FAKE FACE DETECTION USING METADATA ANALYZER AND DEEP LEARNING

A MINI PROJECT REPORT

Submitted by

GOKUL V	**1907010**
KALKIDAS K	**1907018**
SRINIVAS R	**1907049**

In partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

COIMBATORE INSTITUTE OF TECHNOLOGY, COIMBATORE-641014

(Government Aided Autonomous Institution Affiliated to Anna University)

ANNA UNIVERSITY, CHENNAI 600025

MAY 2022

COIMBATORE INSTITUTE OF TECHNOLOGY

(A Govt. Aided Autonomous Institution Affiliated to Anna University)

COIMBATORE – 641014

BONAFIDE CERTIFICATE

Certified that this project report titled "**FAKE FACE DETECTION USING DEEP LEARNING**" is the bonafide work of **GOKUL V (1907010), KALKIDAS K (1907018)** and **SRINIVAS R (1907049)** in partial fulfilment for the award of the Degree of Bachelor of Technology in Information Technology of Anna University, Chennai during the academic year 2021-2022 under my supervision.

Dr.N.K.KARTHIKEYAN
HEAD OF THE DEPARTMENT,
Department of Information Technology,
Coimbatore Institute of Technology,
Coimbatore - 641014.

Mr.C.MURALE,
SUPERVISOR,
Department of Information Technology,
Coimbatore Institute of Technology,
Coimbatore - 641014.

Certified that the candidates were examined by us in the project work viva-voice examination held on …………………

Internal Examiner **External Examiner**

Place:

Date:

s

ACKNOWLEDGEMENT

Our project **“FAKE FACE DETECTION USING DEEP LEARNING”** has been the result of motivation and encouragement from many, whom we would like to thank.

We express our sincere thanks to our Secretary **Dr.R.Prabhakar**,our Principal **Dr.A.Rajeswari** and advisor **Dr.V.Selladurai** for providing us a greater opportunity to carry out our work. The following words are rather very me agree to express our gratitude to them. This work is the outcome of their inspiration and product of plethora of their knowledge and rich experience.

We record the deep sense of gratefulness to **Dr.N.K.Karthikeyan**, Head of the Department of Information Technology, for his encouragement and support during this tenure

We equally tender my sincere gratitude to our project guide **Mr.C.Murale,** Department of Information Technology, for his valuable suggestions and guidance during this course.

During the entire period of study, the entire staff members of the Department of Computer Science and Engineering & Information Technology have offered ungrudging help. It is also a great pleasure to acknowledge the unfailing help we have received from our friends.

It is a matter of great pleasure to thank our parents and family members for their constant support and cooperation in the pursuit of this Endeavour.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

With advancements in technology, it is now possible to create representations of human faces in a seamless manner for fake media, leveraging the large-scale availability of videos. These fake faces can be used to conduct personation attacks on the targeted subjects. Availability of open source software and a variety of commercial applications provides an opportunity to generate fake videos of a particular target subject in a number of ways. In this article, we evaluate the generalizability of the fake face detection methods through a series of studies to benchmark the detection accuracy.

This research aims to evaluate the working of different deep learning techniques in the novel "Real and Fake Face detection" dataset by Computational Intelligence Photography Lab, Yonsei University. For the detection of forged faces, the first step of the proposed method is image normalization for real and fake image recognition. Normalized images are then preprocessed and train to different pre-trained deep learning models. We finetune these models for categorization of 2 classes that are forged and real to evaluate these model's performance.

Another feature used along with deep learning models is image metadata.Here it is used as a supporting parameter for deep learning model. Results of these models were evaluated using confusion matrix evaluation measures and compared with state of the art techniques.

CHAPTER 1

INTRODUCTION

1.1 Need for Fake face detection:

Face biometrics are widely deployed in various applications as it ensures reliable and convenient verification of a data subject. The dominant application of face recognition is for logical or physical access control to for instance restricted security areas. Implicitly the human visual system applies face recognition to determine, which data subject is the communication partner be it in a face to face conversation or be it in consuming messages while observing a media stream (e.g. news channel). With recent advances in deep learning, it is now possible to seamlessly generate manipulated images/videos in real-time using technologies like image morphing, Snap-Chat, Computer Generated Face Image (CGFI), Generative Adversarial Networks (GAN) and Face2Face.These technologies enable an attacker to manipulate the face image either by swapping it with another face or by pixel-wise manipulation to generate a new face image/video. It is well demonstrated in the literature that face recognition techniques fail drastically in detecting generated fake faces. Further fake face samples can also be shared by intention with the social media, in order to spread the fake news associated with the target subject. The challenge is not only posed to the biometric systems but also to the general media perception on social media. Thus it of paramount importance to detect faked face representations to reduce the vulnerability of biometrics systems and to reduce the impact of manipulated social media content.

1.2 Machine Learning:

Machine learning is one of the key application field in thriving Artificial Intelligence (AI). Machine learning has include ready to learn and upgrade the execution of the system utilizing past experience. It doesn't need a particular programming to do as such. Programming in machine learning focuses on getting to information and learning by utilizing the own information. It explicitly utilizes data or information like past models, direct insight, guidelines and required example in information is recognized to take better choices for development of framework in future. Automated learning, evasion of human intervention, performance improvement and suitable changes in activities of system are the key features what's more, attributes of machine learning. Deep learning is a subset of machine learning.Performance measure in numerous useful applications, for example image recognition, sound recognition and so forth is extraordinarily improved when deep learning is used. Machine Learning is now being used by businesses to improve business decisions, increase efficiency, identify disease, predict weather, and many other tasks. We need better tools to understand the data we have now, but we also need to plan for the data we will have in the future, thanksto the exponential growth of technology. To accomplish this, we must create intelligent machines. To do simple tasks, we can write a program. However, hardwiring intelligence intoit is always challenging. The best way to do it is to devise a method for machines to self- learn. A learning mechanism – if a computer can learn from feedback, it can do the heavy lifting for us.

1.3 Image Detection:

The Image Image or Object Detection is a computer technology that processes the image and identifies objects in it. Individuals frequently mistake Image Detection for Image Classification. If you need to classify image items, you use Classification. But if you just need to locate them, for example, find out the number of objects in the picture, you should use Image Detection. Image recognition is the ability of AI to detect the object, classify, and recognize it. The last step is close to the human level of image processing. Image recognition is the ability of a system or software to identify objects, people, places, and actions in images. It uses machine vision technologies with artificial intelligence and trained algorithms to recognize images through a camera system.

1.4 Visual Geometry Group :

The VGG is a classical convolutional neural network architecture. It was based on an analysis of how to increase the depth of such networks. The network utilises small 3 x 3 filters. Otherwise the network is characterized by its simplicity, the only other components being pooling layers and a fully connected layer. The VGG architecture is the basis of ground-breaking object recognition models. Developed as a deep neural network, the VGGNet also surpasses baselines on many tasks and datasets beyond ImageNet. Moreover, it is now still one of the most popular image recognition architectures. The concept of the VGG19 model (also VGGNet-19) is the same as the VGG16 except that it supports 19 layers. The “16” and “19” stand for the number of weight layers in the model (convolutional layers). This means that VGG19 has three more convolutional layers than VGG16.

1.5 Deep Learning:

Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. These neural networks attempt to simulate the behaviour of the human brain—albeit far from matching its ability—allowing it to “learn” from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy. Deep learning algorithms learn progressively more about the image as it goes through each neural network layer. Early layers learn how to detect low-level features like edges, and subsequent layers combine features from earlier layers into a more holistic representation. VGG-19 classifier is a trained Convolutional Neural Network. VGG-19 is a variant of VGG model which in short consists of 19 layers. There are other variants of VGG like VGG11, VGG16 and others. As the number of layers increases in CNN, the ability of the model to fit more complex functions also increases. Hence, more layers promise better performance.

CHAPTER – 2

LITERATURE SURVEY

2.1“IMAGE FORGERY DETECTION USING ERROR LEVEL ANALYSIS AND DEEP LEARNING.”

This paper was proposed by Sudiatmika, I. B. K., & Rahman, F in 2019.

METHODOLOGY

In this paper the dataset get is through CASIA version 2.0. Inside there are 7491 original images and 5123 tampered images. The size of the dataset is changed to 224x224 pixels. In this experiment, they divide the dataset into two namely training set and test set. In the range 50- 90% for the training set and the rest is used for test data. In compiling the dataset, we divide the data train and test data each of which there are 2 categories, namely the category of fake images and the original image. The first step they took was to divide the dataset from Casia V.2 into 2 categories: original and fake images. They normalize the image by processing the image to a size of 224x224 pixels. Then their next step is to perform analysis on the level of compression error image, from the compression result then they use the VGG 16 architecture for CNN in recognizing the original image and fake images according to the ELA. Our next step is to summarize the results of the training.

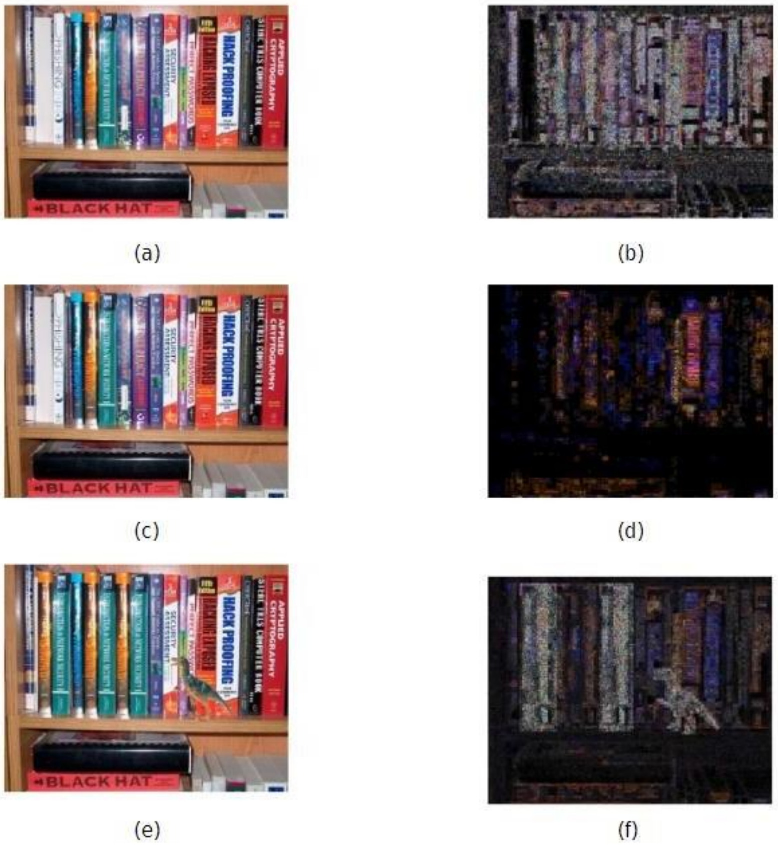


Figure 2. Error level analysis compression: (a) original image, (b) ELA original Image, (c) resave image, (d) ELA resave image, (e) tampered image, (d) ELA tampered image

Image forgery detection using error level analysis and... (Ida Bagus Kresna Sudiatmika)

Figure 2.1 Error Level Analysis images

2.2. "IMAGE RECOGNITION WITH DEEP LEARNING"

This paper was proposed by Md Tohidul Islam, Sagidur Rahman, Taskeed Jabid, B.M.Nafiz Karim Siddique at ICIIBMS, 2018.

METHDOLOGY

In this paper they used convolutional neural network approach to classify food images. The convolutional neural network is a category of neural network that has been proven very efficient in image classification. It learns the filters that in traditional algorithms were hand-engineered. They used an inception v3 model pre- trained with ImageNet.

The dataset consists of 16643 images grouped into 11 major food categories. The 11 food categories are Bread, Dairy products, Dessert, Egg, Fried food, Meat, Pasta, Rice, Seafood, Soup, and Vegetables. The food images are divided into three parts: training set with 9866 images, validation set with 3430 images and evaluation set with 3347 images. The pre-processing such as random rotation and horizontal flips help convolutional neural network models to be insensitive of the exact position of the object in the images. The ZCA whitening reduces the redundancy in the matrix of pixel images and highlights the structures and features of the images to the convolutional neural network. Then resized all the images to 299 x 299 x 3 to increase processing time and also to fit in Inception V3. The learning scheduler to set learning rate to 0.002 after 15 epochs and 0.0004 after 28 epochs. The dataset was divided into three parts: training, validation and evaluation. Then resized the images of evaluation part in 299 x x299 x 3 and evaluated the accuracy of the model by true positive (TP), true negative (TN), falsepositive (FP) and false negative (FN) after classification.

2.3. “IMAGE FORGERY DETECTION BASED ON GABOR WAVELETS AND LOCAL PHASE QUANTIZATION”

This paper was proposed by Meera Mary Isaaca , M Wilscy at Second International Symposium on Computer Vision and the Internet, 2015.

METHDOLOGY

They propose a novel image forgery detection technique by taking texture information of the image as a distinguishing feature. The method relies on Gabor wavelets and Local Phase Quantization (LPQ) which can extract relevant texture features which are fed as input to a Support Vector Machine for classification. The results indicate an accuracy of over 99% on both CASIA v1 and the DVMM color dataset. It outperforms similar state-of-art methods in solving image forgery detection.

2.4. “IMAGE-SPLICING FORGERY DETECTION BASED ON IMPROVED LBP AND K-NEAREST NEIGHBORS ALGORITHM”

This paper was proposed by Khan Hakimi, F., Zanjan, I., & Hariri, I. at Electronics Information and Planning in September 2015.

METHDOLOGY

An effective passive splicing image forgery detection scheme based on Improved Local Binary Pattern (LBP) and Discrete Cosine Transform (DCT) is proposed. First, the chrominance component of the input image is divided into non-overlapping blocks. Then, for each block, Improved LBP is calculated and transformed into frequency domain using 2D DCT. Standard deviations of frequency coefficients for all blocks are calculated and used as features K-Nearest Neighbors (KNN) algorithm is used for classification. Experimental results show the accuracy improvement for the proposed method in terms of the detection performance over CASIA1 and CASIA2 image splicing detection evaluation dataset.

2.5. “DETECTING BOTH MACHINE AND HUMAN CREATEDFAKE FACE IMAGES IN THE WILD.”

This paper was proposed by Tariq, Shahroz, et al at Proceedings of the 2nd international workshop on multimedia privacy and security in 2018.

METHDOLOGY

Creating fake images such as replacing one’s face with other person’s face has become much easier due to the advancement of sophisticated image editing tools. In addition, Generative Adversarial Networks (GANs) enable creating natural looking human faces. However, fake images can cause many potential problems, as they can be misused to abuse information, hurt people, and generate fake identification. Therefore, detecting fake face images is critical for protecting individuals from various misuses. In this work, we propose an image forensic platform using neural networks, FakeFaceDetect, to detect various fake face images. In particular, we focus on detecting fake images automatically created from GANs as well as manually created by humans.

CHAPTER 3

SYSTEM ARCHITECTURE

3.1 PROPOSED SYSTEM

3.1.1 Dataset Description

This dataset consists of RGB images with an extension of (.jpg) which were classified into real and fake classes. After preprocessing the images of the whole dataset, it was converted into ELA images. These images were split up into training and test sets, which were then forwarded to the Deep

CNN model to recognize real and fake images.

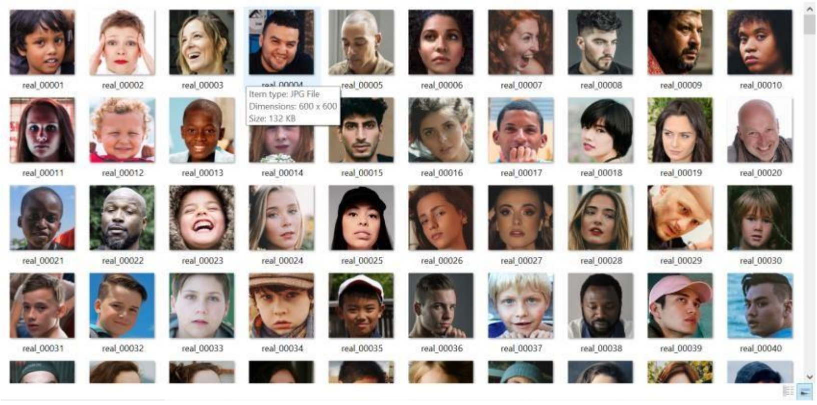


Figure 3.1. Images with real faces.

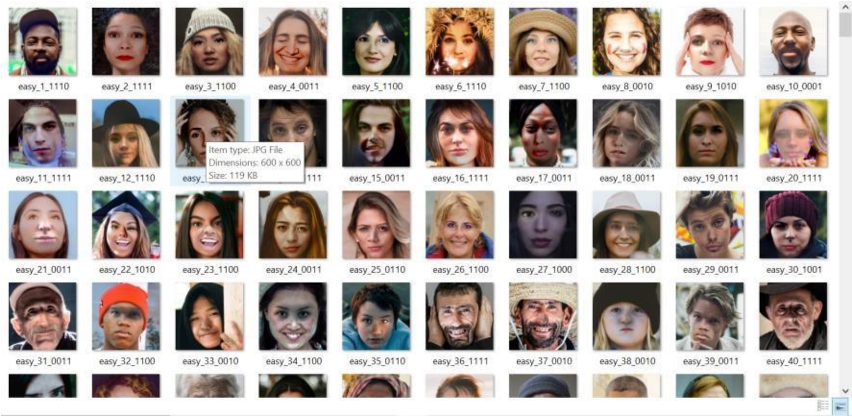


Figure 3.2. Images with fake faces

3.1.2 Methodology

Our proposed system is comprised of five main modules: Our proposed system is comprised of five main modules: Pre-processing, Metadata Analysis,Image Recognition using DeepLearning and Convolutional neural network.

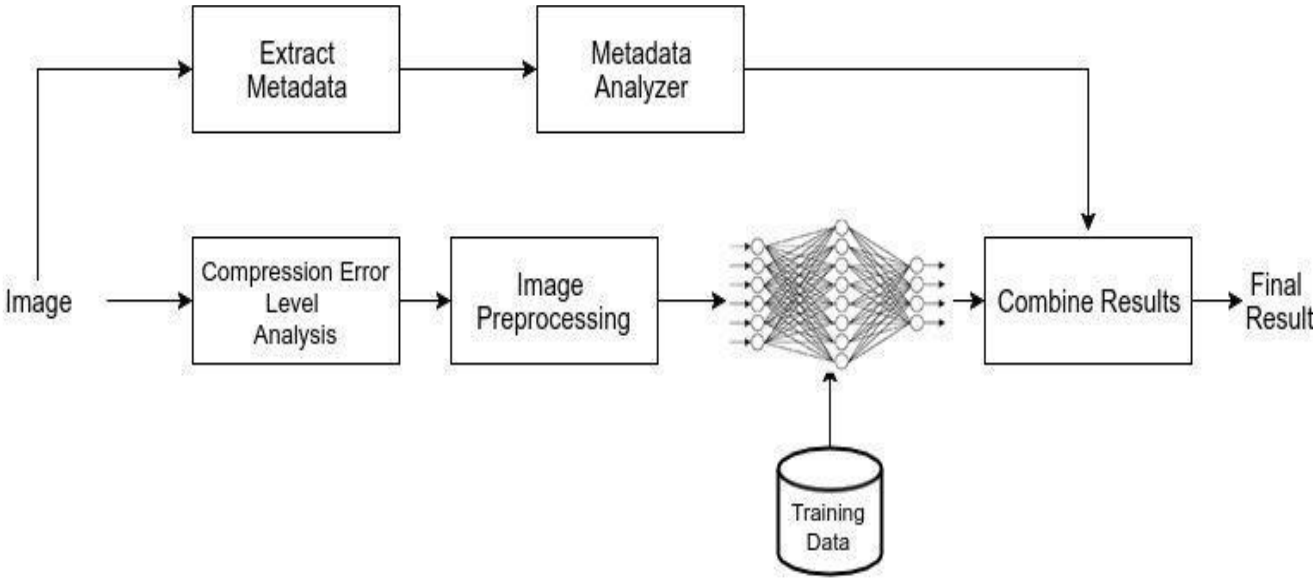


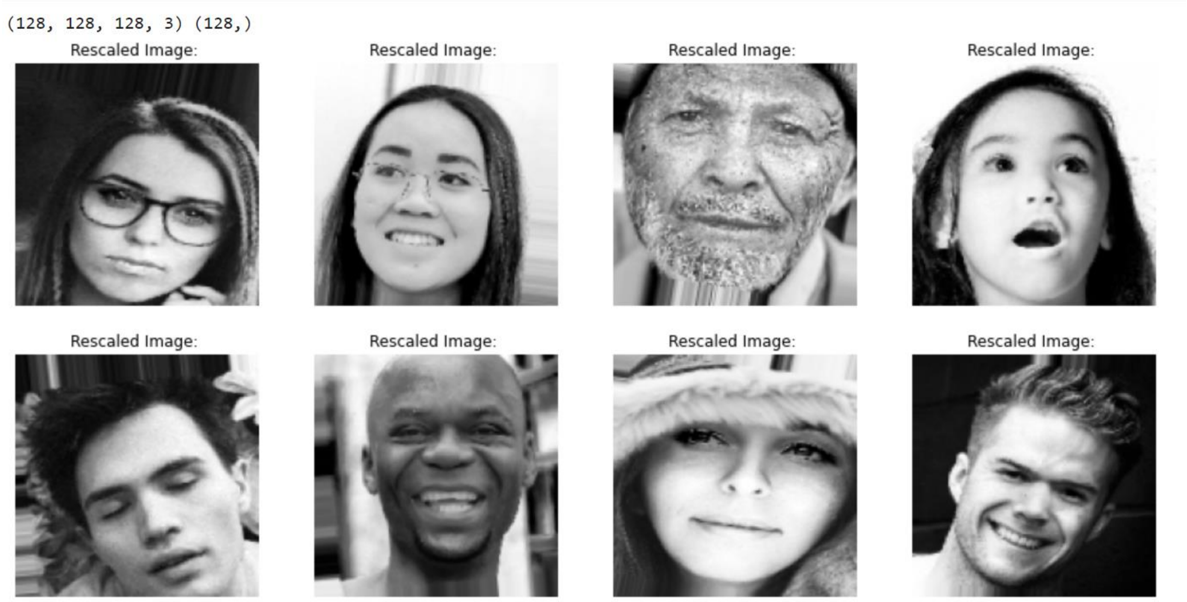
Fig.3.3 Proposed System

3.1.3 Image preprocessing

Deep learning models cannot use real-time data which contains missing values, contour and it also may be in an unsuitable format. To rectify this, preprocessing method is used. The aim of pre-processing is to supply improvement of the image information that suppresses unsought contours for more processing and analysis task. The images taken from the standard dataset will be preprocessed. The images which contains the contours will be removed which helps to upgrade the efficiency and accuracy of model. In order to find the part that contains only the brain of the image, contour technique is used to find the extreme top, bottom, left and right points of the brain.

Read the Image and convert it to Grayscale Format

Read the image from the dataset and convert the image to grayscale format. Converting the image to grayscale is very important as it prepares the image for the next step. The gray scale conversion is achieved using the code COLOR_BGR2GRAY in **cv2.cvtColor** function.

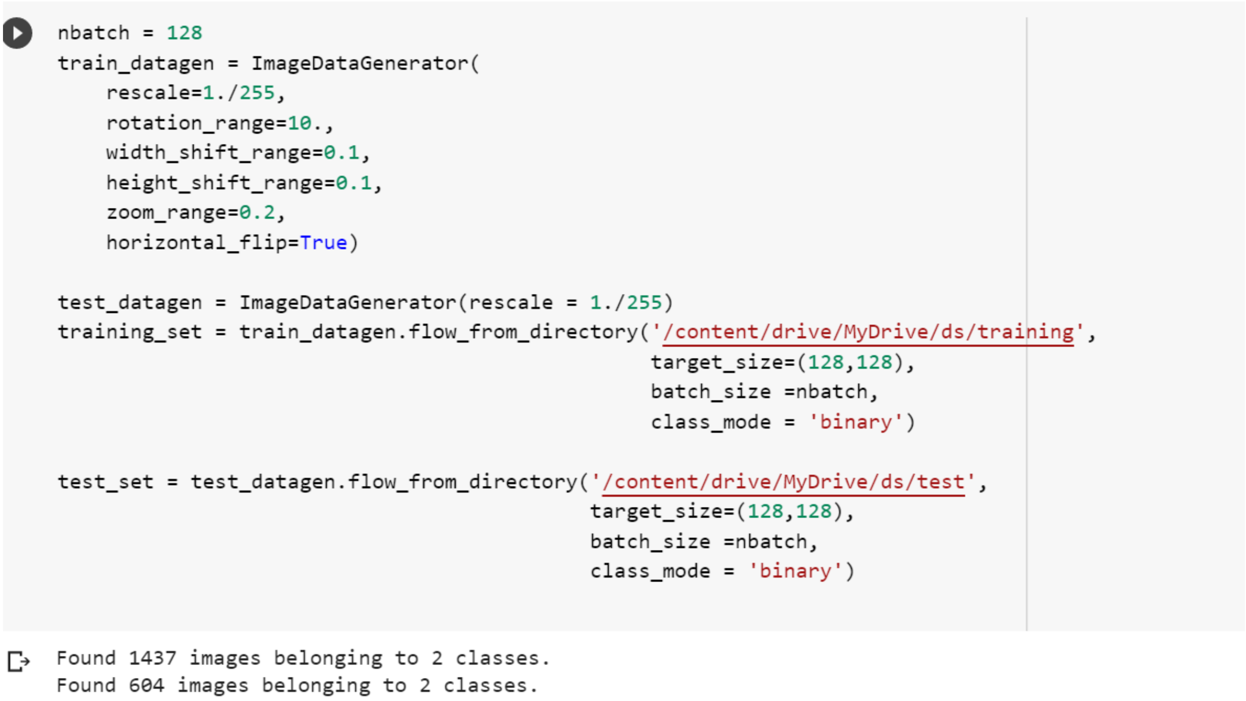


```
(128, 128, 128, 3) (128,)
```

Fig. 3.4 Reading the image from the dataset

ImageDataGenerator

ImageDataGenerator class of keras provides a quick and easy way to augment images. It provides a host of different augmentation techniques like rotation, shifts, flips, brightness change, zoom, and many more. It results in multiple transformed copies of the same image.



```
nbatch = 128
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=10.,
    width_shift_range=0.1,
    height_shift_range=0.1,
    zoom_range=0.2,
    horizontal_flip=True)

test_datagen = ImageDataGenerator(rescale = 1./255)
training_set = train_datagen.flow_from_directory('/content/drive/MyDrive/ds/training',
                                                 target_size=(128,128),
                                                 batch_size =nbatch,
                                                 class_mode = 'binary')

test_set = test_datagen.flow_from_directory('/content/drive/MyDrive/ds/test',
                                            target_size=(128,128),
                                            batch_size =nbatch,
                                            class_mode = 'binary')
```

```
Found 1437 images belonging to 2 classes.
Found 604 images belonging to 2 classes.
```

Fig. 3.5 Image augmentation using ImageDataGenerator.

3.1.4 VGG-16

The VGG16 model achieves almost 92.7% top-5 test accuracy in ImageNet. ImageNet is a dataset consisting of more than 14 million images belonging to nearly 1000 classes. Moreover, it was one of the most popular models submitted to ILSVRC-2014. It replaces the large kernel-sized filters with several 3×3 kernel-sized filters one after the other, thereby making significant improvements over AlexNet. The VGG16 model was trained using Nvidia Titan Black GPUs for multiple weeks.

The VGGNet-16 supports 16 layers and can classify images into 1000 object categories, including keyboard, animals, pencil, mouse, etc. Additionally, the model has an image input size of 128-by-128.

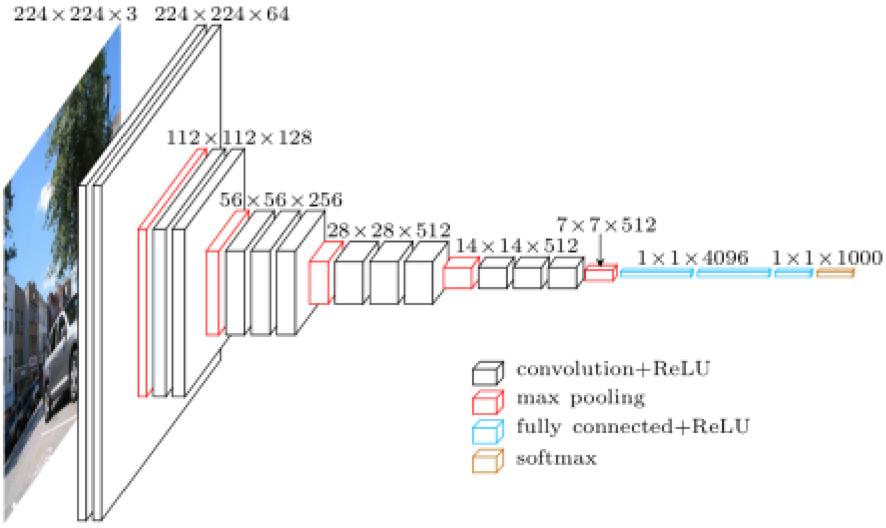


Fig.3.6 Working of VGG 16 Architecture.

Here the type of model built is Sequential. The epochs are run to train the model. Here the optimizer used is Adam. In this project, binary cross entropy is used for the classification.

Adam Optimizer:

Adam is the best among the adaptive optimizers in most of the cases. Adam optimization is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments. Adam is an adaptive learning rate optimization algorithm that’s been designed specifically for training deep neural networks. Adam is a combination of two gradient descent methods, Momentum, and RMSP. Momentum is an optimization algorithm that takes into consideration the 'exponentially weighted average' and accelerates the gradient descent. Root Mean Square Propagation (RMSP) is an adaptive optimization algorithm that is an improved version of AdaGrad. In RMSP we take the 'exponential average'.

Binary cross entropy classification:

Binary cross entropy may be a loss function that's utilized in binary classification tasks. Sigmoid is that the only activation function compatible with the binary cross entropy loss function.

Training loss and Validation loss:

Training loss is the error on the training set. Validation loss is the error after running the validation set of data through the trained network. In **Fig.3.7.** X-axis denotes the number of epochs and Y-axis denotes the loss obtained by each epochs. Orange line indicates the validation loss and blue line indicates the training loss.

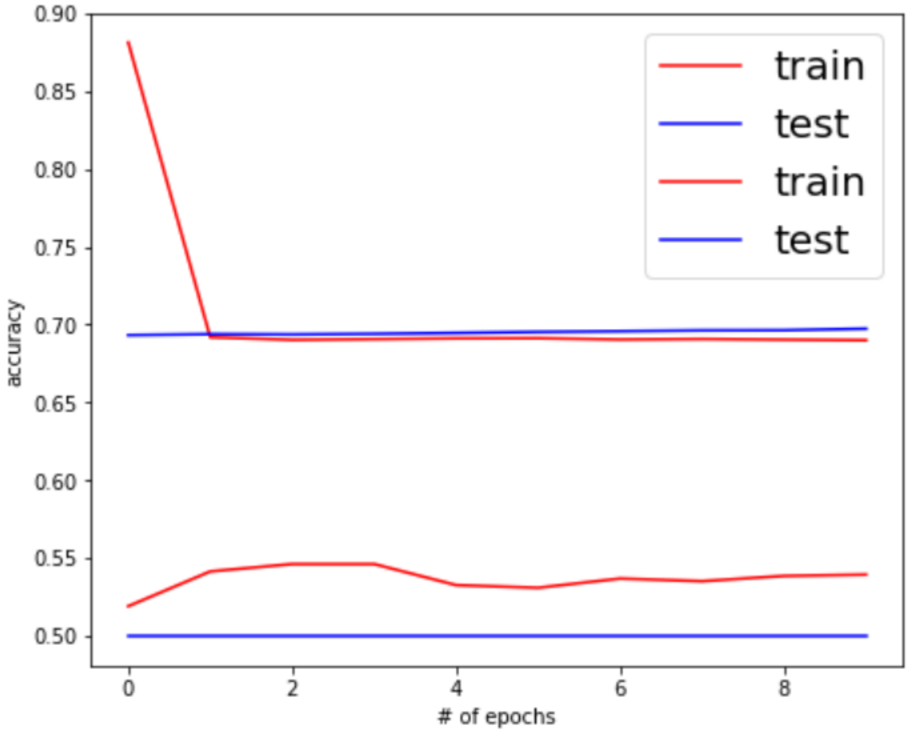


Fig.3.7 Training and validation loss graph

Training accuracy and Validation accuracy:

Training accuracy:

The accuracy of a model on examples it was constructed on.

Validation accuracy:

The test accuracy often refers to the validation accuracy, the accuracy calculated on the data set which is not used for training, but used for training for validating (or "testing") the generalization ability of your model or for "early stopping".

In **Fig.3.8.** X-axis denotes the number of epochs and Y-axis denotes the accuracy obtained by each epochs. Orange line indicates the validation accuracyand blue line indicates the training accuracy.

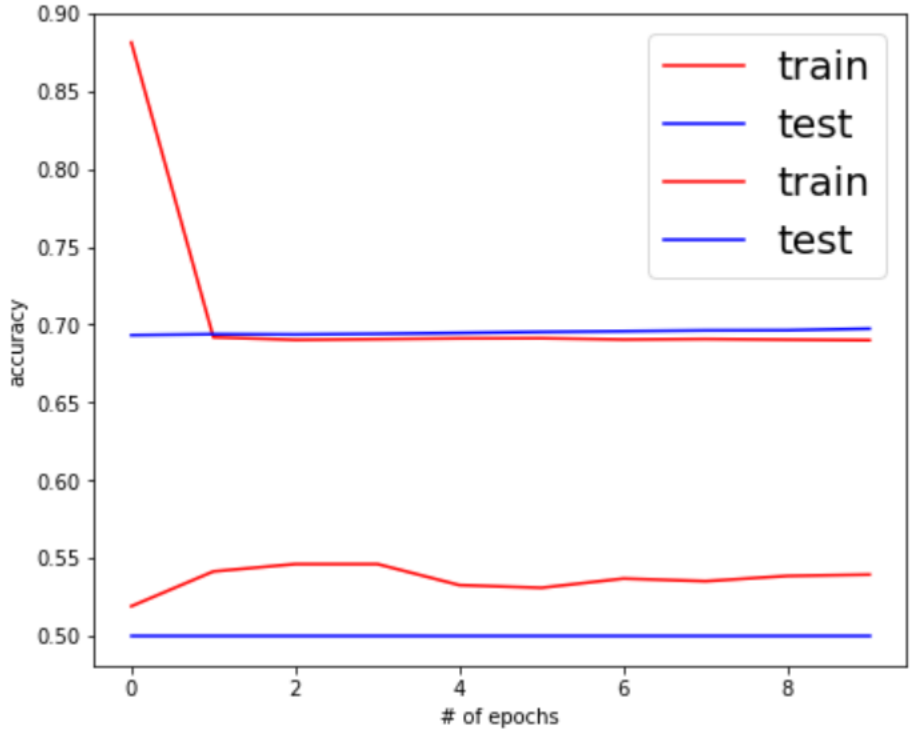
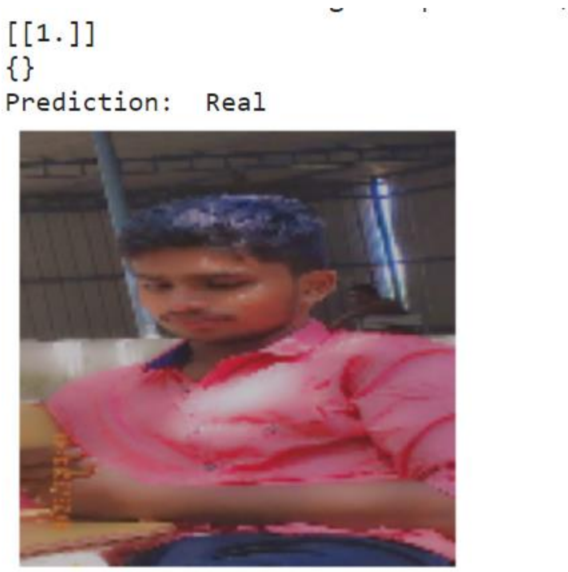


Fig.3.8 Training and validation accuracy graph

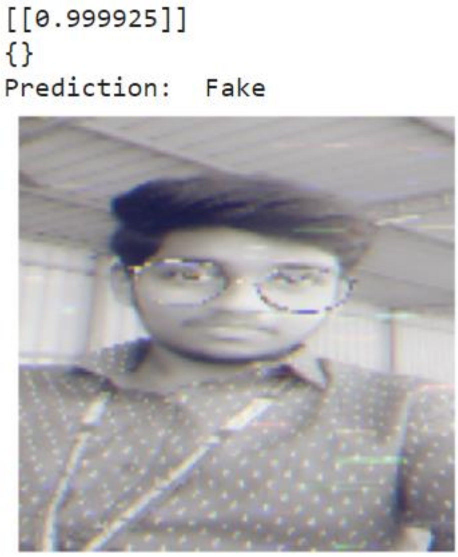
3.1.5 Predictions

Predictions are made using the batch size of 32 and Training and validation loss graph is generated. In addition to that training and validation accuracy graph is also plotted. Finally, the summary of the model is obtained and the model is saved to the disk. After saving the model the prediction is made. The final output is achieved by the processing of above-mentioned model in an efficient way and the maximum accuracy is met with. **Fig.3.9 and Fig.3.10** Shows the predictions of our model.



```
[[1.]]
{}
Prediction:  Real
```

Fig.3.9 Prediction of Real Face



```
[[0.999925]]
{}
Prediction:  Fake
```

Fig.3.10 Prediction of Fake face

CHAPTER 4

SYSTEM SPECIFICATION

This chapter includes the System Specification of our project.

The hardware and software for the system is selected by considering the factors such as CPU processing speed, peripheral channel speed, printer speed, seek time,relational delay of hard disk and communication speed etc. The hardware and software specifications are as follows.

4.1HARDWARE REQUIREMENTS

Processor	intel i3
Memory	4GB RAM
Camera	Web Camera
Monitor	Laptop or System Monitor

Table 4.1 Hardware requirements

4.2 SOFTWARE REQUIREMENTS

Operating System	Windows OS/Mac Os/Linux/Unix
IDE	Jupyter and google colab
Language	Python

Table 4.2 Software requirements

CHAPTER 5

PERFORMANCE ANALYSIS

5.1PERFORMANCE EVALUATION AND METRICS

5.1.1 CONFUSION MATRIX

The performance of our ensemble model was measured using a confusion matrix. Two or more types of classes were obtained as an outcome of the confusion matrix. ACTUAL and PREDICTED were the two dimensions of the confusion matrix. If the value obtained for actual and predicted class is 1, then it falls under true positives (TP). If the value obtained for actual and predicted class is 0, then it falls under the category of true negatives (TN). If the value obtained for actual and predicted class is 0 and 1 respectively, then it falls under false positives (FP). If the value obtained for actual and predicted class is 1 and 0 respectively, then it comes under false negatives (FN).

5.1.2 ACCURACY

It is the most often used statistic for assessing the performance of classification algorithms. It is expressed as the number of right predictions. The following formula is used to compute.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)} \qquad (1)$$

For determining, accuracy of our classification model, accuracy_score function has been used.

5.1.3 PRECISION

The number of correct documents put back by our proposed model can be termed as precision. The formula which is used to determine precision based on a confusion matrix.

$$\text{Precision} = tp / (tp + fp) \qquad (2)$$

5.1.4 RECALL

Recall is nothing but the number of positives put back by our proposed model. For the determination of recall using the confusion matrix [5][13], the formula is

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive + False Negative}} \qquad (3)$$

5.1.5 F1 SCORE

The F1 score is calculated as average of precision and recall. F1 will have a highest value of 1 and a worst value of 0. The formula used to compute F1 score is [1] [5]

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision + Recall}} \qquad (4)$$

CHAPTER 6

DESIGN AND IMPLEMENTATION

This chapter explains the design and implementation of Fake Face prediction.

The system is composed of the following modules

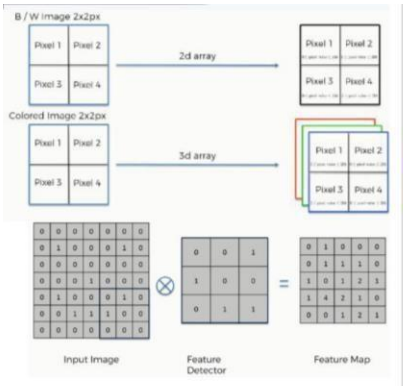
1. Image Processing.
2. Metadata Analyzer

6.1 Image Processing

When the animal's presence is detected the camera starts to capture image in the surrounding. It captures in different angles, those pictures are then compared with the already present dataset which comprises of different photos of various animals in different angles. It yields the result of the type of animal which has entered into the field.

6.1.1 Feature Extraction

In image recognition, the features are the groups of pixels, like edges and points, of an object that the network will analyze for patterns. Feature extraction is the process of pulling the relevant features out from an input image so that these features can be analyzed. This is feature extraction and it creates "feature maps".



0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

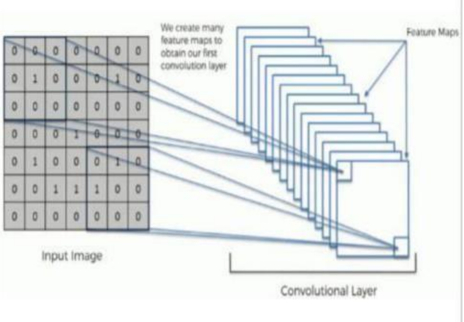
0	0	1
1	0	0
0	1	1

0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1	2	1

Fig 6.1 Feature Extraction

6.1.2 Convolutional Layer

This process of extracting features from an image is accomplished with a “convolutional layer", and convolution is simply forming a representation of part of an image. Convolutional neural network (CNN) most commonly used in image classification/recognition.



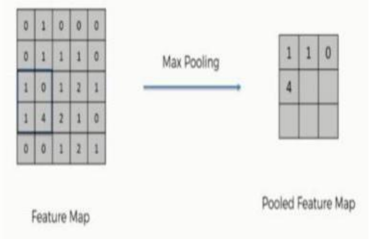
We create many feature maps to obtain our first convolution layer
Feature Maps
Input Image
Convolutional Layer

0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

Fig 6.2 Convolutional Layer

6.1.3 Pooling Layer

After the data is activated, it is sent through a pooling layer. Pooling takes the information which represents the image and compresses it, making it smaller. The pooling process makes the system more flexible and more apt at recognizing objects / images based on the relevant features.



0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1	2	1

1	1	0
4		

Fig 6.3 Pooling Layer

6.1.4 Flattening

The final layers of our CNN, the densely connected layers, require that the data is inthe form of a vector to be processed. For this reason, the data must be "flattened". The values are compressed into a long vector or a column of sequentially ordered numbers.

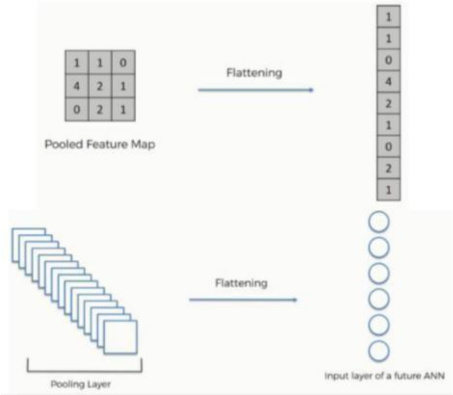


Fig 6.4 Flattening

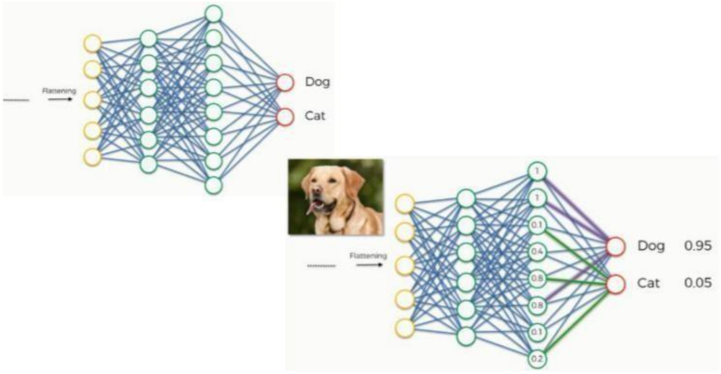


Fig 6.5 Flattening with an example

6.2 Metadata Analyzer

Devices such as digital cameras, smartphones, and scanners use the EXIF standard to save images or audio files. This standard contains many useful tags to extract which can be useful For forensic investigation, such as the make, model of the device, the exact date and time image creation, and even the GPS information on some devices.

6.3 Function:

6.3.1 Image Processor

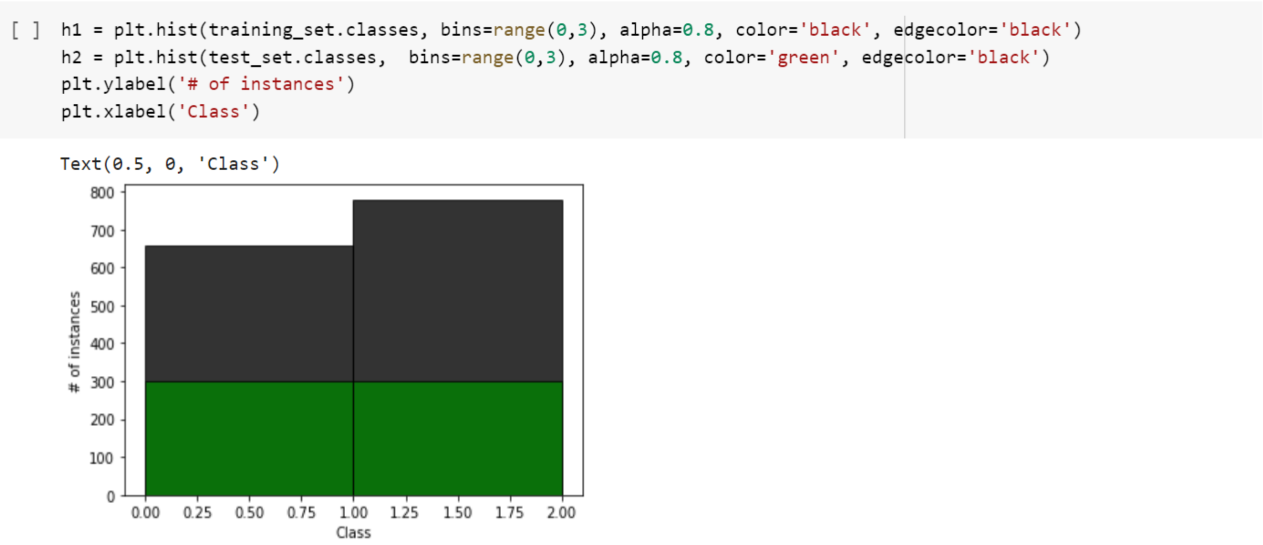
```
Input:Images given by User.
Output: Real/Fake Image.
Begin
  Step 1: When the user passes image it gets preprocessed.
  Step 2: Preprocessed image is passed to model.
  Step 3: Face is identified.
  Step 4: Print result as Real/Fake Image.
  End
```

6.3.2 Metadata Analyzer

```
Input: Images Given by user
Output: Metadata of that image.
Begin
  Step 1: When the user passes image it gets passed to metadata analyzer.
  Step 2: Tags found in image were analyzed
  Step 3: Print the tags in image
End
```

CHAPTER 7

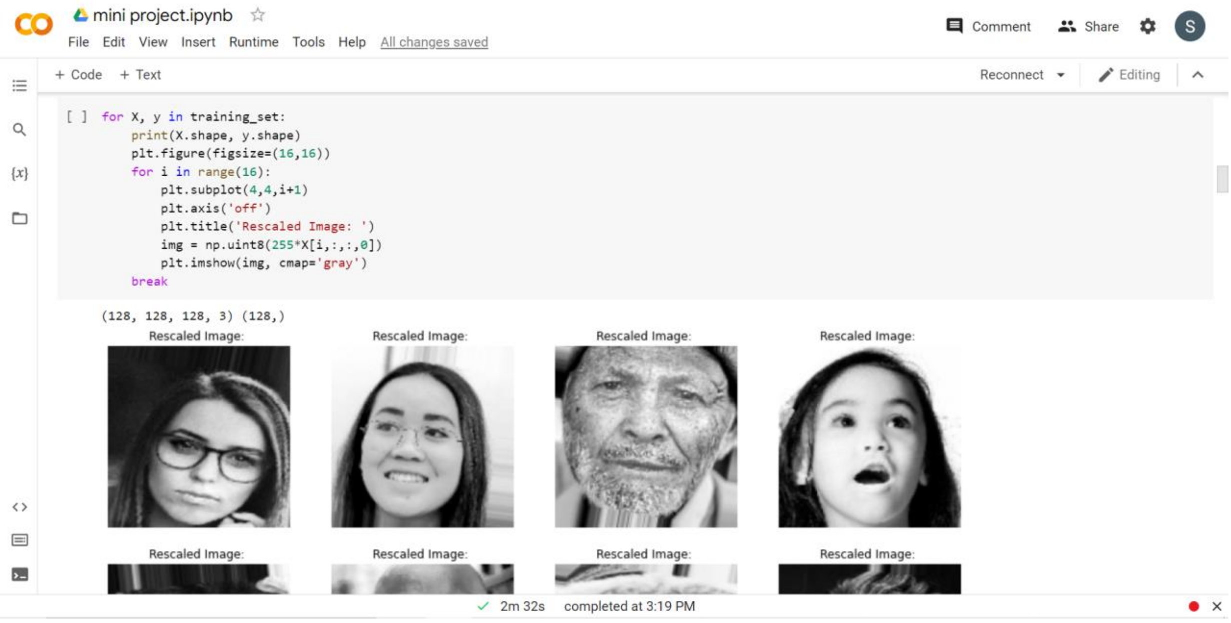
RESULTS



```
[ ]  h1 = plt.hist(training_set.classes, bins=range(0,3), alpha=0.8, color='black', edgecolor='black')
     h2 = plt.hist(test_set.classes,  bins=range(0,3), alpha=0.8, color='green', edgecolor='black')
     plt.ylabel('# of instances')
     plt.xlabel('Class')
```

```
Text(0.5, 0, 'Class')
```

Fig 7.1 Importing Images As Real and Fake classes.



```
[ ]  for X, y in training_set:
         print(X.shape, y.shape)
         plt.figure(figsize=(16,16))
         for i in range(16):
             plt.subplot(4,4,i+1)
             plt.axis('off')
             plt.title('Rescaled Image: ')
             img = np.uint8(255*X[i,:,:,0])
             plt.imshow(img, cmap='gray')
         break
```

```
(128, 128, 128, 3) (128,)
```

Fig 7.2 Preprocessing Image

```
Text(0.5, 0, '# of epochs')
```

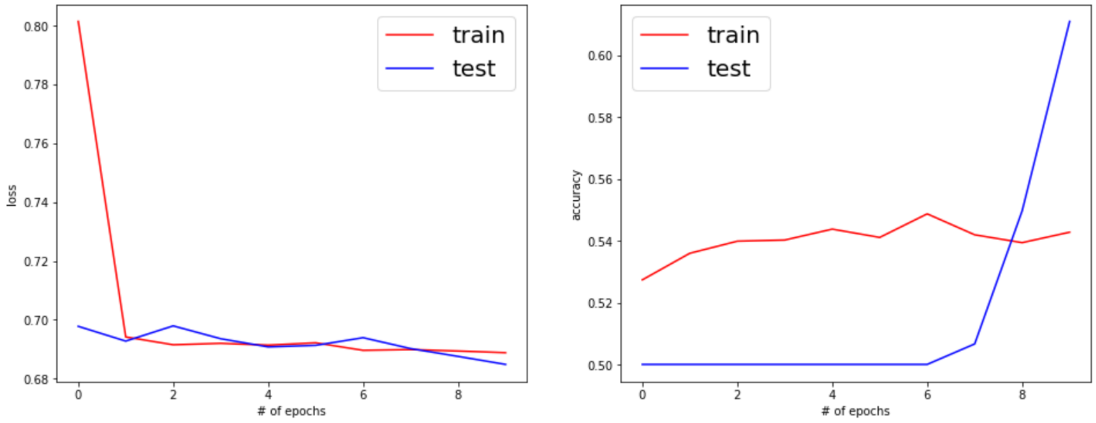
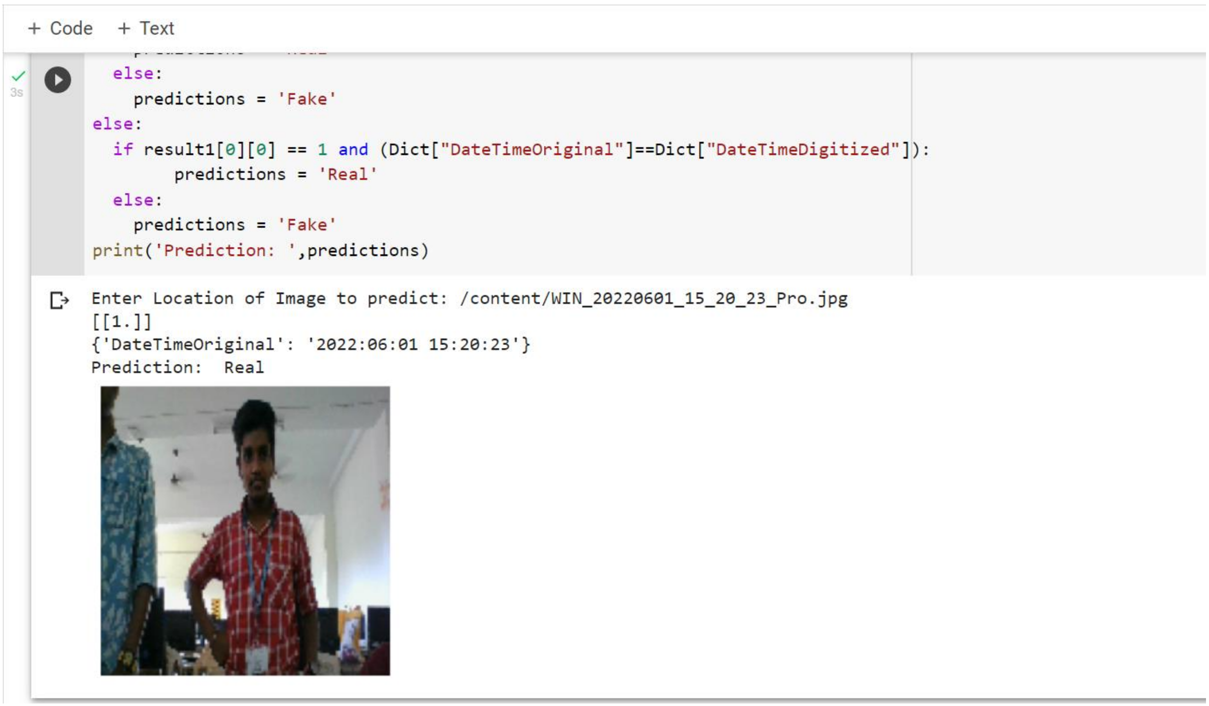


Fig 7.3 Value Loss and Value Accuracy of model.

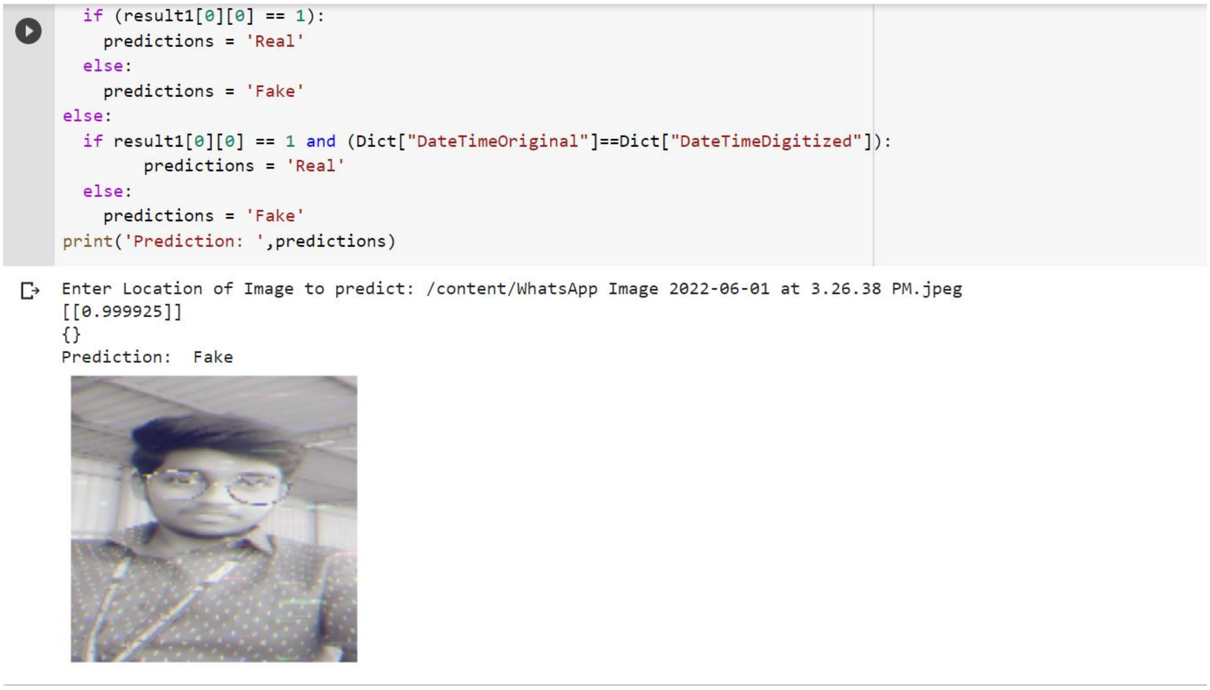


```
  else:
    predictions = 'Fake'
else:
  if result1[0][0] == 1 and (Dict["DateTimeOriginal"]==Dict["DateTimeDigitized"]):
        predictions = 'Real'
  else:
    predictions = 'Fake'
print('Prediction: ',predictions)
```

```
Enter Location of Image to predict: /content/WIN_20220601_15_20_23_Pro.jpg
[[1.]]
{'DateTimeOriginal': '2022:06:01 15:20:23'}
Prediction:  Real
```

Fig 7.4 Prediction of Real Faces.

.



```
  if (result1[0][0] == 1):
    predictions = 'Real'
  else:
    predictions = 'Fake'
else:
  if result1[0][0] == 1 and (Dict["DateTimeOriginal"]==Dict["DateTimeDigitized"]):
        predictions = 'Real'
  else:
    predictions = 'Fake'
print('Prediction: ',predictions)
```

```
Enter Location of Image to predict: /content/WhatsApp Image 2022-06-01 at 3.26.38 PM.jpeg
[[0.999925]]
{}
Prediction:  Fake
```

Fig 7.5 Prediction of Fake Faces.

CHAPTER 8

CONCLUSION AND FUTURE WORK

A user friendly, self-intuitive dataset of customized faces is created to enable ease of use for both society and government. The otherwise cumbersome and heavyweight models is made simpler with the use of VGG 16 framework. We have achieved the maximum accuracy of 92.3% using the VGG 16 algorithm and metadata analyser, which is much better than the results provided by other existing systems.

In future the size of the customizable dataset can be increased by adding more sample images. The dataset could also be made available on a cloud to make it universally accessible. We could detect faces to estimate whether the face is real or get morphed. We could export this model and estimate Originality of face in that picture..An interface can be made for the ease of accessing the system.

CHAPTER 9
APPENDIX

9.1 SOURCE CODE

9.1.1 Model.py:

```
from tensorflow.keras.models import load_model
from keras.preprocessing import image
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
model=load_model("/content/drive/MyDrive/model_checkpoint.hdf5")
```

9.1.2 Prediction.py:

```
from PIL import Image
from PIL.Image import Exif
from PIL.ExifTags import TAGS
Dict={}


def get_exif(file_name) -> Exif:
    image: Image.Image = Image.open(file_name)
    return image.getexif()


def get_exif_ifd(exif):
    for key, value in TAGS.items():
        if value == "ExifOffset":
            break
    info = exif.get_ifd(key)
    for key,value in info.items():
        if(TAGS.get(key, key)=="DateTimeOriginal" or TAGS.get(key, key)=="DateTimeDigitized"):
            Dict.update({TAGS.get(key, key):value})
    return Dict
```

```
def ImagePrediction(loc):
    test_image = image.load_img(loc, target_size = (128,128))
    plt.axis('off')
    plt.imshow(test_image)
    test_image = image.img_to_array(test_image)
    test_image = np.expand_dims(test_image, axis =0)
    result = model.predict(test_image)
    return result
img = input("Enter Location of Image to predict: ")
result1 = ImagePrediction(img)
print(result1)

exif = get_exif(img)
get_exif_ifd(exif)
print(Dict)
if (len(Dict)<2):
  if (result1[0][0] == 1):
    predictions = 'Real'
  else:
    predictions = 'Fake'
else:
  if result1[0][0] == 1 and (Dict["DateTimeOriginal"]==Dict["DateTimeDigitized"]):
        predictions = 'Real'
  else:
    predictions = 'Fake'
print('Prediction: ',predictions)
```