COMPUTER ASSISTED POLICE SKETCHING USING GENERATIVE ADVERSARIAL NETWORKS

Dissertation submitted to Department of CSE,
ASETK, AMITY UNIVERSITY KOLKATA in Partial
Fulfillment for the award of

Bachelor of Technology in
Computer Science & Engineering

By

Malayanil Senapati

A91005215020

Under the Guidance of

Prof. Abhishek Das

ASETK
AMITY UNIVERSITY KOLKATA

2018-19.

Dedication

I dedicate this to my Parents, my Faculty Guide and my Team Members……

DECLARATION

I, **Mr. Malayanil Senapati**, hereby declare that the project report entitled "Computer Assisted Police Sketch using Generative Adversarial Networks" submitted to the Department of **CSE, ASETK, AMITY University Kolkata**, in partial fulfillment for the award of the Degree of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE & ENGINEERING** and that this Report has not previously formed the basis for the award of any other degree, Diploma, Associateship, Fellowship or other title.

Place: Kolkata

Date: 26.04.2019

(…………………...)

ACKNOWLEDGEMENTS

People who had helped / assisted / guides me to do the research are:

1. Prof. Abhishek Das
2. Mr. Romit Ghosh
3. Mr. Sohag Baral

List of Papers Published by the Candidate and included in this Thesis (Optional)

---NULL--

COMPUTER ASSISTED POLICE SKETCHING USING GENERATIVE ADVERSARIAL NETWORKS

A Project Synopsis submitted for Major Project for the completion of Degree of B.Tech. Program in Computer Science & Engineering under Department of Computer Science & Engineering, ASETK, AMITY UNIVERSITY KOLKATA

Under the supervision of

Prof. Abhisek Das
Department of Computer Science & Engineering, ASETK

By

Malayanil Senapati
A91005215020
Department of Computer Science & Engineering
ASETK, A.U.K.
Kolkata 765017.

C O N T E N T S

Sl.No.	INDEX	Page No.
I.	SCOPE OF WORK	
II.	PROBLEM STATEMENT	
III.	APPROACH	
IV.	EXPECTED OUTCOME (AND COMPARISION, IF ANY)	
V.	CONCLUSION AND FUTURE SCOPE	
VI.	REFERENCES	

I. SCOPE OF WORK:

In our current fast-paced world, the crime rate has gone skyrocketing and so did the ever-increasing demand of stopping or rather catching the criminal before they are on their way to another scene yet to commit another crime.

In the modern world, with the advancements in Computer and Technologies, it’s becoming easier to catch the criminal. With the progress in the field of computer vision, we are able to track and locate a person just with the help of a visual representation or image of that person's face. There are already cameras in every corner of our street keeping their eyes open round the clock. They can easily feed the police with movement data of any suspicious person with great accuracy. This input would help them out tremendously, if they are after that person, in order to keep an eye on them or to turn them in.

Now imagine that we are investigating a case. We are very close to solving the case as we got a witness, who saw the person committing the crime. Usually, we would take the witness to a sketch artist. It takes nearly an hour or so for the artist to finish the sketch. In the meantime, the criminal would take advantage of this delay and go undercover. Now we would be in a situation where we know who the criminal is but unable to catch the criminal. It would be a problem for both the police and the victim(s).

Ok, now let us assume that in another case the criminal is very brave and is careless enough to put himself out in busy streets within the public. We would have the sketch of the criminal and we would be able to feed it to the computer in order to identify it. Since what we provided to the computer was a sketch of an actual human, not only it will have a hard time comparing it to actual human images from a live feed but also have a high probability to miss the person among the crowd.

If we could have sketched the image in the first case easily, within a few minutes and make it more realisitc in the second case wouldn’t that be amazing?

This is the exact area which we are trying to pursue and solve with our program. Instead of a sketch, our program will be generating more realistic images according to the inputs given by the user. It will not only take considerably lesser time, but also generate much realistic images of the criminal, according to the criminal. It may seem negligible in terms of saving time they needed to reach this point in the investigation but in the long term, we will be multiplying by solving more cases per month. It will be a great leap for the police department in reducing the amount of time they invest behind solving cases.

II. PROBLEM STATEMENT:

According to current system and customary in the police department, it has a huge time gap between getting a witness to agree to help them in making a sketch and actually having a sketch ready in-hand to pursue the criminal. This time gap can cost a lot to both the police and the victim(s) depending on the weight and importance of the case.

Till the last decade, we didn't have computing power, powerful enough to compete human in a variety of task. But in recent decades, we got technological advancements that helped programs to learn how to do things, sometimes even better than human beings. Currently, we have records of computers beating human beings in chess, video games and even in goal keeping. With time, machine learning algorithms are becoming better and more accurate. It might not be powerful enough to take on humanity in recent future as shown in foreign movies but the progress rate is very high.

Thus, in order to solve the above problem, we can use powerful algorithms and computers to accomplish that task for us more accurately and in fraction of time what it used to take before. It will not only reduce the time spent behind every case but also blur the lines between sketches and real image of criminals. Moreover, machine learning is very much based on data, so over time the algorithms would get more data to work on and its accuracy would increase many folds!

III. APPROACH:

After finding out about the problem, the first thing that came to our mind was data, a lot of data. For us, the data implied faces of random people.

Hence, we began our hunt for gathering images of people. Since we were trying to develop the program for the Indian police force, we searched a lot for proper pictures of Indian faces. Later we came to know that it wasn’t available to the public due to regulatory rules imposed by our government on distributing others’ pictures without their consent.

Therefore, we had to settle with the datasets we found of western people which were available for research purpose on request. The one we used is the color FERET Database, hosted by the National Institute of Standard and Technology (NIST). It consisted of images of approximately 1200 unique faces from different angles. We took only the front facing pictures among the rest of the images since we don’t have as such a powerfully configured computer system to train the model.

Finally, when we were ready with our data, we went on a search for the model we will be using. Once we were done with all the searching, we configured the model, cleaned the data and fed it to the model. At first, we kept only five sliders for manipulating the image which was going to be generated by the program but later we realized that only five wasn’t giving us enough control over the image so we expanded it to ten sliders and eventually ended up with twenty sliders. We also did several tests in order to find the optimal number of sliders, epoch iterations, and dataset size.

IV. EXPECTED OUTCOME:

The outcome or output of our project is an interface screen with a window that displays a completely random human face. The face does not belong to any real person and is, instead created arbitrarily by the GAN model to mimic a real person’s face. Also present in the interface screen are twenty sliders. Each slider is responsible for altering one or more facial features of the human face displayed in the window. These facial features range from skin tone & eye color to hair length & even the color of the T-Shirt worn in the picture. The displayed face in the window can also alter its gender, depending upon the facial features present at any given moment. Each unique slider configuration on the interface will produce a completely unique face on the window. The position of each slider indicates how important it is in altering the image. This means that the first slider on the interface is responsible for the most impactful change on the face while the last slider is responsible for the least impactful change. The idea here is that a user will be able to configure these sliders in a way that the generated face will be able to closely resemble the features of a real world person. Thus, the interface can be used for face sketching of real people and can possibly be utilized for effective police sketching.

Comparing the actual outcome to the ideal outcome we find that the faces generated by out GAN model are a bit distorted. The faces are not as well defined as we would like ideally. That is because we have used far lesser data points in our model that we would have liked to due to hardware limitations on our computers. The project is not quite ready to be used a police sketching tool due to this reason. However, it does stand as a successful proof of concept. With a more capable machine and by utilizing a larger dataset we could significantly improve the quality of the generated faces.

V. CONCLUSION AND FUTURE SCOPE:

The objective of this project was to create an application using Generative Adversarial Networks that would be able to generate unique and arbitrary images of fake human faces. This would be possible using several sliders present in the application, each responsible for altering an individual (or multiple, in some cases) facial feature like Skin Tone, Facial Hair Length, Shape of the Head, etc. A user will thus be able to adjust the different sliders and create a completely random and fake human face, for every slider configuration. These images, although completely fake, can resemble actual people.

The objective of the project has been met. The application is successfully able to create faces resembling real people. The interface is straightforward to use and works as intended. The sliders also correctly alter the face shown on the application window. The extent of alteration changes from the first slider to the last, with the first causing the most notable changes to the face while the last causing the least notable change. Due to limitations in hardware, the full capability of the application is still not fully realized. Once the application is trained in capable hardware, it will be ready to be used in Police Sketching. Then this application can be directly used to create police sketches of persons of interest. These sketches will be better than hand drawn sketches since the images created using the application is almost instantaneous and, being a computer application, will be more convenient and practical to use.

VI. REFERENCES:

1. Introductory Guide to GANs: https://skymind.ai/wiki/generative-adversarial-network-gan

2. GANs in-depth: https://medium.com/ai-society/gans-from-scratch-1-a-deep-introduction-with-code-in-pytorch-and-tensorflow-cb03cdcdba0f

3. OpenCV Documentation: https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_tutorials.html

4. Numpy Documentation and Guide: https://docs.scipy.org/doc/numpy-1.16.1/

5. Theano Documentation: http://deeplearning.net/software/theano/theano.pdf

6. Keras Documentation: https://keras.io/documentation/

7. Matplotlib Documentation: https://matplotlib.org/users/index.html

8. Pydot Documentation: https://kite.com/python/docs/pydot

9. Pygame Documentation: https://www.pygame.org/docs/

10. H5py Documentation: http://docs.h5py.org/en/stable/

ABSTRACT

We, Indians, are the second most populous country in the world. Even though we are the world’s largest democracy, along with several legal bodies to control criminal activities, a lot of cases are overlooked or suspended due to lack of proper timings. Several police cases are suspended or left pending due to a variety of reasons. One such reason is lack of proper sketching artists who are available on time to convict the criminals.

We chose to create a solution to the current customary of Police Sketching with the help of computer technologies. Generally, if a witness agrees to a sketch session, it takes about an hour or so. This time constraint can be a serious loop-hole the criminal can use to escape. Moreover, a hand sketch may not be as accurate and if it is used by a computer to detect faces from a live feed via CCTVs, there is a very high probability that the computer would miss the suspect. On the other hand, if a digital image is provided for comparison, the probability of success goes higher.

We developed a Python program which is able to create random faces and provides an interface with 20 sliders which can manipulate the major and distinguishing features of a face to create a face similar to that of the criminal.

If the program is fed with proper data (Indian, Front Faces with identical background) and trained, it has the potential to help out criminal investigations. It can cut the costs of both time and the sketcher and moreover, it would be available as soon as the program is run. A proper distribution of the program in a platform independent manner would ensure a mighty impact on the Police Departments across our country and help narrowing down suspects and maybe prevent other criminal activities.

TABLE OF CONTENTS

Sl.No.	INDEX
I.	Project Summary
II.	System Requirements
III.	Code Dependencies
IV.	Introduction to Machine Learning
V.	Introduction to Generative Adversarial Networks
VI.	Describing the Dataset
VII.	Code Explanation
VIII.	Sample Outputs
IX.	Weekly Progress Report
X.	Conclusion

I. PROJECT SUMMARY

"Computer Assisted Police Sketching using Generative Adversarial Networks" - The objective of this project was to create an application using Generative Adversarial Networks that would be able to generate unique and arbitrary images of fake human faces. This would be possible through the use of several sliders present in the application, each responsible for altering an individual (or multiple, in some cases) facial feature like Skin Tone, Facial Hair Length, Shape of the Head, etc. A user will thus be able to adjust the different sliders and create a completely random human face, for every slider configuration. These images, although completely fake, can resemble actual people. As a result, this application can be directly used to create police sketches of persons of interest. These sketches are better than hand drawn sketches due to the fact that the images created using the application is almost instantaneous and being a computer application, will be more convenient and practical to use.

II. SYSTEM REQUIREMENTS

The recommended system requirements for this deep learning networks' project implementation titled as Computer Assisted Police Sketching using Generative Adversarial Networks is as follows:

Hardware:

- Processor: Intel or AMD with greater than or equal to 4 cores and clock speed of 3GHz or higher.
- RAM: greater than equal to 12GB with clock speed 2400MHz or higher.
- HDD Space: 10GB
- GPU: GTX 1050Ti or higher for Nvidia. Similar performance card for AMD.

Software:

- Anaconda with Python 3.6
- Conda environment set as path
- Spyder IDE

III. CODE DEPENDENCIES

Our codes need the following Python 3.6 libraries to function properly.

- Open CV
- Open BLAS
- Nvidia CUDA
- H5py
- Pygame
- Theano
- Keras
- Numpy
- Matplotlib

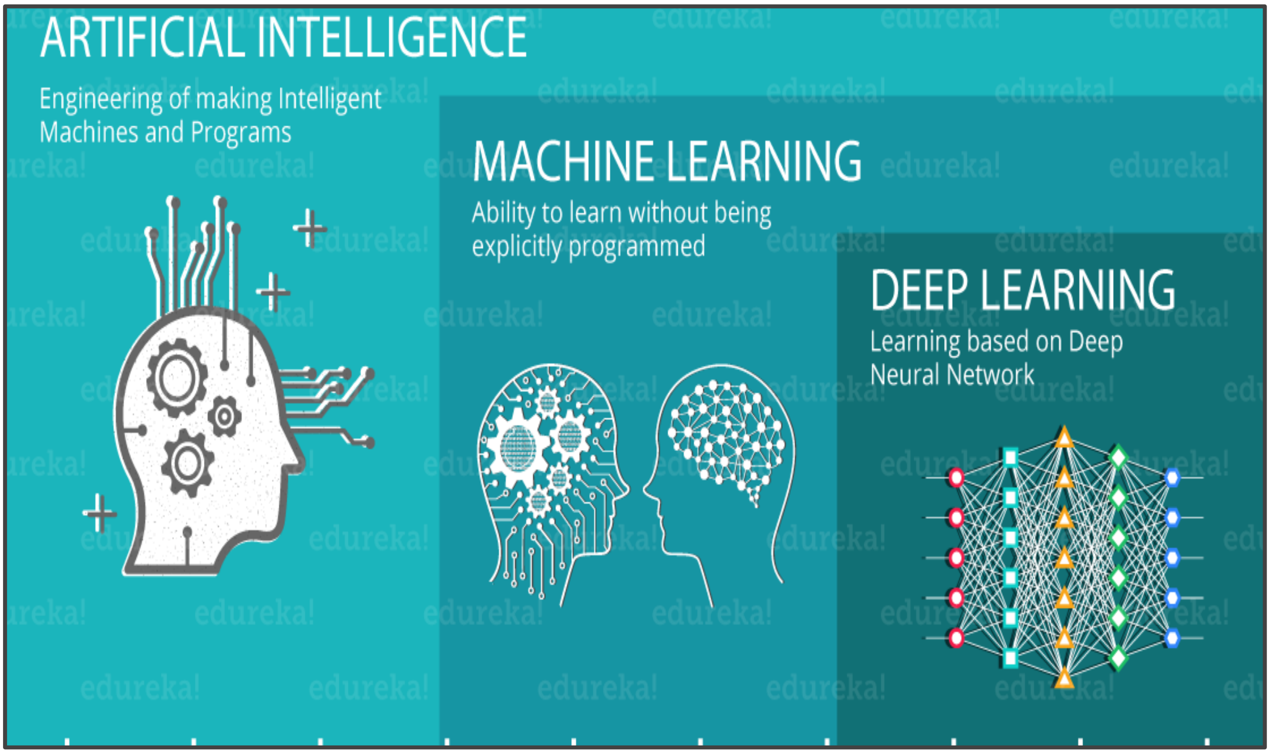
The dataset also needs to be in the same directory as the code in a folder named "PPM Files".

The dataset needs to be clean and precise as possible for better results.

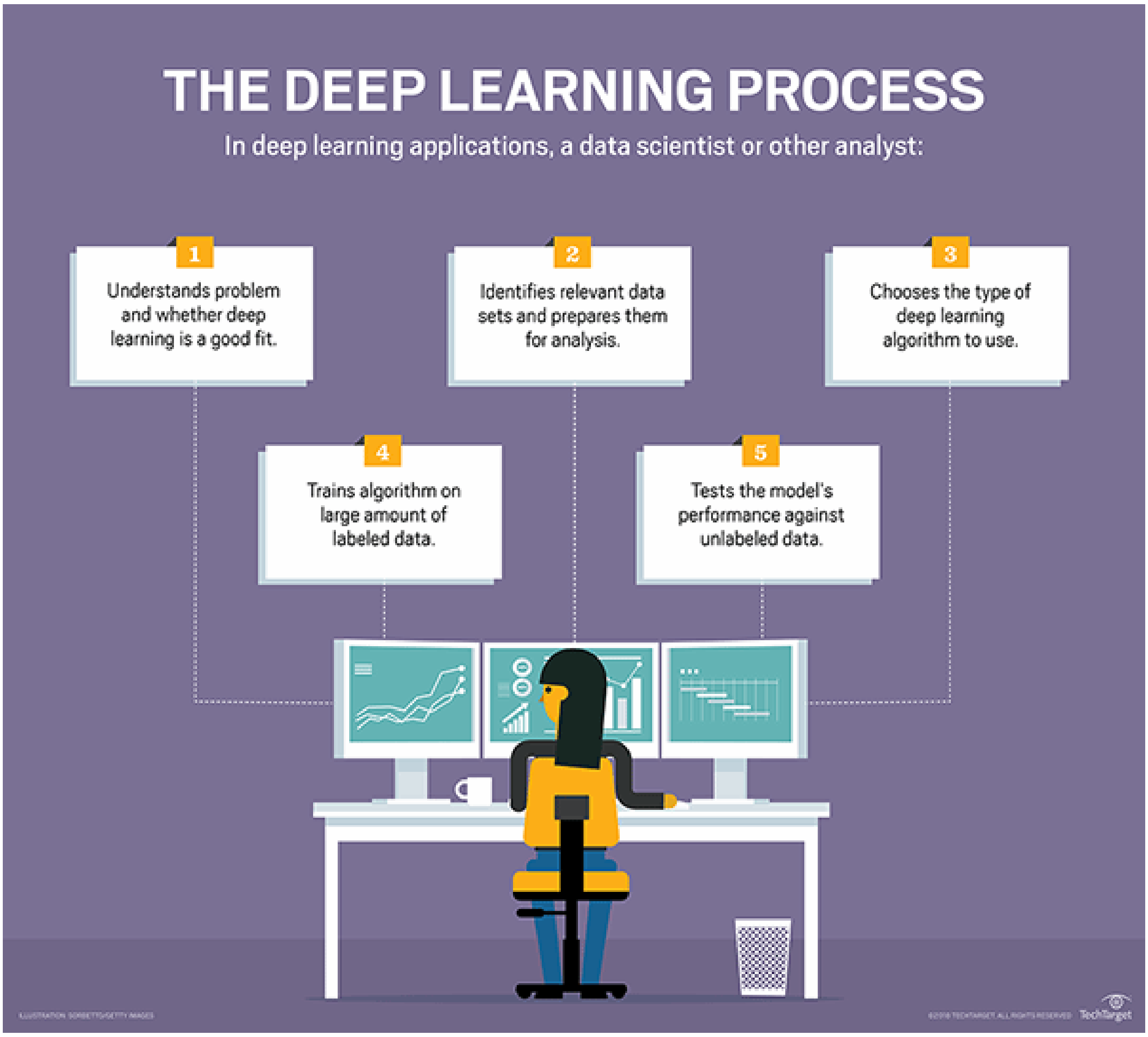
IV. INTRODUCTION TO DEEP LEARNING

Generative Adversarial Networks (GANs) fall under the Deep Learning Architecture. Before we can understand GANs, let us know more about Deep Learning.

Deep Learning is a subset of Machine Learning, concerned with emulating the learning approach used by humans. While traditional machine learning algorithms are linear, deep learning algorithms are stacked in a hierarchy of increasing complexity and abstraction. This **non-linear structure** performs transformations on the given input in order to predict an output. Multiple iterations of the above process continue until we reach an acceptable accuracy in our output. The passing of input data through a number of intermediate layers is what inspires the term *deep*, in this approach.



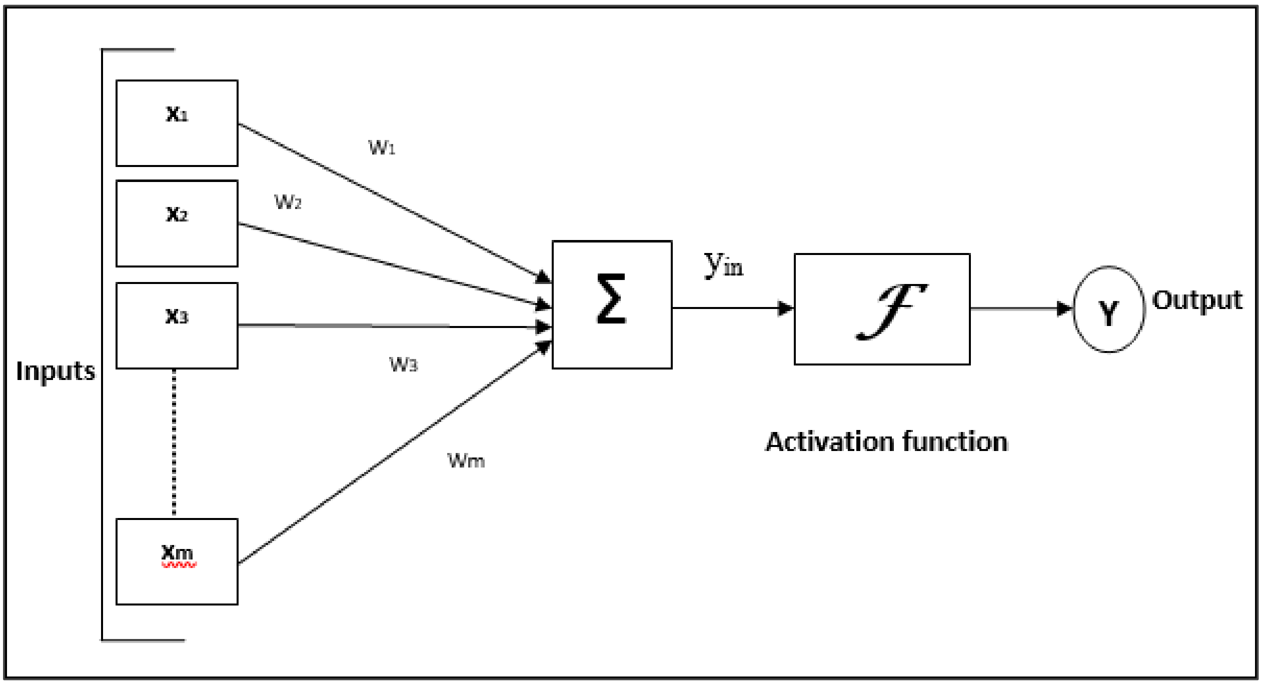
To understand the deep learning process, let us take an example where we create a model that can predict images of dogs. Using traditional machine learning, the learning process is supervised and the programmer has to be very specific when telling the model what types of things it should be looking for when deciding if an image contains a dog or does not contain a dog. This is a laborious process called **feature extraction** and the computer's success rate depends entirely upon the programmer's ability to accurately define a feature set for "dog." The advantage of deep learning is that the program builds the feature set by itself without supervision. Unsupervised learning is not only faster, but it is usually more accurate.



ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks (ANNs) are the basis of Deep Learning. They are an **information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information**. ANNs acquire a large collection of units that are interconnected in a pattern that allows communication between the units. These units, also referred to as nodes or neurons, are simple processors which operate in parallel. Every node is connected with other node through a connection link. Each connection link is associated with a weight that has information about the input signal. This is the most useful information for neurons to solve a particular problem because the weight usually excites or inhibits the signal that is being communicated. Each node has an internal state, which is called an activation signal. Output signals, which are produced after combining the input signals and activation rule, may be sent to other units.

The following diagram represents the general model of ANN followed by its processing.



For the above general model of artificial neural network, the net input can be calculated as follows −

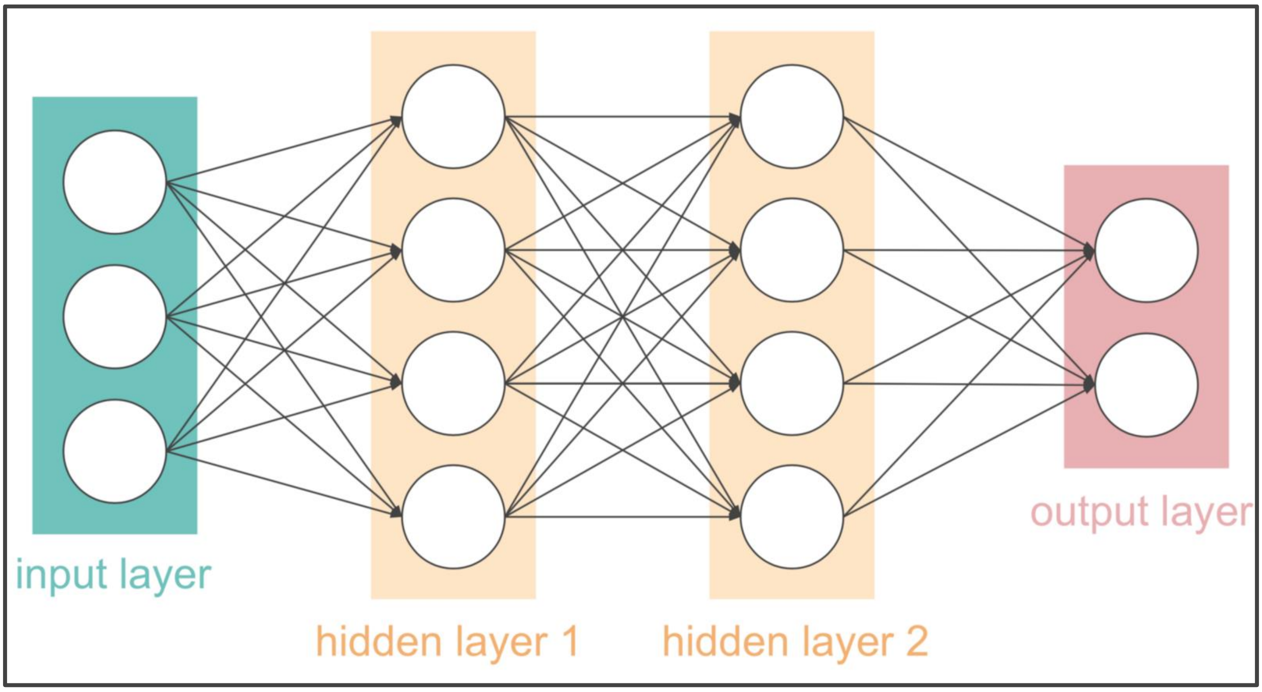
$$y_{in} = x_1 \,.\, w_1 + x_2 \,.\, w_2 + x_3 \,.\, w_3 + \dots + x_m \,.\, w_m$$

I.e, Net Input $y_{in} = \sum_i^m \; x_i \,.\, w_i$

The output can be calculated by applying the activation function over the net input.

$$Y=F(y_{in})$$
Output = Function (net input calculated)

The figure below shows the basic structure of a Neural Network. It comprises of three types of layers: the input layer, one or more hidden layers and an output layer. Each layer has multiple nodes, which represents the actual data for each layer. The hidden layer contains the modifications made to the input data by the neural network required for predicting the output.

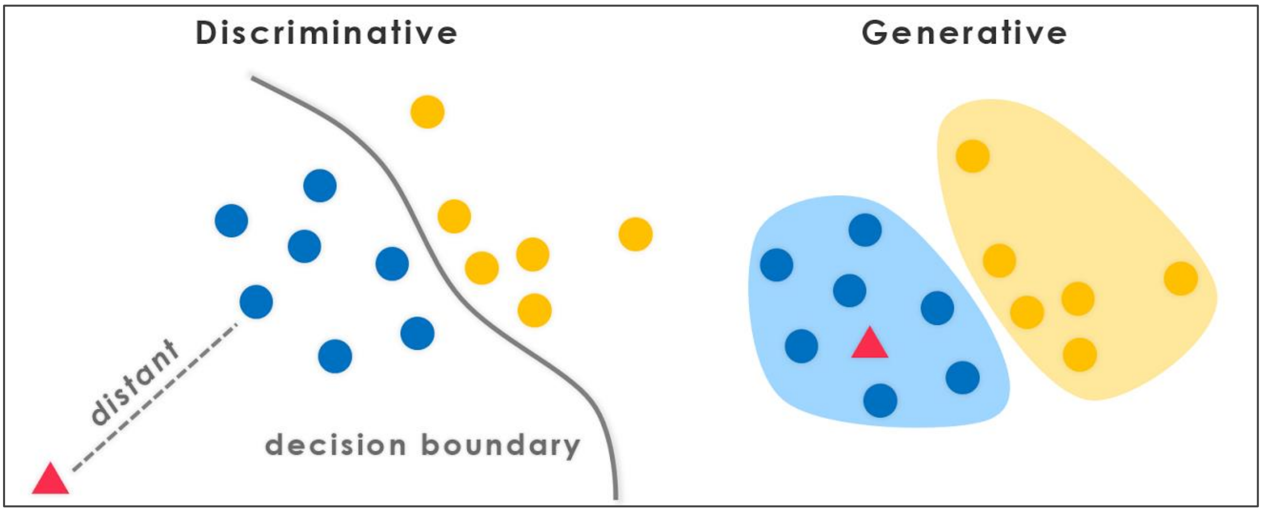


V. INTRODUCTION TO GENERATIVE ADVERSARIAL NETWORKS

Generative adversarial networks (GANs) are deep neural network architectures that are comprised of two networks, pitted one against one other, and are hence called adversaries. The main focus for GAN (Generative Adversarial Networks) is to generate data from scratch. GANs' potential is huge, because they can learn to mimic any distribution of data. That is, GANs can be taught to create worlds very similar to our own in any domain: images, music, speech, prose. The image below is an illustration of the capability of GANs. It is taken from the cycleGAN project, and shows how a GAN converts a picture taken in summer into a completely fake but real-looking picture of the same place in winter.

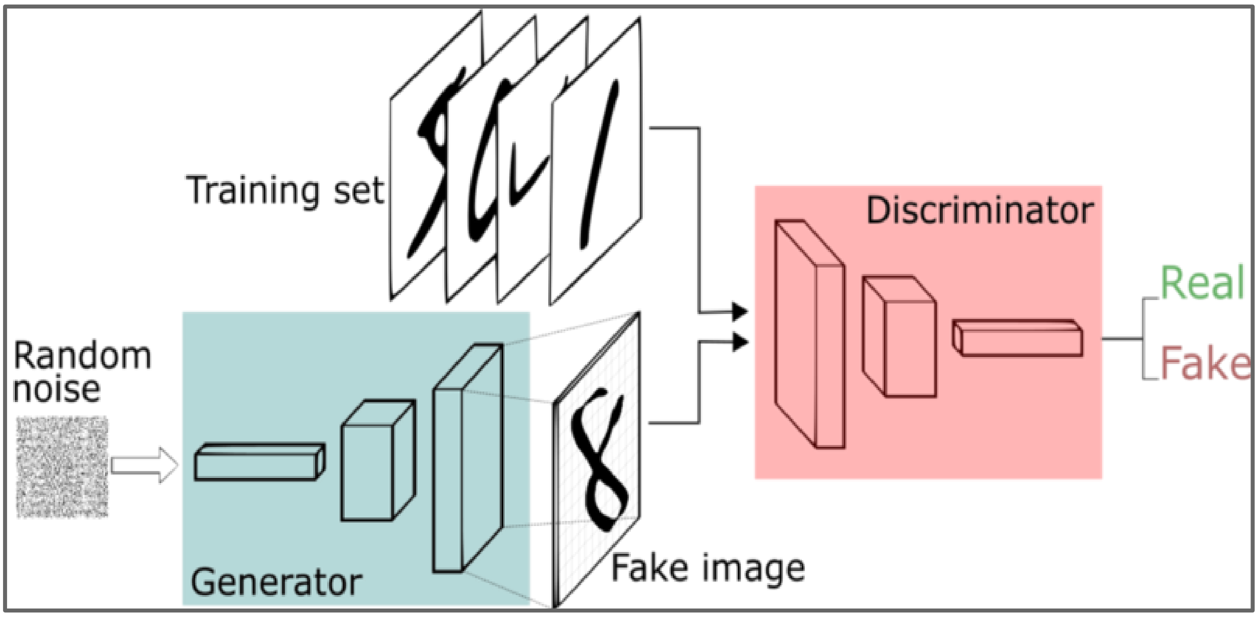
GAN composes of two deep networks, the **generator,** and the **discriminator**. Before finding out how they work in tandem inside GANs, let us define them individually. Discriminators try to classify input data; that is, given the features of an instance of data, they predict a label or category to which that data belongs. Discriminative algorithms map features to labels and are concerned solely with that correlation.

Generative algorithms, in contrast, do the opposite. Instead of predicting a label given certain features, they attempt to predict features given a certain label. Another way to look at them is that discriminative models learn the boundaries between classes of a dataset while generative models learn to model the distribution of individual classes.



EXPLANATION OF HOW GANs WORK

In GAN, one neural network (generator) generates new data instances, while the other, the discriminator, evaluates them for authenticity; i.e. the discriminator decides whether each instance of data that it reviews belongs to the actual training dataset or not. For our project we are concerned with creating real-looking images of human faces. The generator is used to generate real-looking images and the discriminator's job is to identify fake images. The two adversaries are in constant battle as one (generator) tries to fool the other (discriminator), while the other tries not to be fooled. The input, random noise can be a Gaussian distribution and values can be sampled from this distribution and fed into the generator network and an image is generated. Conceptually, this noise represents latent features of the dataset, like image color, shape, etc.



Thus, in order to create fake images that are similar looking to a certain dataset, the steps followed by a GAN are:

- The generator takes in random noise and returns an image.
- This generated image is fed into the discriminator alongside a stream of images taken from the actual, ground-truth dataset.
- The discriminator takes in both real and fake images and returns probabilities, a number between 0 and 1, with 1 representing a prediction of authenticity and 0 representing fake.
- The discriminator is in a feedback loop with the ground truth of the images, obtained from the dataset labels.
- The generator is in a feedback loop with the discriminator.

VI. DESCRIBING THE DATASET

The DOD Counterdrug Technology Program sponsored the Facial Recognition Technology (FERET) program and development of the FERET database. The National Institute of Standards and Technology (NIST) is serving as Technical Agent for distribution of the FERET database. The goal of the FERET program is to develop new techniques, technology, and algorithms for the automatic recognition of human faces. As part of the FERET program, a database of facial imagery was collected between December 1993 and August 1996. The database is used to develop, test, and evaluate face recognition algorithms.

The FERET program set out to establish a large database of facial images that was gathered independently from the algorithm developers. The database collection was a collaborative effort between Dr. Wechsler and Dr. Phillips. The images were collected in a semi-controlled environment. To maintain a degree of consistency throughout the database, the same physical setup was used in each photography session. Because the equipment had to be reassembled for each session, there was some minor variation in images collected on different dates. The FERET database was collected in 15 sessions between August 1993 and July 1996. The database contains 1564 sets of images for a total of 14,126 images that includes 1199 individuals and 365 duplicate sets of images. A duplicate set is a second set of images of a person already in the database and was usually taken on a different day. For some individuals, over two years had elapsed between their first and last sittings, with some subjects being photographed multiple times. This time lapse was important because it enabled e-searchers to study, for the first time, changes in a subject's appearance that occur over a year.

Some samples from the Dataset:

How we obtained the Dataset:

To request an account that will allow you to download the Color FERET database:
1. We had to read the Release Agreement.

2. Then we sent a plain-text email of the form below:
To: colorferet@nist.gov
Subject: Color FERET download request
Body: Please create an account that will allow me to download the Color FERET database. I will abide by the Release Agreement version 1.

3. They sent back an email, detailing how to use the account.
The processing of email requests is mostly-automated, so we didn't significantly change the text in the body of the email. Anything beyond the plain-text format above might have risked the request to not being processable, or of being intercepted by malware, spam or profanity filters. Email requests are eyeballed to prevent the system from being misused to relay spam. Therefore, the reply email was not sent out in real-time, but arrived within two business days. Accounts were deleted after a week, so we began downloading as soon as we were notified that our account had been created. It is a 7.99GB of images along with other meta-data.

VII. CODE EXPLANATION

At first, we cleanse the data and make it suitable for our model to ingest and start extracting the features. We save the data generated in NumPy matrices and keep it for later use.

We then proceed to build our model. We chose the Sequential model of Keras for our project. For training, we use the data we saved and iterate through several epochs with a low learning rate. We save our model with the help of h5py encoder library for later use. It takes a lot of time for training, we spent 72 hours training our model and saved it at specific intervals in case of crash recovery.

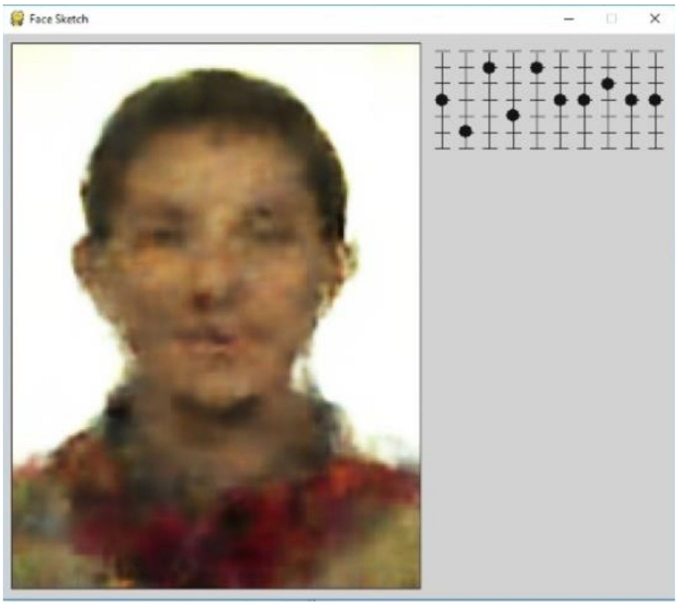
We then developed a GUI Window for the user to be able to interact with the system and generate faces. We provided sliders to change the facial features and adjust them for refining the image generated. The saved encoder is loaded here, which helps in changing the generated faces to change according to the changes made in the sliders.

VIII. SAMPLES

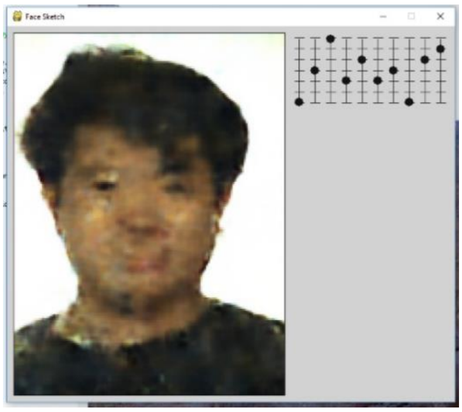
Iteration 1:

In the first phase of development, we kept only 10 sliders in order to change the attributes of pictures.

Output 1:



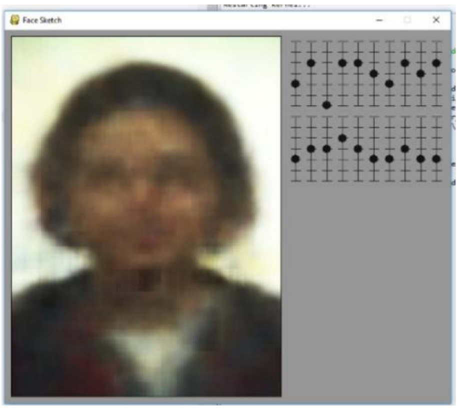
Output 2:



Iteration 2:

In the final phase of development, we increased the number of sliders from 10 to 20 so that we can have more refined and specific control over the changes that were taking place.

Output 3:



Output 4:

IX. WEEKLY PROGRESS REPORT

Week 1: The first week was spent on researching different project topics and finding out a suitable project for our group that would be both challenging and practical for us to do.

Week 2: In the second week, we researched on our chosen topic- Generative Adversarial Networks (GANs) in order to thoroughly understand it. We also spent time planning how to carry out the project and the designated roles of the group members.

Week 3: In this week, we studied different implementations of GAN and determined which implementation was the most suitable for our project.

Week 4: In this week, we considered the different public datasets that were available to us and decided which one to use. We also fetched the dataset and carried out the necessary preprocessing that was needed for our project.

Week 5: This week was spent on coding the GAN model. Any complications that arose during the coding process was systematically solved.

Week 6: This week was also spent on coding and following up on previous week's activities.

Week 7: In the penultimate week, we carried out test runs on our GAN model. We debugged the model and modified it wherever needed.

Week 8: We finalized the changes in the number of sliders and also made a few changes to the intervals our encoder was saved to recover from system crashes. We also stabilized our program and approached for an executable format for portability and reduction of dependencies.

X. CONCLUSION

The objective of this project was to create an application using Generative Adversarial Networks that would be able to generate unique and arbitrary images of fake human faces. We have made this possible by creating an interface screen with a window that displays a completely random human face. The face does not belong to any real person and is, instead created arbitrarily by the GAN model to mimic a real person's face. Also present in the interface screen are twenty sliders. Each slider is responsible for altering one or more facial features of the human face displayed in the window. These facial features range from skin tone & eye color to hair length & even the color of the T-Shirt worn in the picture. The displayed face in the window can also alter its gender, depending upon the facial features present at any given moment. Each unique slider configuration on the interface will produce a completely unique face on the window. The position of each slider indicates how important it is in altering the image. This means that the first slider on the interface is responsible for the most impactful change on the face while the last slider is responsible for the least impactful change. The idea here is that a user will be able to configure these sliders in a way that the generated face will be able to closely resemble the features of a real-world person.

Due to limitations in hardware, the full capability of the application is still not fully realized. Only a small number of images were used to train the model and thus the images of the generated face turn out to be a bit distorted. However, it does stand as a successful proof of concept. With a more capable machine and by utilizing a larger dataset we could significantly improve the quality of the generated faces. And once the application reaches its full potential, it will be ready to be used in Police Sketching. The application will be able to directly create face sketches of persons of interest. Using such an application for police sketching is a far better method than traditional face sketching due to the application's ease of use, portability and efficiency.

VITA

Amity University, Kolkata is a private university in Kolkata in the state of West Bengal, India. It was founded in 2015 and is the eighth university to be established by the Amity Education Group. The Amity Group's presence in Kolkata began at the end of 2009 as Amity Global Business School (Salt Lake, Sector V), which offered management degrees under Amity University. In 2015, the state legislature of West Bengal passed the *Amity University Act, 2014* (West Bengal Act, XXIV of 2014) to enable the establishment of the university in Rajarhat. The university was established on the 6th of January and classes commenced on 31 August 2015. It is the eighth university to be established by the Amity Education Group.

We are the first graduating engineering batch from Amity University Kolkata and we have carried out our project work partially here and at our individual homes.