KOÇ UNIVERSITY

College of Engineering

ELEC 491 – Electrical Engineering Design Project

Final Report

Criminal Face Prediction from Police Sketches

Participant information:

Oğuzhan Yıldırım 60181

Rıdvan Balamur 53569

Project Advisor(s)

Hakan Ürey

10/06/2021

Table of Contents

Abstract

For years, police forces have tried to find the criminals using the composite sketches with the help of eyewitnesses. Even computer-aided robot drawings, which have been used after the latest technological developments, still have a very low accuracy rate for identifying the suspects. One of the reasons for this is that the criminals can easily get away from the sketch by changing their facial attributes such as hair type, color, beard and mustache or they may even get older. The way to overcome this is to predict possible changes in suspects and increase the accuracy of criminal identification with face recognition systems. In this project, we have created a novel tool that can generate realistic faces from police sketches and change certain facial features of the generated face. Using most recent deep learning methods gave us the opportunity to create an easy-to-use web page. With this project, we believe the usefulness of police sketches will increase dramatically.

Section 1: Introduction

Police sketches are the graphical representations of a suspect drawn by forensic artists with the help of one or more eyewitnesses' memory. Those sketches are an assistant tool for security forces and play an important role to identify the criminals. However, the accuracy rate of correct detection of the criminals using police sketches is not that high. The human memory is vulnerable to errors especially in stressful situations, criminals' real appearance may easily be affected by the eyewitness' memory. Moreover, criminals tend to change their appearance by changing their hair style and color, wearing glasses, having a beard and moustache. Even if they do not change their appearance on their own, they may even change naturally if they cannot be caught up for a very long time. They may age by time and their similarity to the initial sketch may be reduced.

In this project, we have designed a software tool that will help security forces to identify the criminals even if their appearance changes. The purpose of our software is to generate different faces with different kinds of facial attributes from a given police sketch. It can also measure the similarity between the generated faces and a given suspect photo and then choose the one with the highest similarity. In this way, we aim to increase the accuracy rate of correct criminal identification using the police sketches.

Background for GANs

Generative Adversarial Networks (GANs) are one of the most popular innovations in Machine Learning [1]. GANs are famous because they can create new data instances like training data. They have been used to create amazingly realistic new human facial images, animal images, and many more. This algorithm consists of two main parts, where one is the generator which learns to create the target output according to the discriminator. The discriminator tries to distinguish between real images and fake images that are produced by the generator, which tries to fool the discriminator until convergence. We can say the network reached convergence if the discriminator cannot distinguish between real data and generated data, where accuracy of the discriminator is %50. We will be using GANs through this project extensively.

Section 2: System Design

Our project consists of five main parts, where each one of them are vital to create a smoothly working automated product. Dividing the project into different parts and completing them one by one gave us a significant advantage during the semester, so we will explain each part step by step in this section.

The big picture of our project is given in figure 1, where each step is numbered according to their work order. First, we have a dataset of face sketches and original images as pairs. Using that dataset, we trained a GAN model that can produce realistic faces for a given sketch. Using that sketch and its original attributes, we can add 8 different attributes to the generated image. Finally, we can use a similarity metric to compare the given suspect image with our 8 images with different attributes.

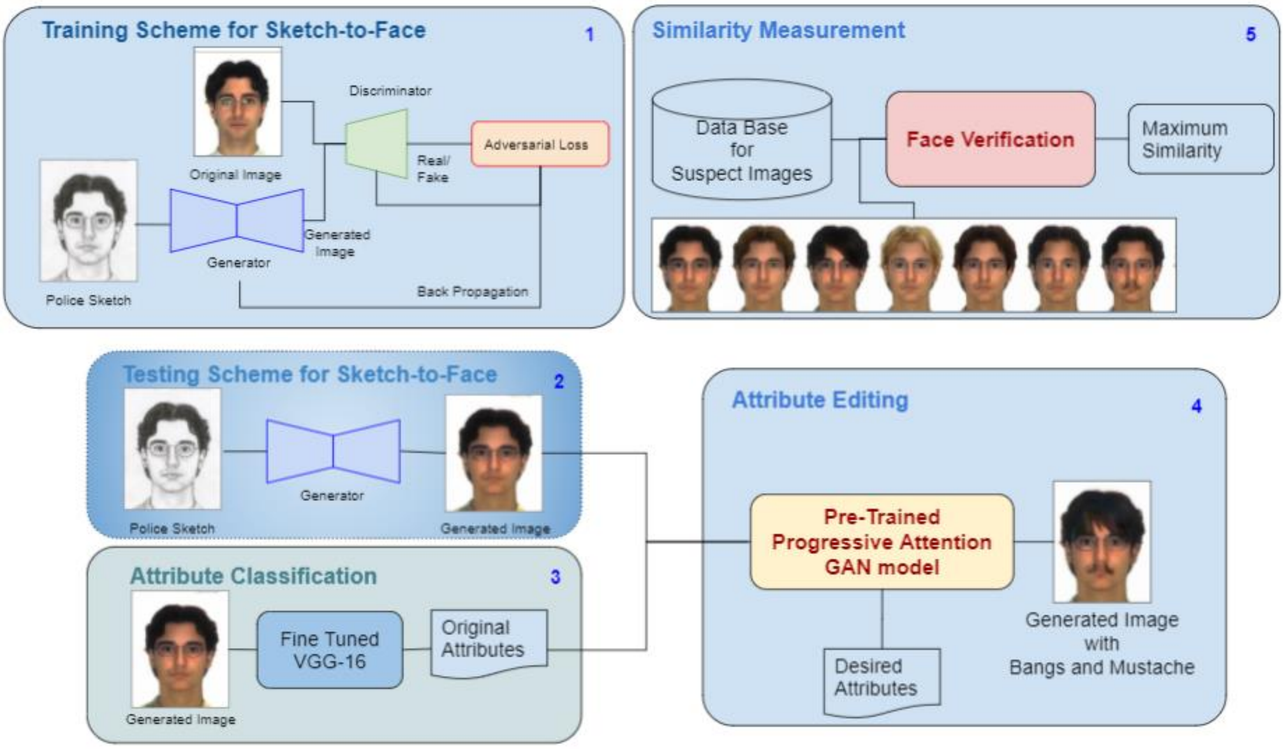

Figure 1 System diagram of the project

2.1 Sketch to Face Generation

In this part we used basic GAN structure to generate realistic looking face images from police sketches. The architecture for the training process of the model is shown in the 1st block of figure 1. As explained in the 1st section, in our case we have the original image and police sketch as the input to the GAN model. Police sketches are the inputs of the generator. Generator initially outputs not realistic looking faces, which depends on the randomly initialized weights of the autoencoder. When output from the generator is not looking like a real face, the discriminator can easily distinguish that it is a fake face and outputs large adversarial loss. As the model is trained more and more, the generator can produce more realistic looking images and it can fool the discriminator. In the 2nd block of figure 1, a testing scheme is shown, where we use only a generator to produce a realistic looking facial image. More visual and statistical results will be shown and discussed in section 3.

In all the state of art models for sketch-to-face generation, numerical and perceptual results are given for models that are trained with a different dataset individually [2]. In our experiments, we observed that models trained on a dataset gave very bad results for sketches from a distinct dataset, while they gave very good results when we used their own test sets. Different sketches have very different styles, and their photo pairs can have different lighting conditions, so a model trained with one dataset normally cannot perform well with a very different sketch. We have two models trained on different datasets. One of the datasets used from AR database [3], while the second dataset is produced by us with a face to sketch algorithm that will be explained in section 2.5.

2.2 Attribute Classification

We have used a pre-trained GAN model for editing facial attributes. This pre-trained model requires both a face image and its 40 attributes as labels, which will be used as original facial attributes [4]. Since we are generating a completely new face from a sketch, we need its original attributes. One of the ways to obtain attributes is visually looking at the image and labelling the attributes by hand. However, this might be overwhelming to do for 40 attributes. One of the main goals of the project is to make it as user friendly as possible, so we used a deep learning method to obtain attributes as shown in 3rd block of the figure 1.

Pre-trained weights of VGG-16 on facial images are used to get original attributes [4]. We applied fine tuning by changing the last two layers of that model. Basically, we deleted the output layer and the last hidden layer. After that we added one dense layer, one dropout layer and another dense layer with size 40 as the output layer. Then we used the Celeb-A dataset to train that new model [4]. Since training takes around a day, the high performance computing centre of our university is used. The results and analysis will be discussed in section 3.

2.3 Attribute Editing

The next step is to change facial attributes of generated faces as shown in the 4th block of figure 1. After doing an extensive literature review, we found out that Progressive-Attention Generative Adversarial Network (PA-GAN) is the best implementation on the internet [5]. It produces visually best-looking attribute editing results as we will show in section 3. In our project, we used a pre-trained model of PA-GAN for 256x256 images. Three inputs of that model are generated face, original attributes of that face, and desired attribute. In the Celeb-A data set, there are 40 different attributes. However, most of them were irrelevant, so we picked ones that can be used by criminals to change their appearance, which are . Results for our experiments will be discussed in the following section.

2.4 Similarity Measurement

Similarity measurements is another crucial task in this project. After generating a face from a sketch and changing its attributes, we need to compare those generated faces with possible suspects, since we cannot ask police forces to compare our generated image with given suspects by only looking at them. For example, there can be thousands of images that police forces would like to scan to see if they have the picture of the same criminal in their database. Looking at them one by one would take days, so we automated that process. We use MTCNN for face detection and Resnet50 for face recognition [6]. First, MTCNN detects the faces in each 8 generated photos and the suspect photo, and then Resnet50 produces an embedding vector for each face that represents the face characteristics of the person. Then we can easily measure the similarity between these vectors by using the well-known Cosine Similarity technique. This technique takes the face embedding vectors and finds the angle between them in multi-dimensional space. Smaller the angle, more similar the two faces are. Even though this technique is not perfect, it is good enough to claim if two faces are similar or not and to calculate the similarity percentage between them.

2.5 Face to Sketch Generation

There are two main purposes of using face-to-sketch generation in this project. Firstly, we suffered significantly during data collection for sketch-to-face implementation. The lack of not having enough data was limiting our results. On the internet, there are many sketches and faces, but they are not paired unfortunately. Therefore, we came up with this idea where we were generating our own sketches from faces using basic image processing techniques. The second reason is to make sure that anyone can experience and enjoy sketch-to-face generation and attribute editing on our website. Therefore, we have added an option that one can upload their photo and see how their sketch would look like. Then, this sketch is used to generate a new face and attributes are added to this new face.

As we mentioned in section 2.1, to generate a face from a sketch, we need a deep learning model that is trained with the same sketching style. Therefore, we trained a new model for our artificial sketches that are produced with image processing techniques. This model is used when a user uploads a photo instead of a sketch.

Section 3: Results and Discussion

In this section, we will show and discuss the results of each step described in section 2. We have implemented two GAN models to generate faces from real sketches and artificial sketches that we produced. 6 different examples including both artificial sketches and real sketches are shown in figure 2. We have used Structural Similarity Index Measure (SSIM) as performance measurement. We found that the average SSIM score for our real-sketch model is 0.63, where the state of art models reported SSIM values around 0.65-0.72 [2], and for our artificial sketch model the average SSIM score is 0.75. The main reasons for this model to have higher SSIM are the size of the training dataset due to availability of more photos, and the fact that artificial sketch-photo pairs look more alike than the real sketch-photo pairs which makes the model easier to generate. As we can see in figure 2, our results are also perceptually very good looking compared to their original pairs.

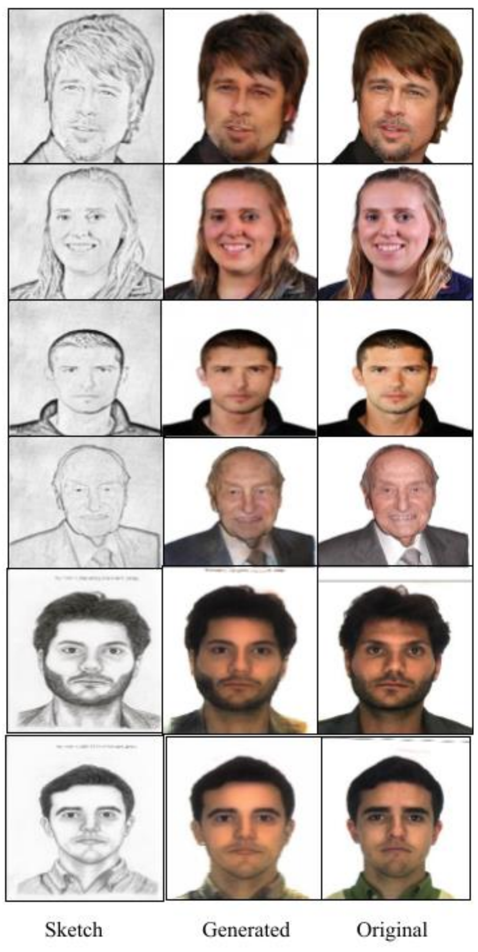

Figure 2: Sketch to Face Results

We fine-tuned VGG-16 for attribute classification in the HPC cluster of Koç University. As we mentioned in section 2.2, we only changed two layers and trained these two layers for 10 epochs with Celeb-A dataset. We obtained around 85 to 90 percent accuracy for most of the attributes as shown in the figure 3.

- 5 o'Clock Shadow: 0.90
- Arched Eyebrows: 0.75
- Attractive: 0.75
- Bags Under Eyes: 0.79
- Bald: 0.97
- Bangs: **0.84**
- Big Lips: 0.67
- Big Nose: 0.80
- Black Hair: **0.72**
- Blond Hair: **0.86**
- Blurry: 0.94
- Brown Hair: **0.82**
- Bushy Eyebrows: 0.87
- Chubby: 0.94
- Double Chin: 0.95
- Eyeglasses: **0.93**
- Goatee: 0.95
- Gray Hair: 0.96
- Heavy Makeup: 0.87
- High Cheekbones: 0.81
- Male: 0.93
- Mouth Slightly Open: 0.76
- Mustache: **0.96**
- Narrow Eyes: 0.85
- No Beard: **0.88**
- Oval Face: 0.70
- Pale Skin: 0.95
- Pointy Nose: 0.71
- Receding Hairline: 0.91
- Rosy Cheeks: 0.92
- Sideburns: 0.95
- Smiling: 0.85
- Straight Hair: 0.79
- Wavy Hair: 0.69
- Wearing Earrings: 0.79
- Wearing Hat: 0.95
- Wearing Lipstick: 0.90
- Wearing Necklace: 0.86
- Wearing Necktie: 0.92
- Young: **0.77**

Figure 3: Results for Attribute Classification

In attribute editing experiments we do not have any statistical metric to measure how the model performs. Since we are using a pre-trained model, we have shown some sample results in figure 4. Those images prove that we can use our own generated image and change its attributes.

Figure 4: Results for Attribute Editing with PA-GAN

Finally, we can use our 8 different images of the same person with different attributes and compare it with a suspect image using a face verification algorithm. One of the most common ways to apply face verification is using cosine similarity as explained in section 2.4. With cosine similarity, we were able to find how similar our generated images are compared to the given suspect face. An image of Brad Pitt is used in both figure 2 and 4. We applied cosine similarity to each face in figure 4 and figure 5, which will be taken as a suspect image. Applying cosine similarity gave us the highest similarity for the blonde attribute and suspect pair as expected visually.

Figure 5: Similarity Measurement

Section 4: Conclusion

To conclude, using police sketches for criminal identification has some serious drawbacks such as the suspects can camouflage themselves or they can change naturally as well, which makes police sketches less useful and have a low accuracy rate. Our project provides a solution to increase the accuracy rate of criminal identification by predicting possible changes in the suspects. For the time being, our project is able to generate real faces with different attributes from the sketch and returns the best-matching face amongst them with the similarity percentage given a suspect photo. Further improvements can be achieved by increasing the SSIM score of sketch-to-face generation models and the quality of the generated faces. Another improvement is to add even more attributes or change multiple attributes at the same time in order to capture more changes in suspects.

Section 5: References

[1] 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, 2014, pp. 2672–2680.

[2]] Y. Lei, W. Du, Q. Hu, Face sketch-to-photo transformation with multi-scale self-attention gan, Neurocomputing (2020)

[3] X. Wang and X. Tang, “Face Photo-Sketch Synthesis and Recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol. 31, 2009.

[4] Liu, Z., Luo, P., Wang, X., & Tang, X. (2015). Deep Learning Face Attributes in the Wild. 2015 IEEE International Conference on Computer Vision (ICCV). https://doi.org/10.1109/iccv.2015.425

[5] Zhenliang He, Meina Kan, Jichao Zhang, and Shiguang Shan. 2020. PA-GAN: Progressive Attention Generative Adversarial Network for Facial Attribute Editing. arXiv preprint arXiv:2007.05892 (2020).

[6] Yu, J., Xu, X., Gao, F., Shi, S., Wang, M., Tao, D., & Huang, Q. (2020). Toward Realistic Face Photo-Sketch Synthesis via Composition-Aided GANs. IEEE Transactions on Cybernetics, 1–13. https://doi.org/10.1109/tcyb.2020.2972944

Section 6: Appendices

Github Link - https://github.com/ridvanBalamur/CriFace_Senior_Design_Project.git
Youtube link- https://www.youtube.com/watch?v=M7MERpQ3uFI