Algorithms For Massive Data Project

FACE/COMIC RECOGNIZER – COMIC FACES

Mansee Agrawal 983635

Declaration

“I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study”

Table of Contents

1. Introduction

In this project I am dealing with the comic or real face problem analysis stage of data AI development life cycle of the AMD project. In this project I am going to discuss about the problem identified in the problem statement, business problem statement, data collection, data preprocessing, train and test data splitting, model building, and deployment solution for the problem identified, proposed system, resource required.

For comic or real face Classification and identification large datasets and domain-specific features are used to best fit the data. In this project, I implemented train, and test state-of-the-art algorithms trained on domain general datasets for the task of identifying comic or real face.

2. Project Designing

Project design is an early phase of the project where a project's key features, structure, criteria for success, and major deliverables are all planned out. The point is to develop one or more designs which can be used to achieve the desired project goals.

2.1 Business Understanding

The entire cycle revolves around the business goal. What will you solve if you do not have a precise problem? It is extremely important to understand the business objective clearly because that will be your final goal of the analysis. After proper understanding only a person can set the specific goal of analysis that is in sync with the business objective. You need to know if the client wants to reduce credit loss, or if they want to predict the price of a commodity, etc.

2.2 Data Understanding

After business understanding, the next step is data understanding. This involves the collection of all the available data. Here you need to closely work with the business team as they are actually aware of what data is present, what data could be used for this business problem and other information. This step involves describing the data, their structure, their relevance, their data type. Explore the data using graphical plots. Basically, extracting any information that you can get about the data by just exploring the data.

2.3 Data Preparation

Next comes the data preparation stage. This includes steps like selecting the relevant data, integrating the data by merging the data sets, cleaning it, treating the missing values by either removing them or imputing them, treating erroneous data by removing them, also check for outliers using box plots and handle them. Constructing new data, derive new features from existing ones. Format the data into the desired structure, remove unwanted columns and features. Data preparation is the most time consuming yet arguably the most important step in the entire life cycle. Your model will be as good as your data.

2.4 Exploratory Data Analysis

This step involves getting some idea about the solution and factors affecting it, before building the actual model. Distribution of data within different variables of a feature is explored graphically using bar-graphs, Relations between different features is captured through graphical representations like scatter plots and heat maps. Many other data visualization techniques are extensively used to explore every feature individually, and by combining them with other features

2.5 Data Modeling

Data modeling is the heart of data science. A model takes the prepared data as input and provides the desired output. This step includes choosing the appropriate type of model, whether the problem is a classification 18 problem, or a regression problem or a clustering problem. After choosing the model family, amongst the various algorithm amongst that family, a person need to carefully choose the algorithms to implement and implement them. A person need to tune the hyperparameters of each model to achieve the desired performance. A person also need to make sure there is a correct balance between performance and generalizability. A person do not want the model to learn the data and perform poorly on new data.

2.6 Model Evaluation

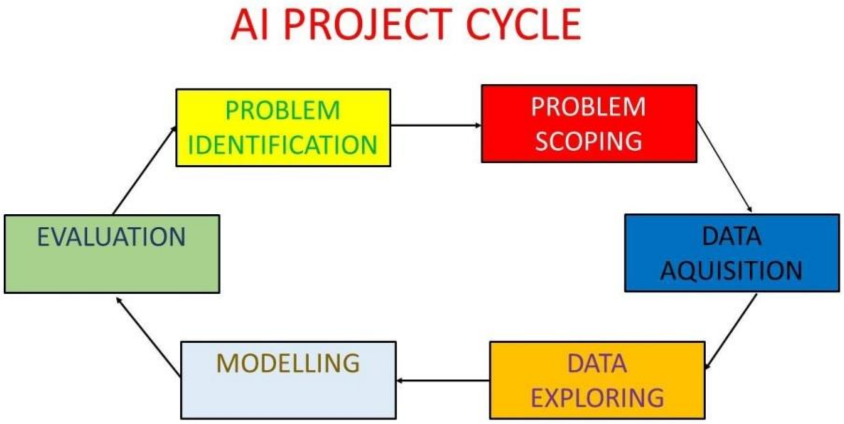
Here the model is evaluated for checking if it is ready to be deployed. The model is tested on an unseen data, evaluated on a carefully thought out set of evaluation metrics. A person also need to make sure that the model conforms to reality. If a person do not obtain a satisfactory result in the evaluation, a person must re-iterate the entire modeling process until the desired level of metrics is achieved. Any data science solution, a machine learning model, just like a human, should evolve, should be able to improve itself with new data, adapt to a new evaluation metric. A person can build multiple models for a certain phenomenon, but a lot of them may be imperfect. Model evaluation helps us choose and build a perfect model.

3. Lifecycle of AI project

Generally, the AI project consists of three main stages:

- Stage 1 – Project Planning and data collection
- Stage 2 – Design and training of the ML model.

- Stage 3 –Maintenance



Stage 1 – Project Planning and Data Collection

This is an initial stage, although very important and crucial as it explores the reasons why you decided to implement artificial intelligence solutions in your operations, as well as anticipating if the solution can be tangible and profitable.

You should think, analyze and evaluate if your problem can be solved with simpler solutions (like simple automation) or if it really requires more complex resources like Artificial Intelligence. In this project problem was to recognize if image is of real face or comic face.

Now after identifying problem you can focus on the solution. There are many solutions in the market that have worked for different businesses, but it is important to understand that your use case is unique and therefore requires a solution that is tailored to the needs of your business. So for solving problem in this project case I used Machine learning algorithm which is Deep learning Convolutional Neural Network (CNN).

From self driving cars to facial emotion recognition, data is at the core of all AI projects. Artificial Intelligence systems are capable of recognizing patterns and making decisions thanks to statistical models; for this to happen, data is required for the system to learn the correct patterns. However, when creating effective solutions, the challenge is not simply the availability of data, but rather the availability of large amounts of varied and high-quality data.Poor quality data not only prolongs projects, but also leads to more costly results, such as Machine Learning models not working properly and not delivering the desired results.Therefore, it is important that the data for your solution is not only of high quality but also relevant to the problem you are facing. My data for face recognition is of high quality because all the images are in HD quality. So there are 2 sources to collect data:

- ✓ Primary sources – Primary sources are sources that provide data that originates from your own company. You can acquire such data from tools like CRM or IoT devices.
- ✓ Secondary sources – Secondary sources refer to external sources that have relevant data of interest to you. Those can be Third-Party data providers or government publications.

Stage 2 – Design and training of the ML model.

Choosing the right ML model depends on a number of factors, such as the type of challenge your business is facing, the type of result you want to achieve, the size of your data, etc. Some of the types of ML models are:

- Binary classification model – As the name implies, ML models for binary classification problems predict a binary outcome. For example:
  Is this email spam or not (yes or no)?
  Is this app review written by a real person or not?
- Multiclass classification model – In this case, models predict an outcome into one of three or more classes. For example:
  Is a child holding a toy, a book, or a pen?
  Is this genre of music rock, jazz or hip hop?
- Regression model – Machine Learning models for regression problems predict the relationship between a single dependent variable and one or more independent variables. . For example:

  - ✓ **What will energy use be in California tomorrow?**
  - ✓ **How much product A will be sold this month?**

Training your model: In this important step, we will feed our data into our ML algorithms. This process gradually improves the ability of the models to produce the desired result (identify the object, predict the outcome, etc).

Like everything in life, practice makes better. This means that during the initial training process your model's output will have low accuracy, however this is normal and there is nothing to worry about, as with more and more training processes your model will improve. After training your model, as well as evaluating and testing its performance, it is time to move on to the next final stage.

Stage 3 – Maintenance

Finally, the last but not least stage, However, just because you've launched your AI solution live doesn't mean the project is done. As in the previous steps, an equally important part is monitoring, reviewing, and making sure that your solution continues to deliver the desired results.

4. Dataset

4.1 Data Download

The first step is to obtain the dataset for **Comic or Real face**. The direct download option through the command line is available on Kaggle's website. The command Kaggle datasets download requires the precise name of the item to be downloaded and the person's login credentials.

4.2 Data description

The dataset **Comic Faces** originates from a publicly accessible source, specifically Kaggle. This is a paired face to comic's dataset, which can be used to train pix2pix or similar networks.This dataset contains a lot of crappy nightmare-fuelish samples, which tend to be useful for full image to comic conversion, as the aim is to teach models to recognize whether it is a comic face or real face.

The dataset has total **20,000 images** which are either comic face or real face.
Sample of comic and real faces are shown below:

4.3 Data organization

The first step is to obtain the dataset Comic Faces. The direct download option through the command line is available on Kaggle's website. The command Kaggle datasets download requires the precise name of the item to be downloaded and the person's login credentials. Downloading Comic Faces Dataset from Kaggle. After we downloaded we have to manage our dataset with respective train and test folders for model training and Data organization with Default amount of 80% and 20%.

5. Kaggle API

The Kaggle API and CLI tool **provide easy ways to interact with Notebooks on Kaggle**. The commands available enable both searching for and downloading published Notebooks and their metadata as well as workflows for creating and running Notebooks using computational resources on Kaggle. In this project I used Kaggle for collecting data.

6. Data pre-processing

Data preprocessing, a component of data preparation, describes any type of processing performed on raw data to prepare it for another data processing procedure. It has traditionally been an important preliminary step for the data mining process. More recently, data preprocessing techniques have been adapted for training machine learning models and AI models and for running inferences against them.Data preprocessing transforms the data into a format that is more easily and effectively processed in data mining, machine learning and other data science tasks. The techniques are generally used at the earliest stages of the machine learning and AI development pipeline to ensure accurate results.
There are several Different tools and methods used for preprocessing data, including the following:

- Sampling, which selects a representative subset from a large population of data;
- Transformation, which manipulates raw data to produce a single input;
- DE noising, which removes noise from data;
- Imputation, which synthesizes statistically relevant data for missing values;
- Normalization, which organizes data for more efficient access; and
- Feature extraction, which pulls out a relevant feature subset that is significant in a particular context.

In this project I didn't do data cleaning and reduction because my data was already in a very good quality and data is relevant to ML tasks. Now for Data validation, the data is split into two sets. The first set is used to train a deep learning model. The second set is the testing data that is used to gauge the accuracy and robustness of the resulting model. This second step helps identify any problems in the hypothesis used in the cleaning and feature engineering of the data. Data is divided in 80% for training model and 20% for testing model.There are 20,000 images in data (10k comic faces and 10k real faces) from which 8,000 images of comic and real faces used for training and 2000 images for testing.

7. Transfer Learning

Transfer learning is a machine learning method where we reuse a pre-trained model as the starting point for a model on a new task.To put it simply—a model trained on one task is repurposed on a second, related task as an optimization that allows rapid progress when modeling the second task. By applying transfer learning to a new task, anyone can achieve significantly higher performance than training with only a small amount of data.
There are so many transfer learning models that is used for different classifications but most popular and commonly used models are following:

- VGG 16
- Inception V3
- Xception
- ResNet 50

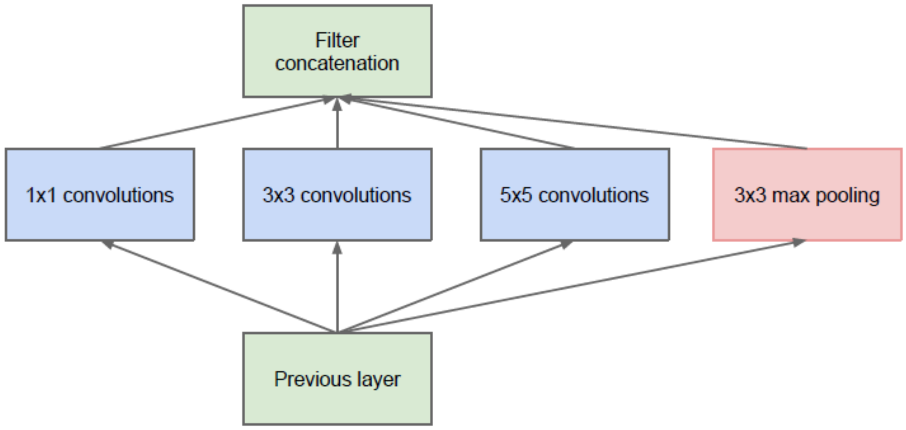
Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Speed
VGG 16	528 MB	0.715	0.901	138 million	Low
Inception V3	92 MB	0.782	0.937	23.62 million	High
Xception	91 MB	0.790	0.945	22.85 million	High
ResNet 50	100MB	0.770	0.933	23 million	High

Note: In this project I used inception V3 transfer learning model.

InceptionV3:

Inception V3is a Convolutional Neural Network for assisting in image analysis and object detection, and got its start as a module for GoogleNet. It is the third edition of Google's Inception Convolutional Neural Network, originally introduced during the ImageNet Recognition Challenge. The design of Inceptionv3 was intended to allow deeper networks while also keeping the number of parameters from growing too large: it has "under 25 million parameters", compared against 60 million for AlexNet.

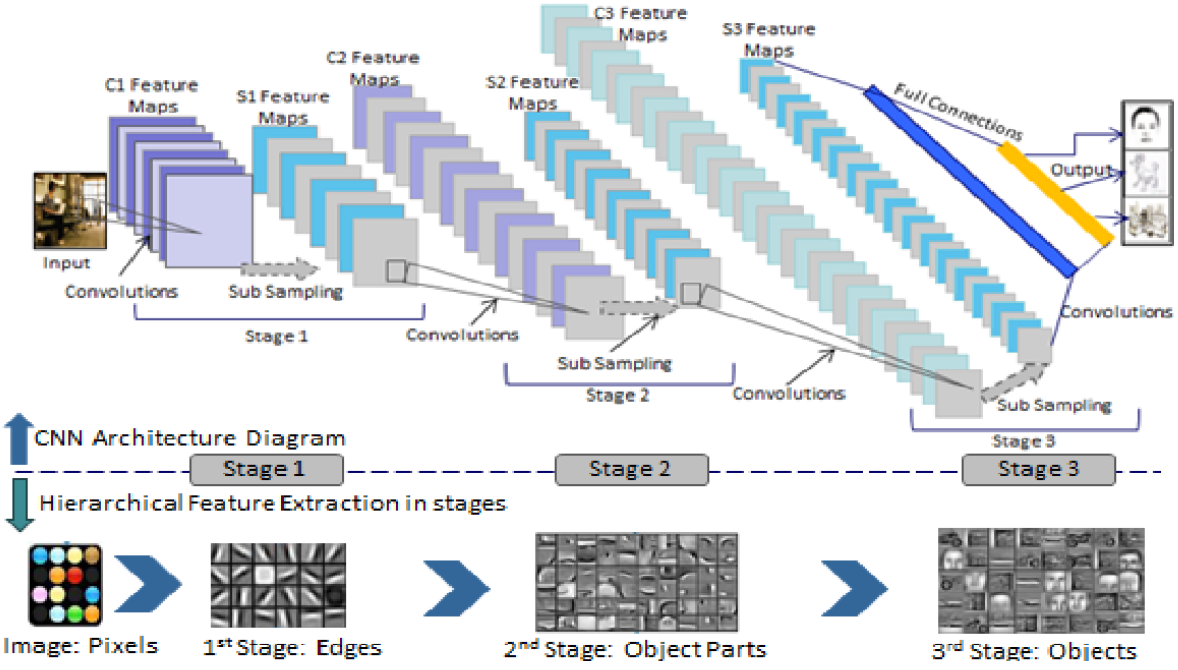
This model I used for training first then after training I got trained model using inception v3 then I used that trained model for testing to get better and efficient results. Moreover it works faster and works for small datasets efficiently as well.



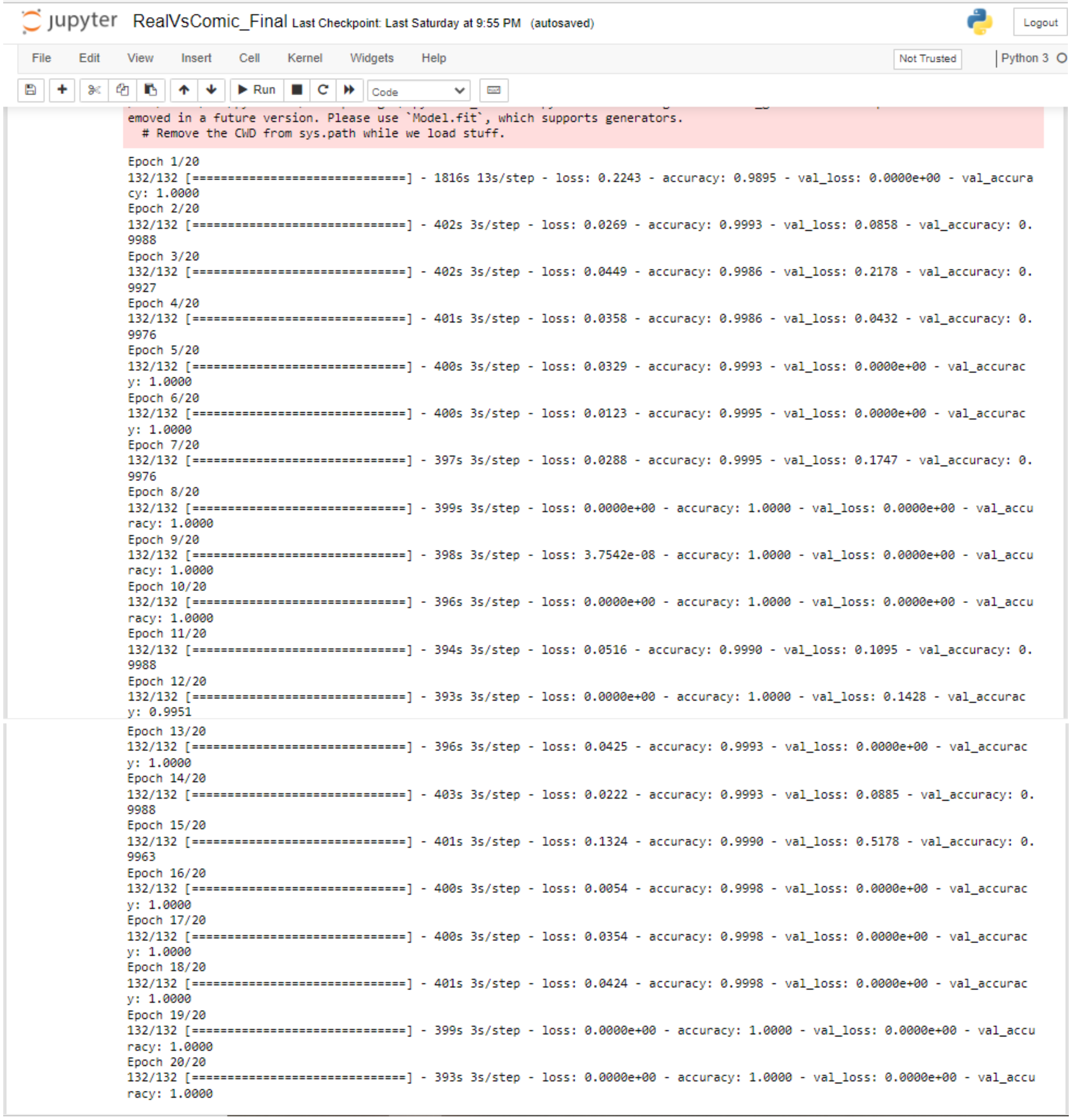
8. Model training

Model training is the phase where we try to fit the best combination of weights and bias to a machine learning algorithm to minimize a loss function over the prediction range.

This is the stage where the ML algorithm is trained by feeding datasets. This is the stage where the learning takes place. Consistent training can significantly improve the prediction rate of the ML model. The weights of the model must be initialized randomly. This way the algorithm will learn to adjust the weights accordingly.



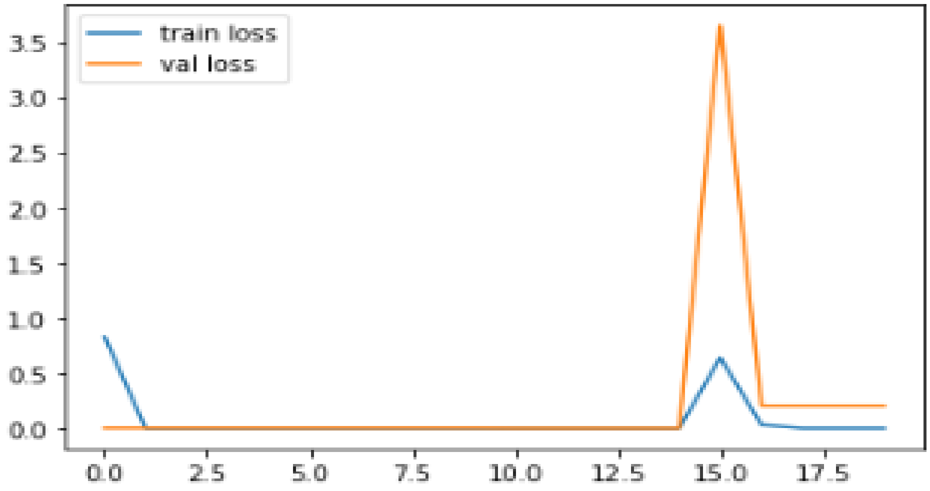
For training model I used 20 epochs if you see below in first epoch training loss is 0.2243 and training accuracy is 98% and for validation, loss is 0 and accuracy is 98%. In the starting epoch itself results are good as I reached up to 20th epoch for training, loss is 0 and accuracy is 100% same goes for validation. At first I tried using more epochs but I got very good results in less epochs so I just used 20 epochs for time saving and convenience.



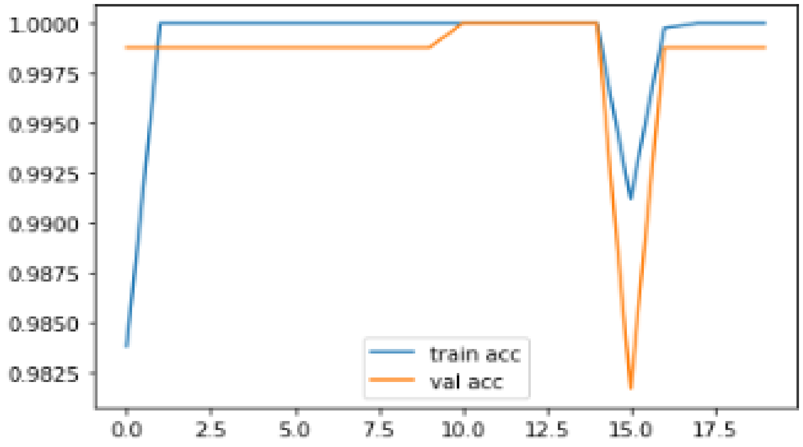
```
emoved in a future version. Please use `Model.fit`, which supports generators.
  # Remove the CWD from sys.path while we load stuff.

Epoch 1/20
132/132 [==============================] - 1816s 13s/step - loss: 0.2243 - accuracy: 0.9895 - val_loss: 0.0000e+00 - val_accura
cy: 1.0000
Epoch 2/20
132/132 [==============================] - 402s 3s/step - loss: 0.0269 - accuracy: 0.9993 - val_loss: 0.0858 - val_accuracy: 0.
9988
Epoch 3/20
132/132 [==============================] - 402s 3s/step - loss: 0.0449 - accuracy: 0.9986 - val_loss: 0.2178 - val_accuracy: 0.
9927
Epoch 4/20
132/132 [==============================] - 401s 3s/step - loss: 0.0358 - accuracy: 0.9986 - val_loss: 0.0432 - val_accuracy: 0.
9976
Epoch 5/20
132/132 [==============================] - 400s 3s/step - loss: 0.0329 - accuracy: 0.9993 - val_loss: 0.0000e+00 - val_accurac
y: 1.0000
Epoch 6/20
132/132 [==============================] - 400s 3s/step - loss: 0.0123 - accuracy: 0.9995 - val_loss: 0.0000e+00 - val_accurac
y: 1.0000
Epoch 7/20
132/132 [==============================] - 397s 3s/step - loss: 0.0288 - accuracy: 0.9995 - val_loss: 0.1747 - val_accuracy: 0.
9976
Epoch 8/20
132/132 [==============================] - 399s 3s/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accu
racy: 1.0000
Epoch 9/20
132/132 [==============================] - 398s 3s/step - loss: 3.7542e-08 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accu
racy: 1.0000
Epoch 10/20
132/132 [==============================] - 396s 3s/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accu
racy: 1.0000
Epoch 11/20
132/132 [==============================] - 394s 3s/step - loss: 0.0516 - accuracy: 0.9990 - val_loss: 0.1095 - val_accuracy: 0.
9988
Epoch 12/20
132/132 [==============================] - 393s 3s/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.1428 - val_accurac
y: 0.9951
Epoch 13/20
132/132 [==============================] - 396s 3s/step - loss: 0.0425 - accuracy: 0.9993 - val_loss: 0.0000e+00 - val_accurac
y: 1.0000
Epoch 14/20
132/132 [==============================] - 403s 3s/step - loss: 0.0222 - accuracy: 0.9993 - val_loss: 0.0885 - val_accuracy: 0.
9988
Epoch 15/20
132/132 [==============================] - 401s 3s/step - loss: 0.1324 - accuracy: 0.9990 - val_loss: 0.5178 - val_accuracy: 0.
9963
Epoch 16/20
132/132 [==============================] - 400s 3s/step - loss: 0.0054 - accuracy: 0.9998 - val_loss: 0.0000e+00 - val_accurac
y: 1.0000
Epoch 17/20
132/132 [==============================] - 400s 3s/step - loss: 0.0354 - accuracy: 0.9998 - val_loss: 0.0000e+00 - val_accurac
y: 1.0000
Epoch 18/20
132/132 [==============================] - 401s 3s/step - loss: 0.0424 - accuracy: 0.9998 - val_loss: 0.0000e+00 - val_accurac
y: 1.0000
Epoch 19/20
132/132 [==============================] - 399s 3s/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accu
racy: 1.0000
Epoch 20/20
132/132 [==============================] - 393s 3s/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accu
racy: 1.0000
```

9. Data Visualization

Data visualization is **the practice of translating information into a visual context, such as a map or graph, to make data easier for the human brain to understand and pull insights from**. The main goal of data visualization is to make it easier to identify patterns, trends and outliers in large data sets.



Here I plotted a line graph for a Loss. Orange line is representing Validation loss whereas Blue line is representing Training loss. As you can see in the above graph both training and validation loss decreased.

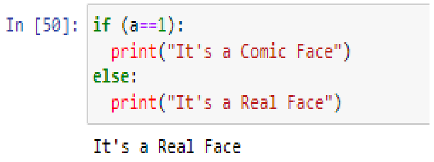


Here I again plotted a line graph for showing Accuracy. Again Orange line is representing Validation Accuracy whereas Blue line is representing Training Accuracy. As you can see in the above graph both training and validation Accuracy increased.

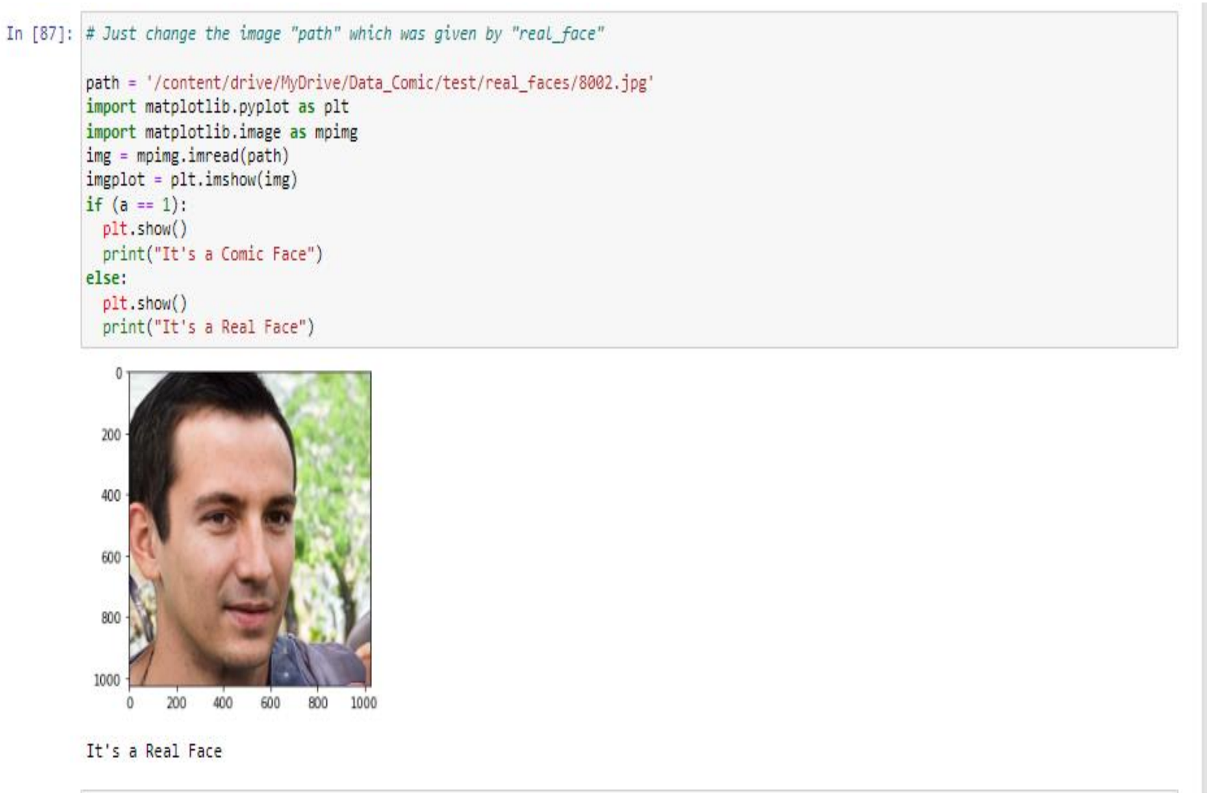
10.Model testing

Model testing is referred to as the process where the performance of a fully trained model is evaluated on a testing set. After testing the trained model by using inception v3 model was working correctly in both cases. First I tested for real faces and again I did same for testing comic faces.

In real faces case:



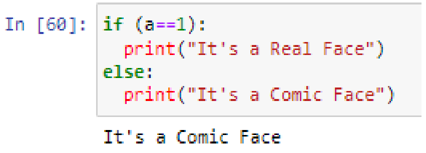
```
In [50]: if (a==1):
           print("It's a Comic Face")
         else:
           print("It's a Real Face")

         It's a Real Face
```



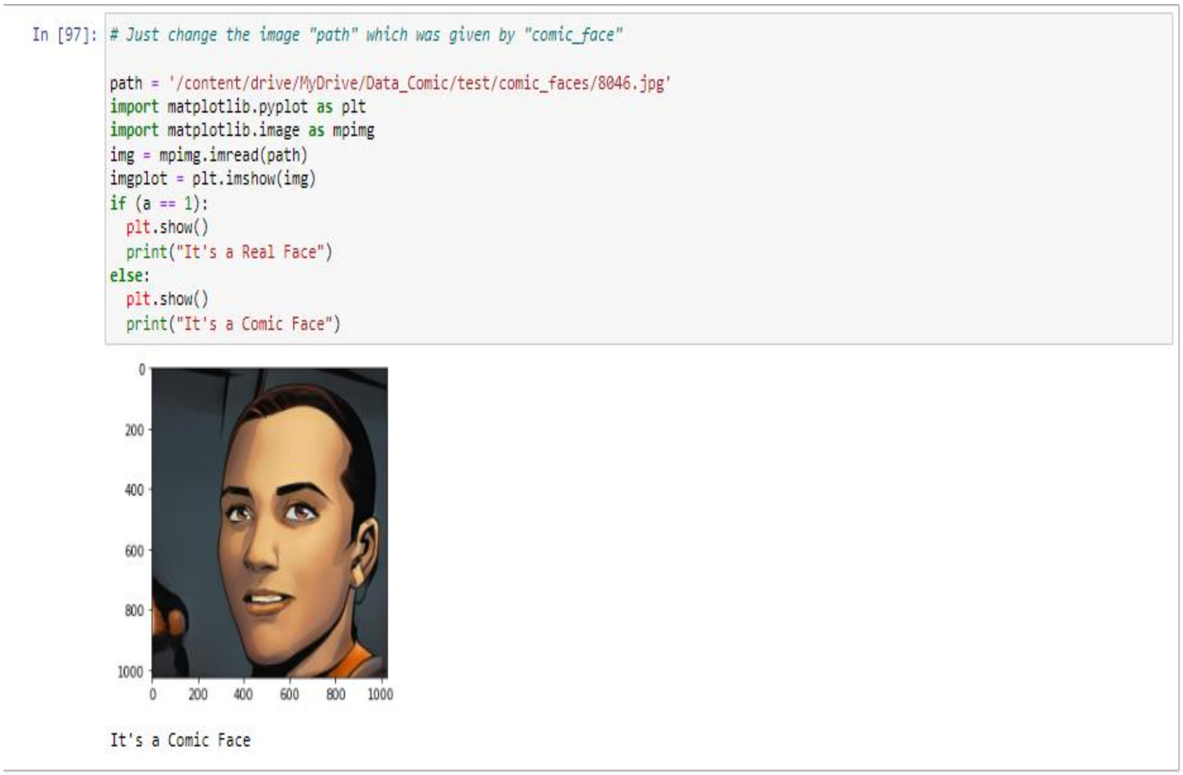
```
In [87]: # Just change the image "path" which was given by "real_face"

         path = '/content/drive/MyDrive/Data_Comic/test/real_faces/8002.jpg'
         import matplotlib.pyplot as plt
         import matplotlib.image as mpimg
         img = mpimg.imread(path)
         imgplot = plt.imshow(img)
         if (a == 1):
           plt.show()
           print("It's a Comic Face")
         else:
           plt.show()
           print("It's a Real Face")
```

```
It's a Real Face
```

In Comic faces case:



```
In [60]: if (a==1):
           print("It's a Real Face")
         else:
           print("It's a Comic Face")

         It's a Comic Face
```



```
In [97]: # Just change the image "path" which was given by "comic_face"

         path = '/content/drive/MyDrive/Data_Comic/test/comic_faces/8046.jpg'
         import matplotlib.pyplot as plt
         import matplotlib.image as mpimg
         img = mpimg.imread(path)
         imgplot = plt.imshow(img)
         if (a == 1):
           plt.show()
           print("It's a Real Face")
         else:
           plt.show()
           print("It's a Comic Face")
```

```
It's a Comic Face
```

11. Conclusion

Face recognition technology has come a long way in the last twenty years. Today, machines are able to automatically verify identity information for secure transactions, for surveillance and security tasks, and for access control to buildings etc. These applications usually work in controlled environments and recognition algorithms can take advantage of the environmental constraints to obtain high recognition accuracy. However, next generation face recognition systems are going to have widespread application in smart environments -- where computers and machines are more like helpful assistants.

The implementations and results demonstrate that deep learning techniques have great Performance in Face detection. Transfer learning model should be used to minimize time and also for the better performance of the model you can get quite good results as I got in this model. I got 100% accuracy for both training and testing model. Finally, there is still no robust face detection and recognition technique for unconstraint real-world applications.

For future work on this,we can convert this model to develop for mobile application just similar to snapchat. We can also try to convert this model into age detection application or gender detection application by changing the data accordingly.