the concept of the generative model. I tested if involving this concept can generate sketches that closely resembles the target face.

I utilized pre-trained generators to generate high-quality images. First, a python code that uses the GAN (Generative Adversarial Network) generator produces a series of images. The code then accepts ratings from the user as to how well each generated image resembles the target image. The variable that the generator generates its images from was then updated using the accepted ratings to produce images similar to the pre-produced images with higher ratings.

The code showed its best performance at the step size of 0.5. In terms of datasets used, the Celeb-AHQ dataset showed less bias than the FFHQ dataset in terms of race, gender, and age orientation of the generated images. Overall, the code managed to produce images that resemble a target image taking the user ratings into a full account.

This study shows that involving a generative model can be effective in generating images for criminal investigation. With a careful selection of dataset, depending on the country of usage, and detailed engineering of the program, using a generative model in police sketch production can outperform the existing system and eliminate the issues of innocent citizens falsely accused due to an unreliable sketch.

Introduction

False accusation of crimes is a major problem in society, with hundreds of innocent people going to jail in the United States of America alone. According to the Innocence Project (Innocence Project, 2020), about 69 percent of the falsely accused citizens in the US involved eyewitness misidentification. With sketches produced by sketch artists and computer algorithms given a major relevance in the criminal investigation in the lack of clear evidence, it has been the major cause of numerous innocent citizens serving multiple years in jail. In addition to this, states across the globe are facing a tremendous cost in hiring the sketch artists, who are paid about 60,000 US dollars on average in the United States. With the current system of criminal sketch production raising issues of falsely accusing the innocent and high costs, the system must consider a new approach in sketch production that will minimize these issues.

A picture of Kirk Bloodsworth (left), who was sentenced to death, and served 9 years in prison. His sentence largely relied on the police sketch (right) produced by a sketch artist. He was later exonerated by a DNA testing after 9 years of imprisonment.(Source: https://www.sutori.com/story/kirk-bloodsworth--pJxZ9AWXVEHQjketW1zVMKYk)

In this project, I intend to suggest a new approach in criminal sketch production by introducing machine learning technology, generative models to be specific. Generative model refers to a technology that utilizes machine learning (a concept in Artificial Intelligence that uses data to learn and improve) to generate new data points by learning data distribution of the training set available to the model.

Utilizing generative models in producing sketches will function by taking an approach different from the one used by the already-existing methods. Originally, most computer

programs and sketch artists enquire for specific details from eyewitnesses in sketching the face of the criminal, for example, they will ask for a specific shape of the nose for them to sketch. However, as different research studies suggest, this approach fails to produce reliable face sketches due to human eye's nature: we view human faces as a whole instead of viewing them in separate features, in other words, we have a holistic view of human faces. The approach that will be used for my project will involve human users viewing a fully produced face and providing a rating as to if the sketch produced resembles the intended human face. This will allow the model to take into full account the holistic memory the user has of the human face.

Recently, there have been developments in a few computer algorithms, including EvoFIT, where implementation of involving holistic dimensions to their software was attempted. The new development involved displaying about 70 available sketches for the eyewitness to select the sketches that closely resembled the criminal. The algorithm then combines the features of the selected sketches to generate the final sketch.

However, the main flaw in this recent development is that it's quality and reliability of produced sketches are limited to the sketches available on their database. The reliability of the produced sketches is highly dependent on the variety of the sketches primarily shown to the eyewitness, and the quality is highly dependent on the quality of the sketches available on the database. With the reliability of the produced sketch being the primary importance in the sketch producing system involving generative models, it must place its ultimate goal in producing the most reliable sketches with the highest quality, as fast as possible. This implies that the system will make use of pre-trained generative models and online datasets to maximize its quality and reliability. To be specific, I will be using 2 pre-trained generators for this project, which are StyleGAN and Progressive GAN. These two generators use the FFHQ database and the Celeb-AHQ database, respectively.

(StyleGAN: https://arxiv.org/abs/1812.04948, FFHQ dataset: https://github.com/NVlabs/ffhq-dataset, PGAN: https://arxiv.org/abs/1710.10196, Celeb-AHQ dataset: https://github.com/tkarras/progressive_growing_of_gans)

I hypothesize that by the end of this project, the engineered program will generate images that successfully resemble the target face.

Methodology

Loading the pre-trained generator

Training a generative model can take a long time, sometimes taking up to months. As such, I will be using a pre-trained generator available online which will allow me to generate high quality images without having to train the models. To be specific, I will be using the StyleGAN generator that uses the FFHQ dataset.

To load the generator:

```
use_gpu = True if torch.cuda.is_available() else False
device = torch.device("cuda:0" if use_gpu else "cpu")
model = torch.hub.load(
    github="facebookresearch/pytorch_GAN_zoo",
    model="StyleGAN",
    pretrained=True,
    useGPU=use_gpu
)
```

For the experimentation aspect of my project, I varied the dataset and the generator, for the reasons that will be explained in the experimentation section of the report. The other generator I used was Progressive GAN (PGAN) that utilizes the Celeb-AHQ dataset.

The iteration procedure

During each iteration, 8 images are generated from the latent variable, ratings are accepted from the user, and the latent variable is updated based on the accepted user ratings. First, the latent space is defined:

```
z = torch.zeros((dim_z,))
```

Here, latent space refers to, simply put, a series of 'compressed version' of an image that goes through the generator to produce the actual image. Different points in the latent space produce different images, and points that produce similar images are grouped closely within the latent space. Here's a diagram to aid the explanation:

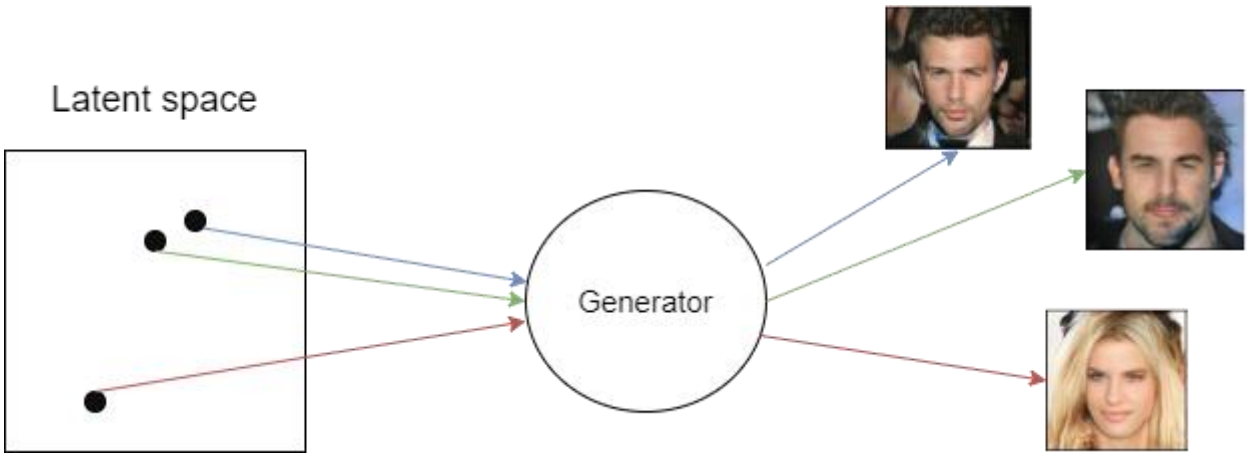


Fig 1. How image(s) are produced from a latent space

The iteration process then begins using a for loop, which is illustrated by the block of code below:

```
for iteration in range(1, n_iterations + 1):

    print("=" * 40)
    print(" Iteration {}/{}".format(iteration, n_iterations))
    print("=" * 40)
    noise = torch.randn((batch_size, dim_z))

    with torch.no_grad():
      z_noised = z.repeat(batch_size,1) + noise*0.1
      generated_images = model.test(z_noised)

    plot_images(generated_images)
```

As the loop that iterates begins, noise is defined. Here, noise refers to a random set of values with the shape being the number of images to be produced by the dimensions of the latent variable. Noise is generated to influence the latent variable by a margin to generate images that vary and are not identical.

Noise is added to the latent variable, 'z', which runs through the generator to produce images.

This is followed by a code that compiles the rating data accepted from the user:

```
  F = []
      for image_num in range(0, batch_size):
          F_n = get_user_rating(image_num)
          F.append(F_n)
       F = np.array(F, dtype=np.float64)
```

Updating the latent variable

At the end of each iteration, latent variable z is updated based on user ratings. With different approaches available for updating the latent variable, I coded 4 methods for updating the latent variable. The ultimate goal for all methods is to move the latent variable towards the area in the latent space that produces the best image, by calculating a vector to be added to the latent variable depending on the ratings.

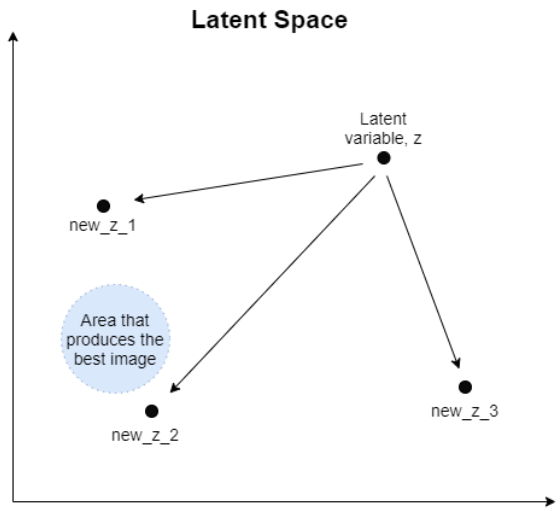


Fig 2. How latent variable is updated. In this figure, the goal will be to update z towards the blue area.

Algorithm for method 1:

```
def method_1(z_noised, F):
  var = np.dot(F, noise)
  sum_F = np.sum(F)
  average = var/sum_F
  new_z = z + average*learning_rate
  return new_z
```

This method aims to calculate the average of the noise applied in the previous iteration weighted by the ratings they received. This is added to the original latent variable to produce a new, updated latent variable. By doing so, this method allows consideration of all images' and their ratings in calculating the vector that will be added to the latent variable. This approach is built upon the mathematical formula below (Equation 1):

$$z_1 = z + \frac{1}{sum(F)} \sum_{i=1}^{n} F_i \varepsilon_i$$ - Equation 1.

Where n is the number of images produced, F is the list of the ratings, and epsilon (ε) is the noise that was added to the latent variable

Algorithm for method 2:

```
def method_2(z_noised, F):
  max_indx = np.argmax(F)
  update = noise[max_indx, :]

  new_z = z + update*learning_rate

  return new_z
```

This method also builds upon a mathematical equation below (Equation 2):

$$z_1 = z + \varepsilon_i$$ – Equation 2.

Here, epsilon base i (ε_i) refers to the noise applied to the image with the highest rating

This method identifies the noise added to a specific image that received the highest rating, which is added to the original latent variable to produce an updated latent variable. By doing so, this method aims to only consider one image that the user considers to be the best in terms of its resemblance to the target image.

Algorithm for method 3:

```
def method_3(z_noised, F, inp):
  idx = F[np.argsort(F)[-inp:]]
  mask = (F >= idx[0])
  F = F * mask
  sum_F = np.sum(F)
  var = np.dot(F, noise)
  sum_F = np.sum(F)
  average = var/sum_F
  new_z = z + average*learning_rate
  return new_z
```

This method is quite similar to method 2 since it only considers specific images to update the latent variable. However, in this method, the code takes the top n pictures that received the best ratings. In other words, if the user decides to consider only 3 images from 8 generated images, the code will pick the images with the top 3 ratings. When that is done, it takes the same procedure as the first method where the weighted average of the picked images is calculated for it to be added to the latent variable.

Mathematically, this will imply that n in equation 1 will now be the number of images considered in updating the latent variable.

Algorithm for method 4:

```
def method_4(z_noised, F, threshold):
  mask = (F >= threshold)
  F = F * mask
  sum_F = np.sum(F)
  var = np.dot(F, noise)
  sum_F = np.sum(F)
  average = var/sum_F
  is_all_below = np.all((F <= threshold))
  if is_all_below:
    new_z = z
  else:
    new_z = z + average*learning_rate
  return new_z
```

This method only considers the images that received ratings higher than a given threshold. The latent variable is then updated by taking the same procedure as method 1 and 3, where the weighted average of the images is calculated. This method is also very similar to method 3 in the sense that it picks the best n images to be considered for updating, although this method uses a threshold to select the best n images.

Results and Analysis

For the experimentation aspect of my project, I varied the following data to observe its effects on the sketch production:

1. The step size (defined as learning_rate in the code)
2. Datasets used by the generator
3. Methods of updating the latent variable

Note:

- Due to the nature of my project where evaluation of images had to be done, quantitative analysis of my results could not be done.
- There may be a slight bias in the ratings inputted to the program due to the lack of human subjects to provide ratings.

Results

Varying step size for both datasets

Step size is the variable that controls the degree by which the latent variable is updated. In other words, it acts as a numerical limitation to prevent the updating of the latent variable by an excessively large margin.

In varying the step size, I focused on observing how the different values for step size influence the procedure by which the latent variable is updated, and determining the best step size value for each dataset.

In this variation, I attempted producing an image with this query: “Young, light-skinned female with blonde hair”

I used 3 values for step size: 0.1, 0.3 and 0.5

The results of this experiment for the two datasets are illustrated in the next page:

1. Celeb-AHQ

Below images show the images produced by the program for different learning rates.

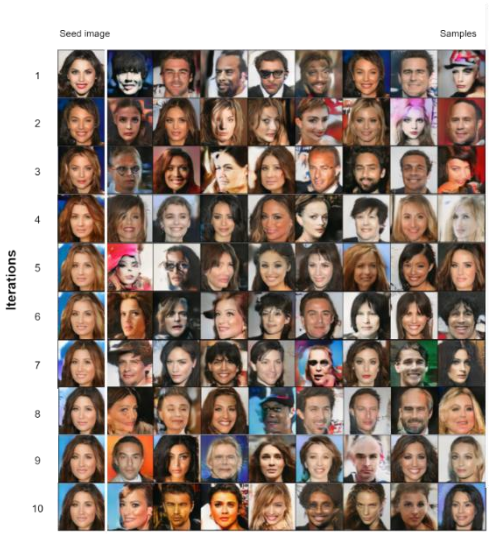


Fig 3.1. Images produced at Learning_rate = 0.1 for Celeb-AHQ dataset

Fig 3.2. Images produced at Learning_rate = 0.3 for Celeb-AHQ dataset

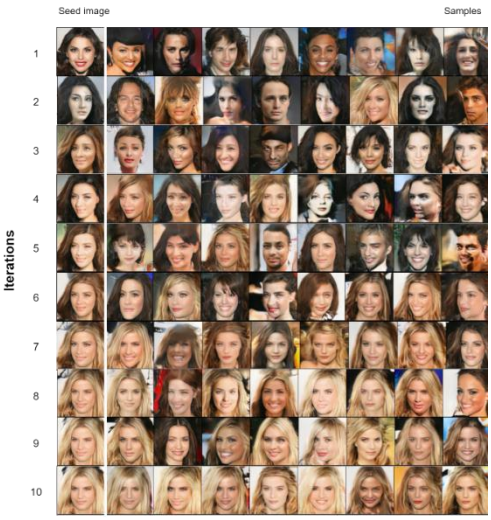


Fig 3.3. Images produced at Learning_rate = 0.5 for Celeb-AHQ dataset

As it can be observed from the images, greater value of step size showed a better performance by approaching to the images with the intended features faster than when the step size was smaller. Although learning_rate of 0.3 showed a similar performance as 0.5, it can be observed that the sample images produced in the first iteration coincidently contained the intended features which must have affected its performance positively. Hence, I concluded that with the Celeb-AHQ dataset, step size value of 0.5 shows the best performance.

2. FFHQ

Fig 3.4. Images produced at Learning_rate = 0.1 for FFHQ dataset

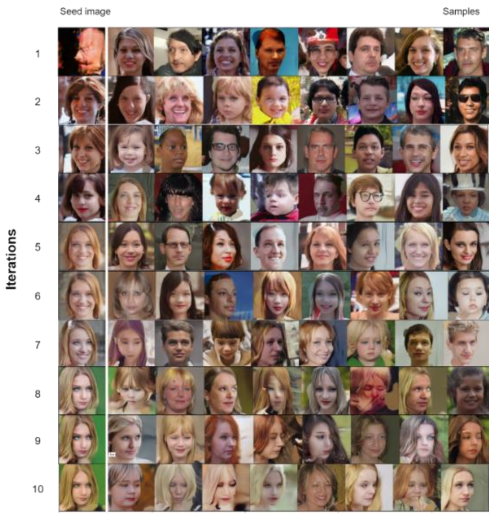


Fig 3.4. Images produced at Learning_rate = 0.1 for FFHQ dataset

Fig 3.4. Images produced at Learning_rate = 0.1 for FFHQ dataset

Similarly, the FFHQ dataset showed a better performance with greater step size values. I also concluded that the step size value of 0.5 shows the best performance with the FFHQ dataset

Varying datasets

In varying the datasets, I attempted producing images with three different queries to determine the feature-by-feature bias for each dataset. The three datasets are:

Query 1: “Young, light-skinned female with blonde hair”

Query 2: “Middle-aged, dark-skinned male with afro-textured hair”

Query 3: “Old, Asian male with dark and short hair”

Note: For both datasets, step size value of 0.5 was used.

The results for this experiment are illustrated in the next page:

1. Celeb-AHQ

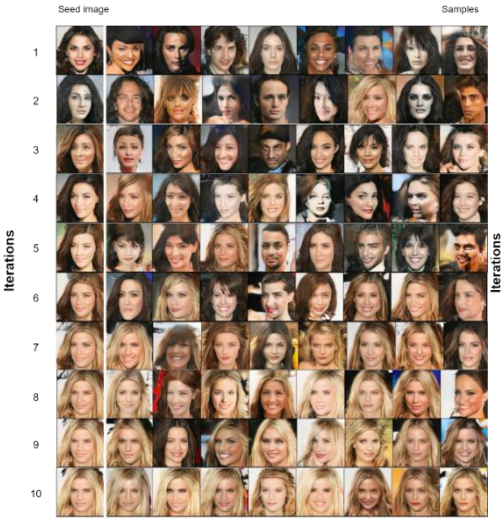


Fig 4.1. Query: "Young, light-skinned female with blonde hair", Celeb-AHQ

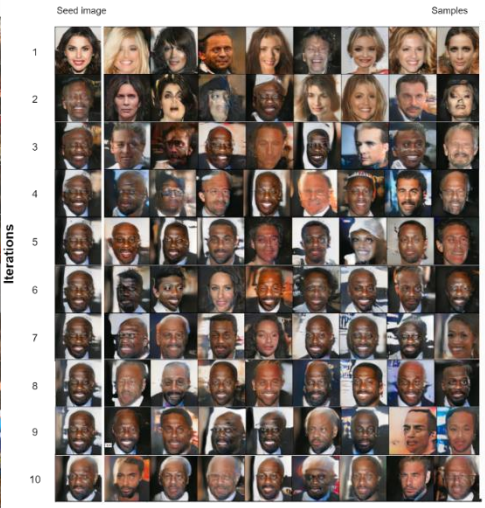


Fig 4.2. Query: "Middle-aged, dark-skinned male with afro-textured hair", Celeb-AHQ

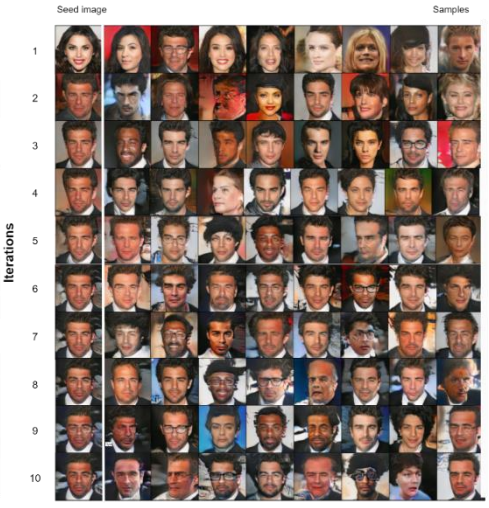


Fig 4.3. Query: "Old, Asian male with short black hair", Celeb-AHQ

Here are the observations from producing different images with Celeb-AHQ:

- It had no problem in producing white and black people, however, the database showed a huge bias against producing Asian people. As it can be observed from fig 4.3, the database failed to produce any image with Asian characteristics at the end of the procedure. This may be due to the fact that the database consists of images of Hollywood celebrities which is dominated by the white and black population.
- It showed no bias in terms of gender.
- It showed a consistent performance in terms of facial expressions and direction. The majority of the images contained people smiling and facing the camera.

2. FFHQ

Fig 4.4. Query: "Young, light-skinned female with blonde hair, FFHQ database

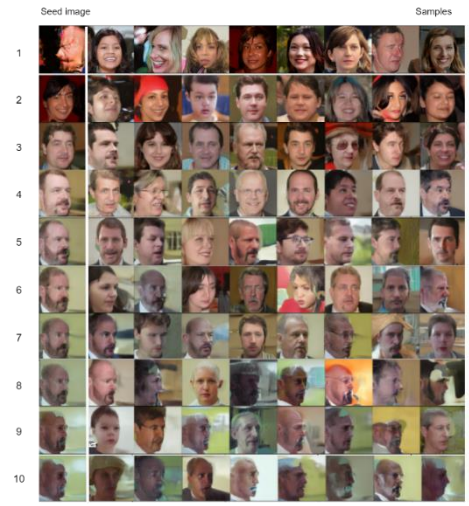


Fig 4.5. Query: "Middle-aged, dark-skinned male with afro-textured hair", FFHQ database

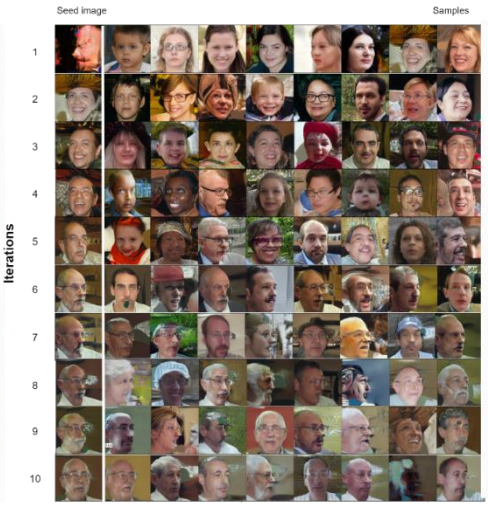


Fig 4.6. Query: "Old, Asian male with short black hair", FFHQ database

Here are the observations from producing different images with Celeb-AHQ:

- It showed a rather more extreme racial bias, as it failed to produce images of black and Asian people, although it had no problem in producing images with white people.
- It showed no bias in terms of gender.
- Unlike the Celeb-AHQ database, the images produced varied greatly in facial expressions and directions.

Therefore, although both datasets demonstrated some amount of bias in different categories, Celeb-AHQ showed less bias as compared to the FFHQ dataset.

Varying the methods used to update the latent variable

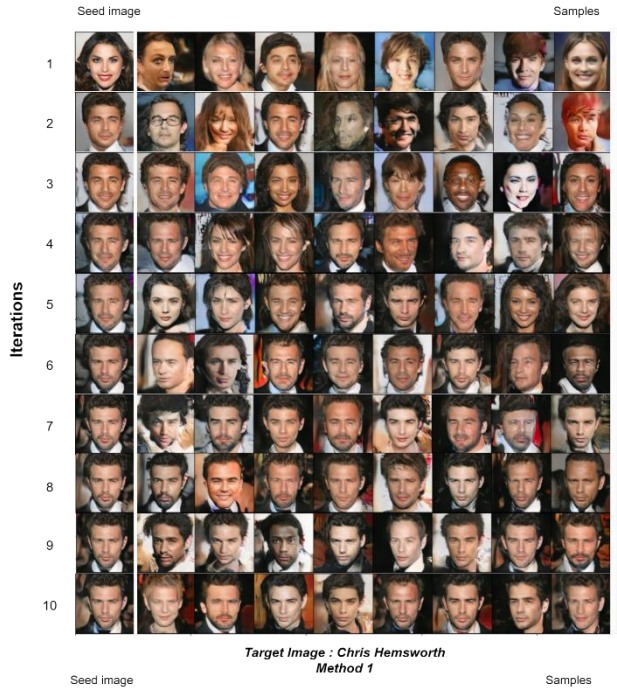
In varying the methods, I attempted producing an image that resembles closely to a chosen target image.

I decided to use the following:

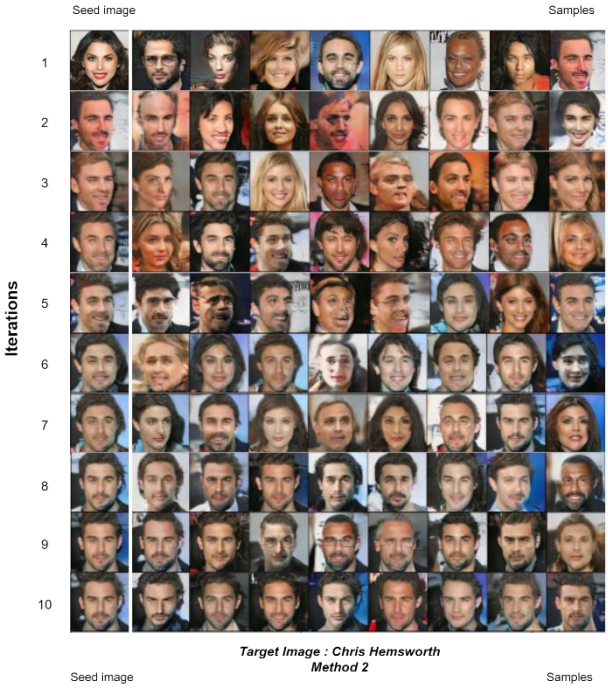
- Dataset: Celeb-AHQ. This was because the previous experiments revealed that it shows less bias compared to FFHQ dataset.
- Step size: 0.5
- Target image: Image of Christ Hamsworth. I decided to use an image of a celebrity to take full advantage of the dataset which consists of images of celebrities.

Target image:

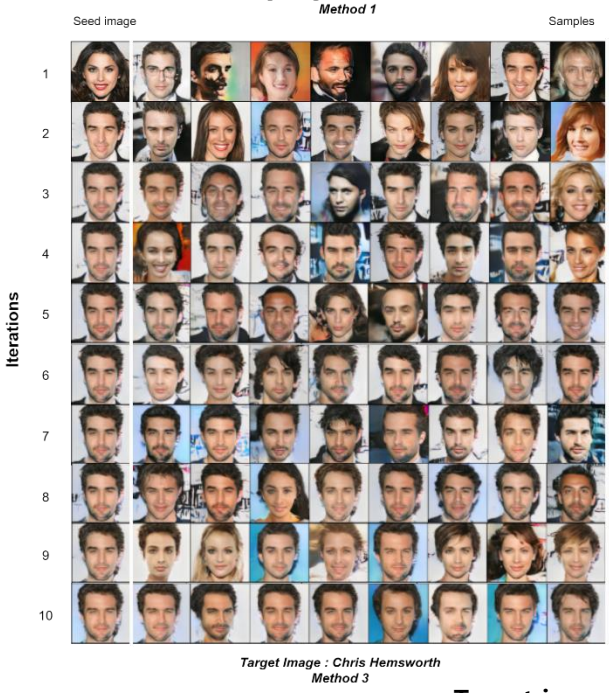
LONDON, ENGLAND - APRIL 10: Chris Hemsworth attends the "Avengers Endgame" UK Fan Event at Picturehouse Central on April 10, 2019 in London, England. (Photo by Dave J Hogan/Dave J Hogan/Getty Images)



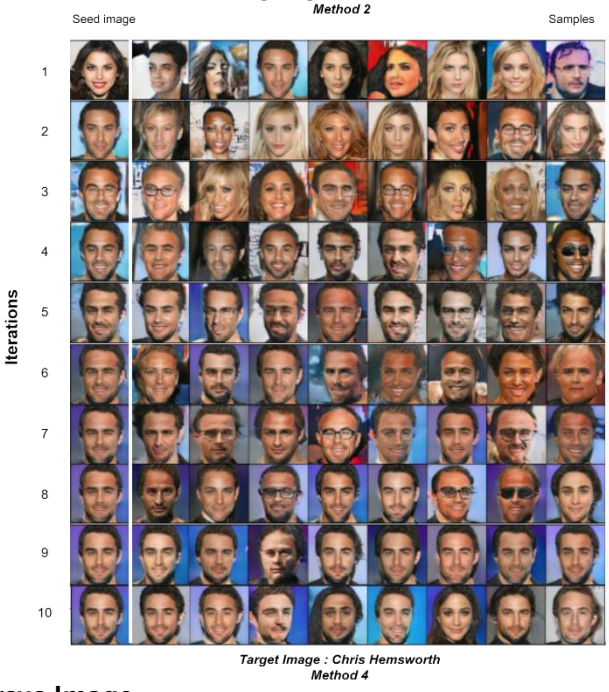
Target Image : Chris Hemsworth
Method 1



Target Image : Chris Hemsworth
Method 2



Target Image : Chris Hemsworth
Method 3



Target Image : Chris Hemsworth
Method 4

Target image versus Image produced during the final iteration

Target image

Method 1

Method 2

Method 3

Method 4

The 4 methods did not show any significant differences in the image producing procedure.

From the finally produced images, I considered the following to evaluate the qualities of the produced iamges:

- Does the produced image have the basic facial features of the target face?
- Does the image resemble the target image at a glance? (Subjective)

From these criteria, I concluded that method 1 performed the best as it produced an image that best resembles the target image.

Note: This conclusion is subjective to my view of the images.

Reference: Table 1. Variables used for each experiment

Variables\Experiments	**Varying learning_rate**	**Varying dataset**	**Varying updating methods**
Batch_size	8	8	8
Learning_rate	0.1, 0.3, 0.5	0.5	0.5
Updating method	Method 1	Method 1	Method 1, 2, 3, 4
Dataset	Celeb_AHQ, FFHQ	Celeb_AHQ,FFHQ	Celeb_AHQ

Conclusion

In this project, I have studied the potential generative model has in the field of crime investigation. My results are a clear indication that involving the generative model in criminal sketch production has a huge potential, as I was able to produce multiple images that resemble the target image with a minimal amount of code, and determine the best values for the different variables that I involved in the experimentation. Hence, with a careful selection of database depending on the country using the program and more detailed engineering of the code, the generative model will outperform the currently existing sketch artists and computer algorithms, and significantly reduce the possibility of an innocent citizen being sentenced to imprison due to an unreliable police sketch.

Scope of Research

This research needs further studies and exploration if it is to be used in the criminal investigation process of a country. This could be done by the following:

- Training of GAN generator with a unique dataset:

  If this research could be recognized by the police department and is implemented, training of a new GAN generator using a new dataset will be required, and will significantly advance the quality of the images produced. A suggestion would be to use the pictures of the citizens' face (possibly from the citizens identification database of the police department) as a dataset to train the generator. This will allow the generator to produce images that are relevant in terms of the citizens' general race, etc.

- Experimentation with lower step size and higher iteration number:

  Step size limits the progression of the program; in other words, if the step size is very small, it will allow a more detailed updating of the latent variable, however, this will require a higher iteration number to allow the final image to resemble the target image. Reducing step size and increasing the iteration number will make the program more time consuming, but it will allow a more delicate updating of the latent variable

and hence a higher chance of the image resembling the target image, with incorporation of little details that may have been ignored with higher step size and lower iteration number.

- Experimentation using human subjects

  A suggestion would be to allow the human subjects to view an image of a person for a few seconds (the average amount of time an eye-witness is exposed to see the criminal in an actual crime would be ideal), and letting the subjects produce images using the program to resemble the image shown to them. When this is done with a large number of human subjects, the justification of this research will be more reliable and relevant.

Acknowledgement

I would like to show my greatest gratitude for Karan Grewal, my research supervisor, for spending his time guiding me through the research process. Also, I would like to express my appreciation for Joel Dogoe, the executive director of MISE foundation, for opening me up to this research opportunity.

Bibliography

Mcquiston, Dawn & Topp, Lisa & Malpass, Roy. (2006). Use of facial composite systems in US law enforcement agencies. Psychology Crime & Law - PSYCHOL CRIME LAW. 12. 505-517. 10.1080/10683160500254904.

Zahradníková, Barbora & Duchovičová, Soňa & Schreiber, Peter. (2016). Facial composite systems: review. Artificial Intelligence Review. 1-22. 10.1007/s10462-016-9519-1.

Frowd, Charlie & Bruce, Vicki & Mcintyre, Alex & Ross, David & Stephen, Fields & Plenderleith, Yvonne & Hancock, Peter. (2006). Implementing Holistic Dimensions for a Facial Composite System. Journal of Multimedia. 1. 10.4304/jmm.1.3.42-51.

"Innocence Project." *Innocence Project* , 28 July 2020, www.innocenceproject.org/ [Accessed 12 August 2020].

"Sketch Artist Annual Salary ($58,004 Avg: Jul 2020)." ***ZipRecruiter*** **,** www.ziprecruiter.com/Salaries/Sketch-Artist-Salary [Accessed 12 August 2020].

Sutori.com. 2020. *Sutori*. [online] Available at:

https://www.sutori.com/story/kirk-bloodsworth--pJxZ9AWXVEHQjketW1zVMKYk [Accessed 16 August 2020].

Karras, T., Laine, S. and Aila, T., 2020. "*A Style-Based Generator Architecture For Generative Adversarial Networks*." [online] arXiv.org. Available at: <https://arxiv.org/abs/1812.04948> [Accessed 16 August 2020].

Karras, T., Aila, T., Laine, S. and Lehtinen, J., 2020. *Progressive Growing Of Gans For Improved Quality, Stability, And Variation*. [online] arXiv.org. Available at: <https://arxiv.org/abs/1710.10196> [Accessed 16 August 2020].

GitHub. 2020. ***Nvlabs/Ffhq-Dataset***. [online] Available at: <https://github.com/NVlabs/ffhq-dataset> [Accessed 16 August 2020].

GitHub. 2020. ***Tkarras/Progressive_Growing_Of_Gans***. [online] Available at: <https://github.com/tkarras/progressive_growing_of_gans> [Accessed 16 August 2020].