KLE Society's
KLE Technological University, Hubballi.

A Minor Project Report

on

Building Android Application To Detect Doctored Images Using Deep Learning Model

submitted in partial fulfillment of the requirement for the degree of

Bachelor of Engineering

in

Computer Science and Engineering

Submitted by

Kiran Jarali	01FE18BCS093
Komal Lonakar	01FE18BCS096
Swapnil Kore	01FE18BCS098
Naazmin P Maldar	01FE18BCS117

Under the guidance of
Asso. Prof. Lalita Madanbhavi
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Hubballi – 580 031

2020 -21

KLE Society's
KLE Technological University, Hubballi.

2020 - 2021

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that Minor Project titled “Building Android Application To Detect Doctored Images Using Deep Learning Model” is a bonafied work carried out by the student team comprising of Kiran Jarali (01FE18BCS093), Komal Lonakar (01FE18BCS096), Swapnil Kore (01FE18BCS098), Naazmin P Maldar (01FE18BCS117) for partial fulfillment of completion of sixth semester B.E. in Computer Science and Engineering during the academic year 2020-21.

Guide Head, SoCSE

Asso. Prof. Lalita Madanbhavi Dr. Meena S. M

Viva -Voce:

Name of the Examiners Signature with date

1.

2.

ACKNOWLEDGEMENT

The satisfaction that accompanies the completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement ground our efforts and success. One cannot even imagine the power of the force that guides us all and neither can succeed without acknowledging it. Our deepest gratitude to the almighty for holding our hands and guiding us throughout our lives. We take the privilege of thanking our guide Mrs. Lalita Madanbhavi, Associate Professor of School of Computer Science and Engineering KLE Technological University Hubballi for her expert guidance, encouragement invaluable support at every step.

We take this opportunity to thank Dr. Ashok Shettar, Vice-chancellor and to Dr. N H Ayachit, Registrar, KLE Technological University, Hubballi.

We also take this opportunity to thank Dr. Meena S M, Professor and Head of Department, Department of Computer Science and Engineering for having provided us academic environment which nurtured our practical skills contributing to the success of our project.

We sincerely thank our guide Ms. Lalita Madanbhavi Designation, Department of Computer Science and Engineering for his guidance, inspiration and wholehearted co-operation during the course of completion.

We sincerely thank our project co-ordinators Dr. P.G Sunita Hiremath and Dr. Sujatha C. for their support, inspiration and wholehearted co-operation during the course of completion.

Our gratitude will not be complete without thanking the Almighty God, our beloved parents, our seniors and our friends who have been a constant source of blessings and aspirations.

Kiran Jarali - 01FE18BCS093
Komal Lonakar - 01FE18BCS096
Swapnil Kore - 01FE18BCS098
Naazmin P Maldar - 01FE18BCS117

ABSTRACT

Digital Image Forgery can be done by deceiving the digital image to mask some meaningful or important data of the image. It is usually difficult to spot out the manipulated region of the original image. To sustain the uprightness and legitimacy of the image, the detection of forgery in the image is mandating. Acclimation of the modern way of life and advancement in photography gadgetry has made exploitation of digital image easy with the help of image editing software.Therefore, it is crucial to detect such image forgery operations in the images. The image forgery detection can be done based on object removal, object addition, unusual size modifications in the image. Images are one of the powerful media for communication.

The primary contributors to the dissemination of fraudulent photographs are social media. Fake photos are images that have been modified using software or other methods to distort the information they communicate. Fake photos shared on social media platforms lead to distortion and divisiveness among the public. The detection of false photos uploaded on social media sites is important to halting their spread. Textual data is frequently coupled with fake visuals. As a result, a multi-modal framework based on visual and textual feature learning is used. However, just a few multi-modal frameworks have been presented, and they all rely on additional activities to understand the relationship across modalities.

Keywords : *Cloning, Splicing, Retouching, Morphing, Copy-move forgery.*

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

We live in an era when we have access to a wealth of digital imagery. We used to have blind trust in the purity and authenticity of this imagery, but modern technology has eroded that trust. From prestigious periodicals to the media industry, courts, fashion outlets, scientific journals, political campaigns, and photographic satire that fall in our e-mail inboxes and social media platforms Photographs that have been forged are becoming more common. Without a question, image authenticity is a major source of concern these days.

With the proliferation of advanced picture editing tools and software available on portable devices such as smartphones and laptops, it is now possible to change photos quickly and easily for a variety of purposes. The photos can be changed for a better goal, but if they're changed for a bad reason, it's referred to as a fabrication. The photograph forgery will be accomplished in order to conceal a few important records, such as a man or woman or an object in the photograph. The modified photographs are used in court as fictitious evidence in order to gain money by increasing social media traffic. gaining notoriety or popularity, and so forth.There may be a need to verify the integrity of the images in order to prevent the distribution or sale of bogus records, as well as to avoid trusting and considering the modified images as evidence in a court of law.

There are various types of picture modification, with Copy-move and Splicing being the most common. Copy-move forgery involves copying a small section of an image and pasting it into a similar photo in order to trade the information contained in the image. Splicing forgery is accomplished by replacing a portion of an image with a portion of an extraordinary photograph in order to control the information transmitted. The tampering is done in such a way that the changes are difficult to detect with the naked eye. The solid region can be rotated, scaled, blurred, and other effects can be applied to it throughout the technique. To blend in with the surroundings and avoid discovery. Occasionally, falsified photos are put through up-processing operations such as smoothing to erase traces that appear at the rims during the modification of photos.As a result, locating cast images is a significant undertaking because they can be exploited for unscrupulous purposes.

1.1 Motivation

The rapid development of image processing software (for example, Adobe Photoshop(a)) and advances in digital cameras have resulted in a large number of doctored images with no obvious traces, creating a high demand for automatic forgery detection algorithms to determine the trustworthiness of a candidate image.A passive forgery detection system requires no prior knowledge of the image content or any protection mechanisms such as watermarks.

Image doctoring is becoming a popular research topic among image forensics experts. Duplicating one region in a picture, adding bits of another image, adjusting image attributes like as brightness, contrast, saturation, and hue, eliminating portions of images, and so on are all part of the process.

Advances in technology have made it simple to doctor an authentic image in a variety of ways, and because photos are used as evidence in a variety of situations, determining the authenticity of an image is critical.

1.2 Literature Review

1. We used the bispectral analysis approach and the Expectation-Maximization algorithm to solve the challenge of detecting image doctoring. For spliced, cloned, and retouched doctored photos, we investigated the detection of doctoring. The detection of doctoring has been proposed utilising the bicoherence magnitude and phase response. The doctoring process improves the magnitude and phase aspects of the bicoherence, as well as the prediction of phase bias.

   We tested the algorithm on a variety of doctored photographs and found that it had an average accuracy of 60% for the entire collection of photographs For spliced images, bispectral analysis is more effective. As a result, the Expectation-Maximization technique is utilised to recognise cloned and retouched photos that require an interpolation step in order to produce good doctored images. To detect doctoring, we leverage the periodicity in the Fourier transform of the probability maps. The technique was demonstrated on a variety of doctored photos, including cloned and retouched photographs.

2. One of the most active study fields in the field of blind image forensics is copy-move forgery detection (CMFD). The majority of known algorithms are based on block and

key-point approaches, or a combination of both. Some deep convolutional neural networks methods have recently been used in picture classification, image forensics, image hashing retrieval, and other areas, and have demonstrated to perform better than standard methods. A novel copy-move forgery detection system based on convolutional neural networks is proposed in this paper.

The suggested method takes an already trained model from a big database, such as ImageNet, and tweaks the net structure significantly with small training examples. The proposed method achieves satisfactory performance when compared to a forgery picture generated automatically by a computer using a basic image copy-move operation, according to experimental results.

3. The proposed CNN is intended for applications such as picture splicing and copy-move detection. The weights at the first layer of our network are initialised with the basic high-pass filter set used in the calculation of residual maps in the spatial rich model (SRM), which serves as a regularizer to effectively suppress the effect of image contents and capture the subtle artefacts introduced by the tampering operations, rather than a random strategy. The pre-trained CNN is utilised as a patch descriptor to extract dense features from the test images, and then a feature fusion technique is employed to get the final discriminative features for SVM classification.The suggested CNN-based model outperforms various state-of-the-art approaches, according to experimental results on multiple public datasets.

   A new image fraud detection system based on deep learning technology is presented in this paper, which uses a convolutional neural network (CNN) to automatically learn hierarchical representations from input RGB colour photographs.

4. Verifying the integrity of photographs and identifying indications of tampering without additional knowledge of the image content or any embedded watermarks is a key study area. A study of current breakthroughs in the field of digital picture forgery detection is attempted, and a complete bibliography on blind forgery detection approaches is offered.Methods that are blind or passive do not require any prior knowledge of the image. After classifying several picture forgery detection algorithms, a generalised structure is built. The available blind forgery detection approaches are discussed, as well as an overview of passive picture authentication. The current state of image forgery detection is presented, along with a research recommendation for the future.

Because of powerful computers, advanced photo-editing software packages, and high-resolution capturing devices, manipulating digital photographs has become simple.

5. We present an excellent method for detecting copy-move forgeries in digital photos in this study. This method begins by extracting an image's SIFT descriptors, which are invariant to changes in illumination, rotation, scale, and other factors. Descriptors are then checked against each other to look for any probable forgeries in images due to the resemblance between pasted and copied regions. Experiments were conducted to illustrate the method's efficacy on various forgeries and to evaluate its robustness and sensitivity to post-image processing, such as additive noise and lossy JPEG compression, or even compound processing.

   It got easier to modify photos without leaving any tiny traces because to the existence of highly sophisticated tools for modifying digital images. The most popular approach is copy-move forgery, which involves copying a portion of a picture and pasting it into another section of the same image. As a result, copy-move forgery detection (CMFD) systems seek to identify regions in fabricated images that are the same or similar.After the Copy-move operation, some post-processing actions can be done on the faked images, making forging detection more difficult. Typically, post-processing processes such as geometric transformations are used to hide the counterfeit.

The problems that would be solved on a technical level

- To ensure that digitally stored intellectual property is genuine.
- To improve the visual clarity of photographs without changing the meaning or provenance of digital materials.
- Information migration from digital sources.

1.3 Problem Statement

Develop an Android app that can detect digitally modified photos in multiple formats (Jpeg, png) and anticipate whether the output will be Authentic or Doctored based on the image states.

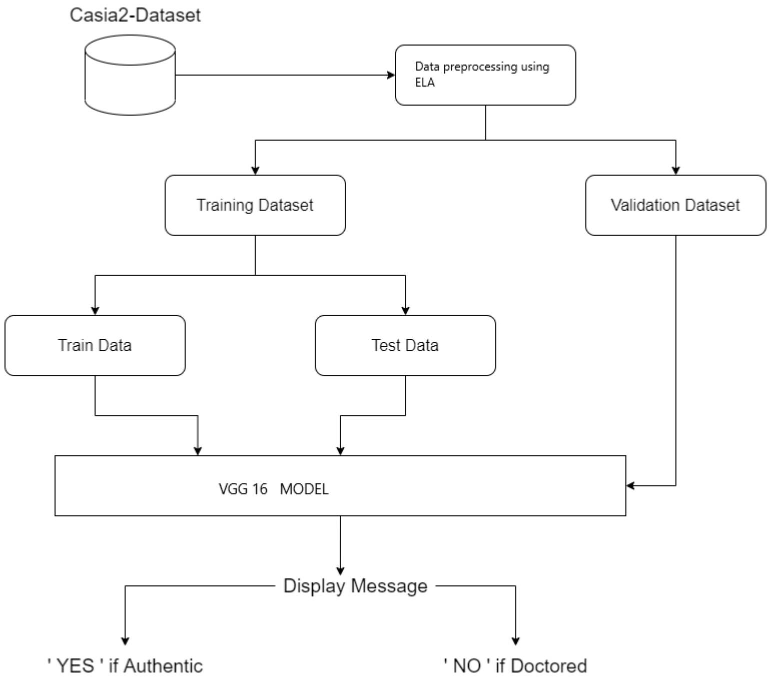


Figure 1.1: Flow diagram

1.4 Applications

- Investigations in Forensics - Aids in the detection of digital manipulation of photographs and intellectual property.
- To remove bogus photographs from social media that promote incorrect information among its users.
- News Networks - Before posting the news, it was necessary to double-check the photos.

1.5 Objectives and Scope of the project

The goal of this project is to cover the fundamental ideas involved in detecting forged images, as listed below, including providing appropriate visuals to explain key topics and presenting data that might be utilised to proportion vital strengthening parts.

By employing data augmentations and putting pristine images in the train data set with default mask, there is a lot of room to improve the performance of the Vgg16 Model.

1.5.1 Objectives

- To build a model to detect the doctored images and the area being tampered.

- Model should be capable of supporting multiple image format.
- Developing android application.

1.5.2 Scope of the project

The future scope would be to expand the training picture collection for better image validation, and to develop a better application that could be hosted on a server and produce more accurate results.

Chapter 2

REQUIREMENT ANALYSIS

In this chapter, we'll look at the system model, the steps of requirement analysis, and how to increase the quality of requirements. We'll also learn about the project's specified functional and non-functional needs, as well as its hardware and software specifications. The goal of the project's requirements analysis phase is to identify basic baseline functional requirements, policies, and procedures for a digital archive that are compliant with accepted standards.

Following elicitation, requirement analysis is an important and necessary process. To create uniform and unambiguous requirements, we examine, improve, and scrutinise the obtained requirements. This exercise goes over all of the requirements and may show a graphical representation of the full system. The project's understandability is predicted to improve greatly following the conclusion of the analysis.

The tasks of requirements analysis include determining the needs or conditions to meet the new or altered product or project, taking into account the possibly conflicting requirements of various stakeholders, analysing, documenting, validating, and managing software or system requirements, and analysing, documenting, validating, and managing software or system requirements.

The success or failure of a systems or software project is determined by the requirements analysis. The requirements should be documented, actionable, quantifiable, testable, and traceable, as well as tied to identified business needs or opportunities and described in sufficient detail for system design.

2.1 Functional Requirements

Function specifications define how everything must be accomplished by stating the function, activity, and operation that must be completed. The functional analysis of the top-level functions will be based on functional specifications. The following are the functional requirements as identified in the project:

2.1.1 User Level

- The user shall get to know if the image is original or doctored.
- The user shall be able to upload any image of any format into the system.

2.1.2 System Level

- The system shall be able to display output.
- The system shall be able to detect whether an image is original or doctored.

2.2 Non Functional Requirements

Nonfunctional requirements (NFR) are a set of guidelines for evaluating a system's performance without taking into account other elements. These are often known as a system's "quality qualities."From traits like stability and portability, NFR are frequently referred to as "qualities," "qualiy goals," "quality of service requirements," "constraints," "non-behavioral requirements," and "technical requirements."

2.2.1 Performance Requirements

- Once given the input, the response should be spontaneous.
- Detection should be at least 70% accurate.
- Detect whether the image is original or doctored. NFR1

2.2.2 Safety and Security Requirements

- The complete manipulation of particular user data in a database only for that user.

2.2.3 Environment Requirements

- Tool: Python 3.6
- Portability: The software can be deployed on any machine having python3 or above installed with other essential APIs like tensorflow, OpenCV, and tflearn

2.3 Hardware Requirements

- Processor: Intel i3 2120 or above
- Memory: 4GB or above
- GPU(optional): Gforce 210 1GB ddr3 or above
- HDD: 5400rpm or above, SSD is preferred
- Monitor, Mouse and other essential computer peripherals Python 3.6

2.4 Software Requirements

- Python 3.6
- Tensorflow 1.5
- Tflearn 1.18
- Open CV2
- Other python libraries such as pandas,numpy etc

Chapter 3

SYSTEM DESIGN

The process of defining the elements of a system, such as the architecture, modules, and components, as well as the many interfaces between those components and the data that flows through it, is known as system design. Understanding component pieces and how they interact with one another is the essence of system design.

System design is meant to be the link between the system architecture (at whichever point in the systems engineering process this milestone is defined) and the implementation of technological system pieces that make up the system's physical architecture model. Architectural design, logical design, and physical design are the most important aspects of system design. The architectural design of a system focuses on the architecture of the system, which describes the structure, behaviour, and other viewpoints and analyses of that system.

The objective of the System Design is to enhance the system architecture by giving knowledge and data that is valuable and required for the system elements to be implemented. The process of generating, expressing, documenting, and communicating the reality of a system's architecture through a complete set of design characteristics stated in a manner suitable for execution is known as design definition.

3.1 Architectural Framework

Layered architecture is depicted in the diagram above. This model shows the interconnecting of sub-systems. This method divides the system into layers (or abstract machines), each of which provides a set of services. Supports the gradual development of subsystems across several layers. Only the neighbouring layer is affected when a layer interface changes. However, it is frequently unnatural to arrange systems in this manner.

The user interface with the system is the topmost layer. It contains the application's graphical user interface. It's also known as the application's frontend. The authentication service is included in the second layer. The model that predicts whether the image is original or doctored, as well as the graphical format, makes up the third layer. All of the layers in this design provide various functions and can be changed independently of one another.

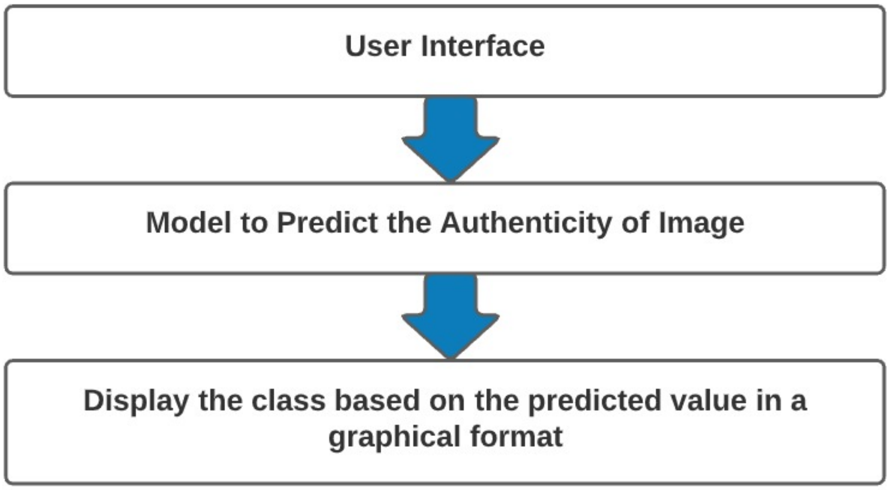

User Interface
Model to Predict the Authenticity of Image
Display the class based on the predicted value in a graphical format

Figure 3.1: Architecture of the system

3.2 Design Principles

A design principle is a set of rules that an organisation, project, or designer follows to make design decisions easier. They're meant to reflect a design's goals and get designers working in the same direction to produce consistent work.

3.2.1 ELA Method

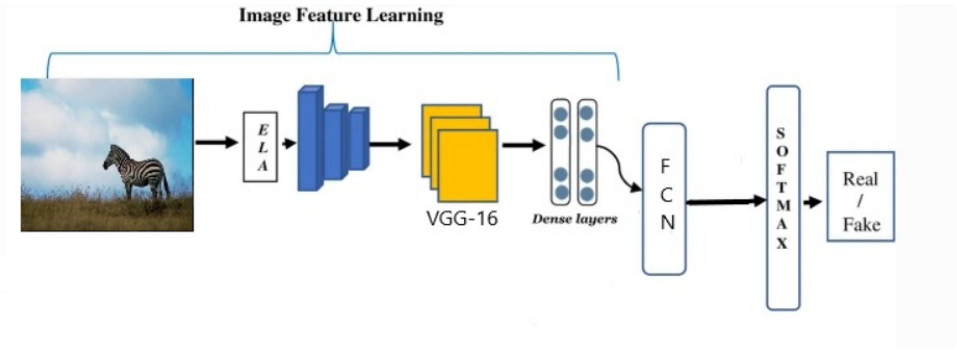


Figure 3.2: ELA Model

This graphic shows the full architecture of the proposed model. There are two components: Learning the Image feature—The fundamental characteristics of false pictures are learned.

Softmax is utilised as a classification device that uses the fused characteristics to classify the picture.

We have utilised the last particular CNN model named VGG-16 to extract the latent characteristics of the pictures.Although normal pictures in the dataset are utilised, their ELA images are used in the pre-processing step.ELA emphasises the compression features inside a picture.The application of any image processing filter helps improve the capacity for generalisation and speeds up the convergence of the deep learning networks.

In order to produce image embedding from the output of their third to their last caption, ELA pictures are transmitted to vgg-16 pretrained model and transfer knowledge from vgg16.In order to understand the picture properties, the image embeddedings are passed to two layers with completely linked thick layers. Here, functional vectors from the modes are learnt and the final Softmax classification classification is passed.The Softmax forecasts that false pictures are probable to appear.The Figure 3.2

Chapter 4

IMPLEMENTATION

This chapter provides a quick overview of the system's implementation details by describing each component and its code skeleton.

4.1 Proposed Methodology

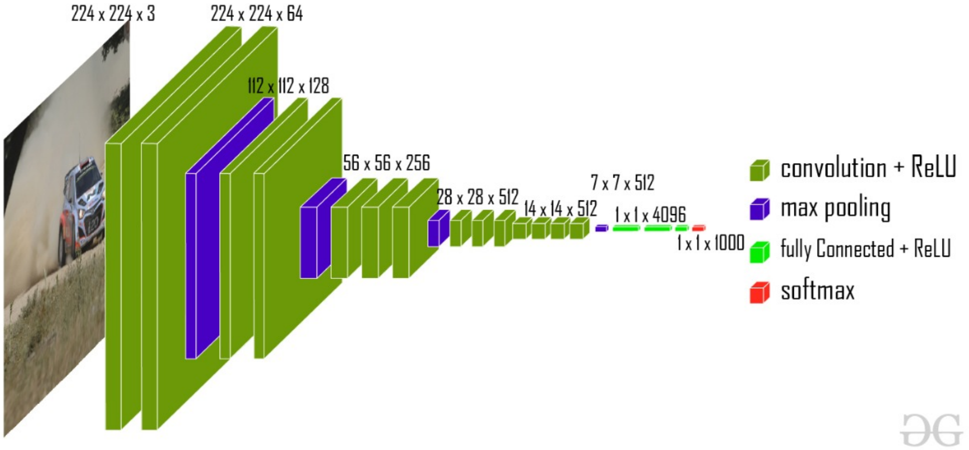


Figure 4.1: Design of proposed system

The model uses (224 x 224 x 3) as the input image size, which we set in the pre-processing section, with the 3 referring to the RGB colour space.

The first two layers of the model have the same padding and have 64 channels of 3*3 filter size. Then, after a stride (2, 2) max pool layer, two layers of convolution layers of 256 filter size and filter size (3, 3). This is followed by a stride (2, 2) max pooling layer, which is the same as the previous layer. Following that, there are two convolution layers with filter sizes of 3 and 3 and a 256 filter. Following that, there are two sets of three convolution layers, as well as a max pool layer. Each has 512 filters of the same size (3, 3) and padding.Following that, the image is sent to a stack of two convolution layers. The filters we utilise in these convolution and max pooling layers are 3*3 in size. After that, we receive a feature map (7 × 7 x 512). This is flattened and then connected to the fully connected layers, and then we added a softmax layer with 2 classes to classify it (Authentic or Doctored).

4.2 Modules

4.2.1 Module 1 : Data Preprocessing

We deleted certain unsupported image formats and transformed the rest of the images to a fixed size of (224 × 224) after obtaining the dataset.

4.2.2 Module 2 : Data Preparation with ELA

The preprocessed data is then delivered to the Error Level Analysis, where each image is processed in order to gain a clear knowledge of image parts with varied compression levels.

4.2.3 Module 3 : Data Preparation

Following the completion of the ELA, the preprocessed data is separated into two portions in an 80:20 ratio: train and validation datasets.

4.2.4 Module 4 : Model Definition and Training

The VGG-16 model, or picture classification model, was chosen because it had a 92.7% accuracy on the ImageNet dataset. This model will be used to distinguish between two types of information: authentic and doctored.

Adam Optimizer was used to optimise the learning rate. We used binary cross entropy as a loss function, which is a frequent choice for classification models. With a batch size of 32, we trained our model for 25 epochs.

Now we stored the model in.h5 file and tested it with several sample images to see if it can predict whether an image is doctored or legitimate, as well as the value of how doctored or authentic it is.

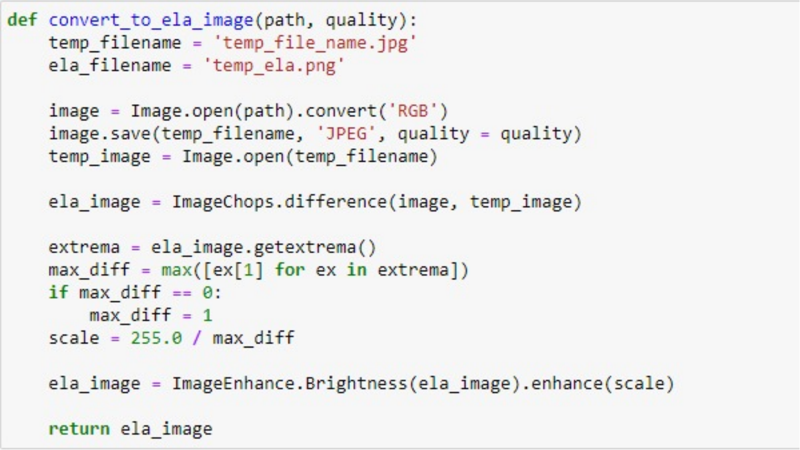
Later, we changed our model to tflite in order to deploy or use it in the creation of Android applications.

4.2.5 Module 5 : Android app

We used the tflite model we created previously, as well as the labels.txt file, to create an android app that accepts an image as input and determines if the image is authentic or doctored.

4.3 Code Snippets

4.3.1 Error level analysis

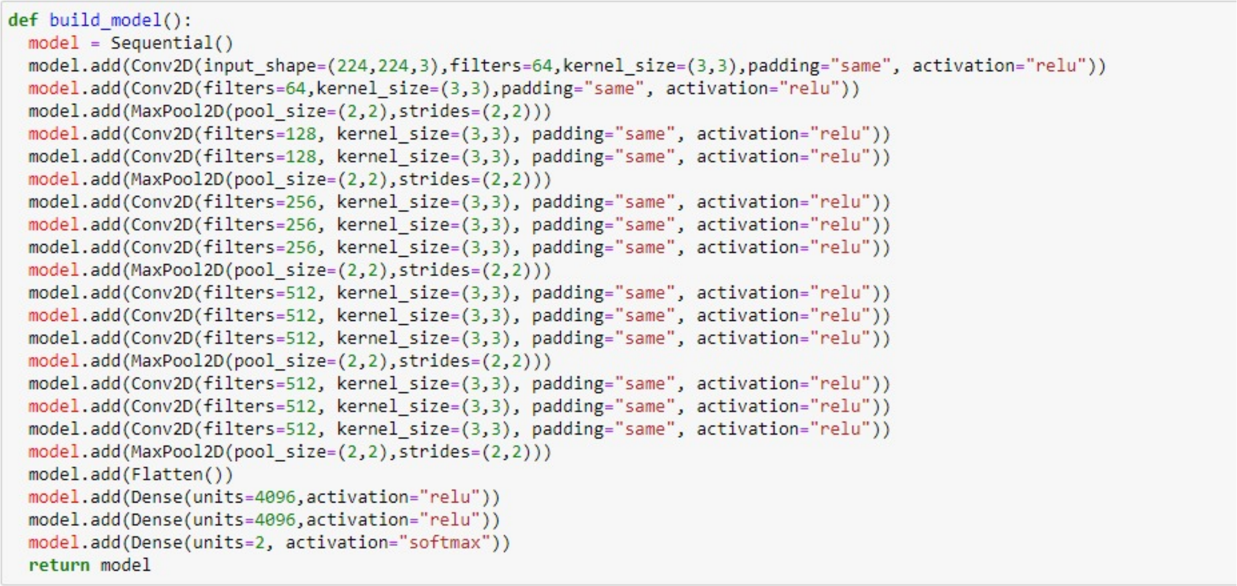


```
def convert_to_ela_image(path, quality):
    temp_filename = 'temp_file_name.jpg'
    ela_filename = 'temp_ela.png'

    image = Image.open(path).convert('RGB')
    image.save(temp_filename, 'JPEG', quality = quality)
    temp_image = Image.open(temp_filename)

    ela_image = ImageChops.difference(image, temp_image)

    extrema = ela_image.getextrema()
    max_diff = max([ex[1] for ex in extrema])
    if max_diff == 0:
        max_diff = 1
    scale = 255.0 / max_diff

    ela_image = ImageEnhance.Brightness(ela_image).enhance(scale)

    return ela_image
```

Figure 4.2: ELA

The function depicted in the diagram is used to provide a properly preprocessed image, which is accomplished using error level analysis so that we may obtain a clean image for future analysis.

4.3.2 Model build

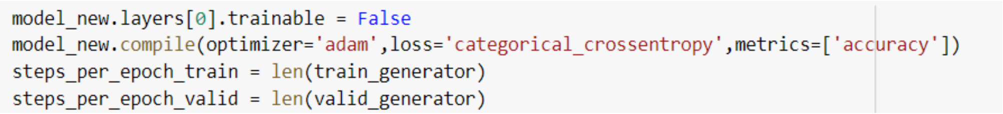


```
def build_model():
  model = Sequential()
  model.add(Conv2D(input_shape=(224,224,3),filters=64,kernel_size=(3,3),padding="same", activation="relu"))
  model.add(Conv2D(filters=64,kernel_size=(3,3),padding="same", activation="relu"))
  model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
  model.add(Conv2D(filters=128, kernel_size=(3,3), padding="same", activation="relu"))
  model.add(Conv2D(filters=128, kernel_size=(3,3), padding="same", activation="relu"))
  model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
  model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="relu"))
  model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="relu"))
  model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="relu"))
  model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
  model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
  model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
  model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
  model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
  model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
  model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
  model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="relu"))
  model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
  model.add(Flatten())
  model.add(Dense(units=4096,activation="relu"))
  model.add(Dense(units=4096,activation="relu"))
  model.add(Dense(units=2, activation="softmax"))
  return model
```

Figure 4.3: Building of Model

The code snippet describes the layers used to construct the model, as well as the configuration of each layer that is being processed.

4.3.3 Optimizer

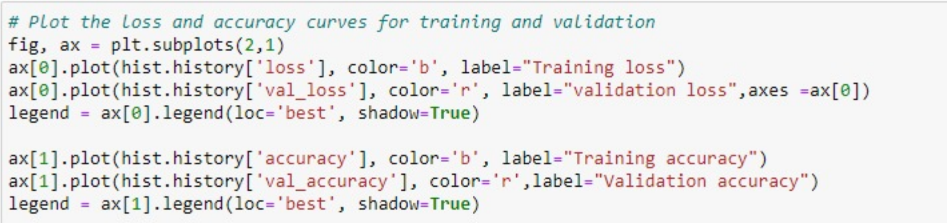


```
model_new.layers[0].trainable = False
model_new.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
steps_per_epoch_train = len(train_generator)
steps_per_epoch_valid = len(valid_generator)
```

Figure 4.4: adam optimizer

The optimizer in use, Adam Optimizer Adaptive Moment Estimation, is a gradient descent optimization technique, as seen in the graph. The method is quite efficient when dealing with large problems with a lot of data or parameters. It is quick and uses little memory.

4.3.4 ELA loss and accuracy

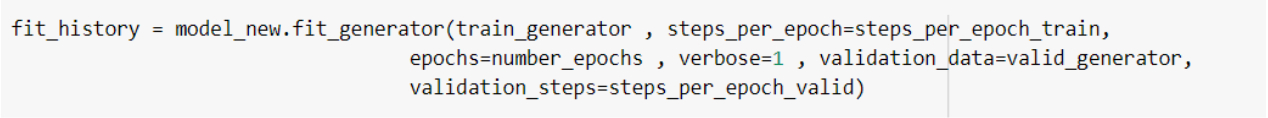


```
# Plot the loss and accuracy curves for training and validation
fig, ax = plt.subplots(2,1)
ax[0].plot(hist.history['loss'], color='b', label="Training loss")
ax[0].plot(hist.history['val_loss'], color='r', label="validation loss",axes =ax[0])
legend = ax[0].legend(loc='best', shadow=True)

ax[1].plot(hist.history['accuracy'], color='b', label="Training accuracy")
ax[1].plot(hist.history['val_accuracy'], color='r',label="Validation accuracy")
legend = ax[1].legend(loc='best', shadow=True)
```

Figure 4.5: loss and accuracy plot function

The function is defined to have both training and validation sets' loss and accuracy charts.

4.3.5 Model Training

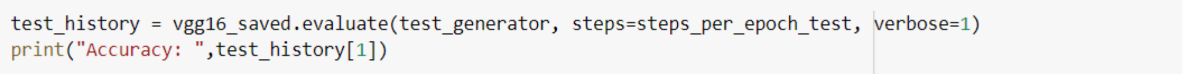


```
fit_history = model_new.fit_generator(train_generator , steps_per_epoch=steps_per_epoch_train,
                                      epochs=number_epochs , verbose=1 , validation_data=valid_generator,
                                      validation_steps=steps_per_epoch_valid)
```

Figure 4.6: Training of model

The VGG16 model is trained using the hist function, as illustrated in the image. The data is confirmed based on the number of epochs, and each verified step is processed with respect to the number of epoch steps.

4.3.6 Test set training

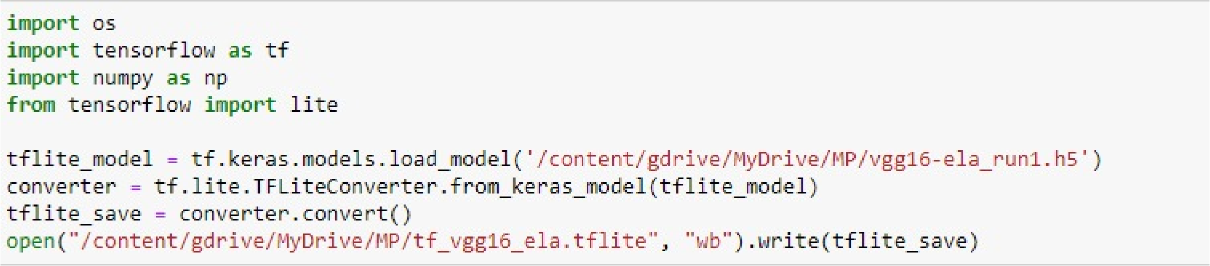


```
test_history = vgg16_saved.evaluate(test_generator, steps=steps_per_epoch_test, verbose=1)
print("Accuracy: ",test_history[1])
```

Figure 4.7: Training of test set

As illustrated in the diagram, the test set is trained using the test history function. Following the training of the test set, the accuracy of that test set is presented. The sole difference between test set training and model training is that test set training does not use epochs as inputs.

4.3.7 TFlite Generation



```
import os
import tensorflow as tf
import numpy as np
from tensorflow import lite

tflite_model = tf.keras.models.load_model('/content/gdrive/MyDrive/MP/vgg16-ela_run1.h5')
converter = tf.lite.TFLiteConverter.from_keras_model(tflite_model)
tflite_save = converter.convert()
open("/content/gdrive/MyDrive/MP/tf_vgg16_ela.tflite", "wb").write(tflite_save)
```

Figure 4.8: tflite function

The graphic depicts the tflite function, which is used to generate the tflite file, which must then be imported into the android app for use.

Chapter 5

RESULTS AND DISCUSSIONS

We created an Android software that takes an image and categorises it as either doctored or authorised. To do so, we employed the Error Level Analysis and VGG-16 models. Each image is handled in Error Level Analysis to acquire a thorough understanding of image sections with different compression levels. VGG16 is a convolutional network designed for image classification, whereas CNN is a neural network idea. The results of using the VGG16 model during training are as follows.

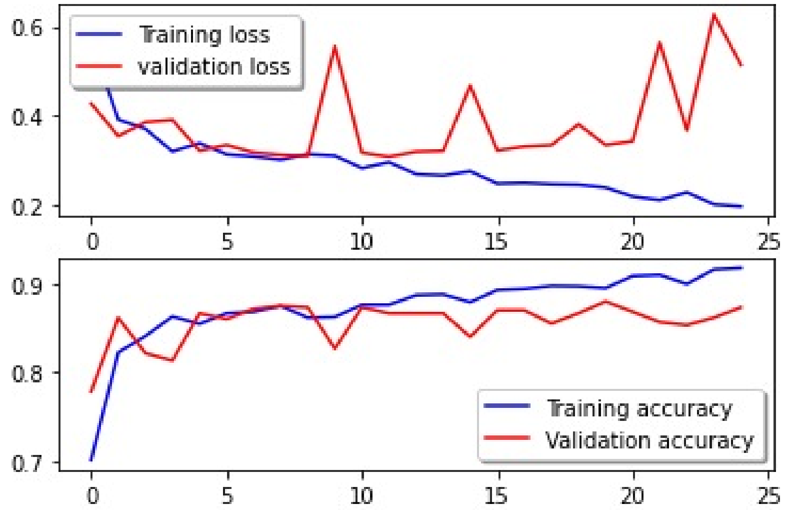
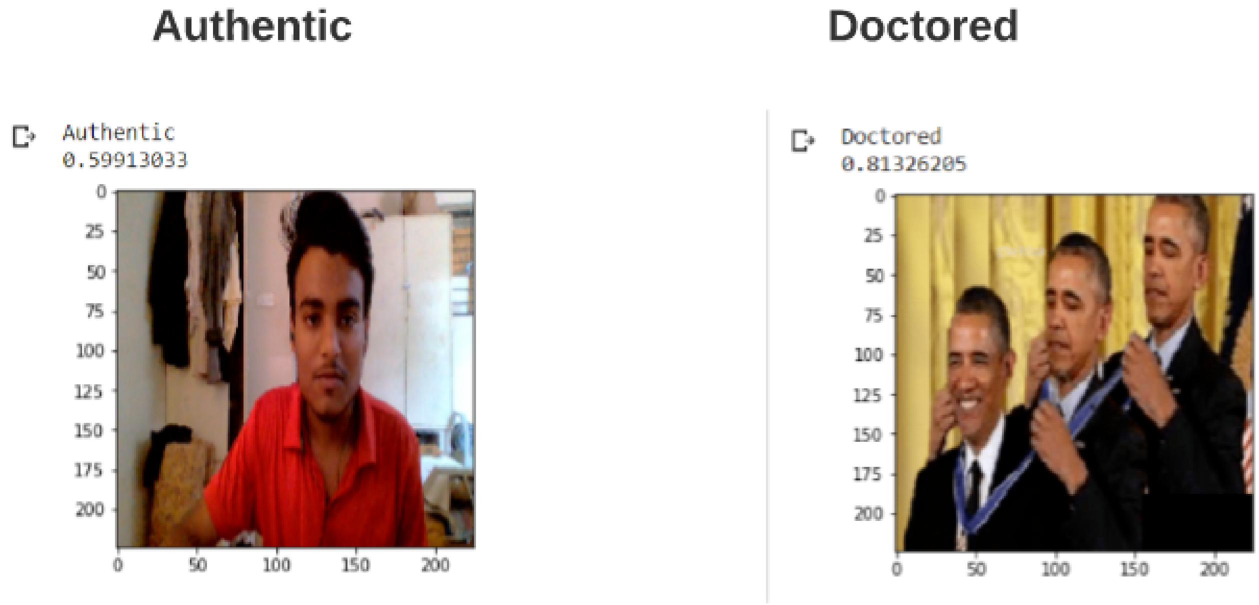


Figure 5.1: Graph of ELA

The model's performance is evaluated during training in the diagram above. Over the accuracy and epoch graph, the model accuracy and the amount of loss are shown. The curves for both training accuracy (blue) and validation accuracy (orange) are increasing as the number of epochs increases, as seen in the plot.

Furthermore, as the number of epochs rises, the train loss (green) and validation loss (red) plots decrease. Due to a lack of processing resources, we are only training our model for two epochs, resulting in an accuracy of 73.72% for the validation dataset and a loss of roughly 52.7%.If we run the model for more epochs on a high-performance computer, the accuracy will improve as the quantity of loss decreases as the number of epochs increases.

In the first plot above, the amount of loss for both train and validation data is presented. The curve for both training loss (blue) and epoch number (green) is reducing as the number of epochs increases, i.e. 20.11 percent, as can be seen in the plot. The graph varies and indicates a small rise above the original value of 51.42 percent when validation loss (red) happens. The accuracy of both the train and validation data is shown in the second plot. Both the training and validation accuracy curves (blue and red, respectively) expand as the number of epochs grows, i.e. 91.5 percent for training and 87.33 percent for validation dataset.



```
Authentic
0.59913033
```

```
Doctored
0.81326205
```

Figure 5.2: Result

The following are some of the vgg16 model's test results. The findings acquired are more accurate than CNN due to its depth and number of completely connected nodes.

The image will be loaded into the Android application, which will train the dataset using a model and classify the image as doctored or legitimate, and present the classification results in the form of a bar graph.

5.1 Dataset

The CASIA-2 dataset has been used for our Deep Learning Model.The dataset has 7492 Authentic images , 5125 Tampered images and 5123 groundtruth images.The Figure 5.3

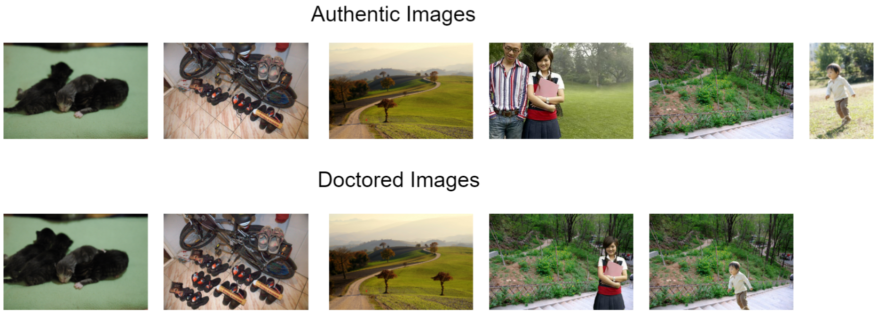


Figure 5.3: dataset samples

This image will be used as input test data for the model, and the result will be a bar chart with around 20% of the image doctored and 80% of the image as a real feature. This is what the software's output looks like. That this is a real image by comparing the largest fraction of this number.The Figure 5.4

We may deduce that this is a doctored image by comparing the greatest proportion of this figure, which has roughly 85% as doctored and 15% as real.This is what the software's output looks like. That this is a real image by comparing the largest fraction of this number.The Figure 5.5

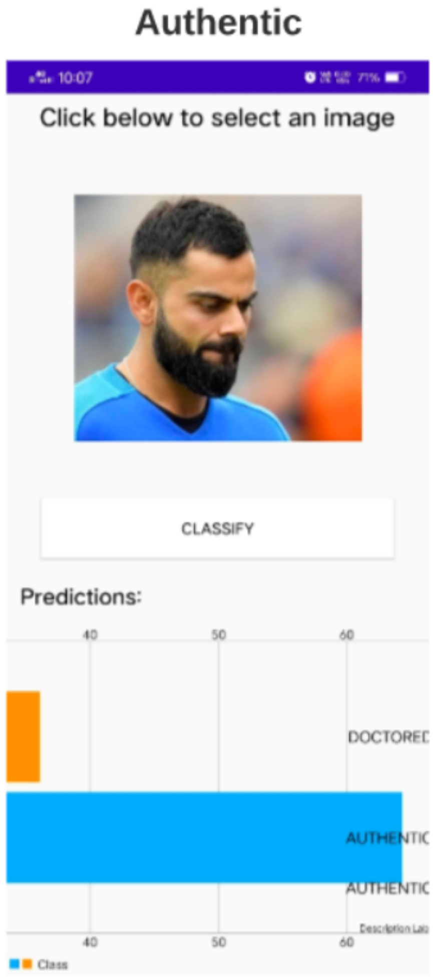
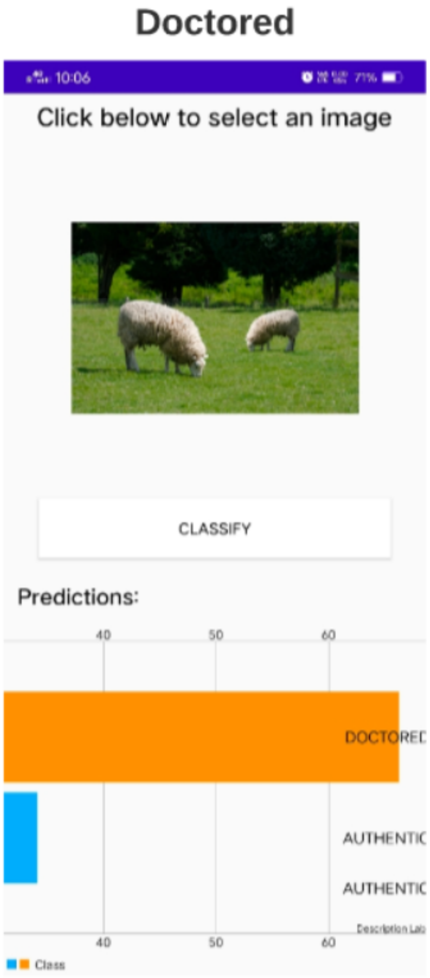


Figure 5.4: App Result of authentic image



Doctored

Click below to select an image

Figure 5.5: App Result of doctored image

Chapter 6

CONCLUSION AND FUTURE SCOPE OF THE WORK

6.1 Conclusion

We successfully constructed a Deep Learning-based classifier and an Android application in this project, which can recognise and classify any image into Authentic or Doctored. As a result, it performs the function of a Doctored Image Detection System (DIDS).

To categorise the photos, the current model employs a Deep Learning approach. The VGG-16 model is created and saved as a '.h5' file. The model's architecture, weights values, and compilation information are all contained in the h5 file. As a result, the trained model may be quickly deployed on any edge device or network interface for real-time picture classification.For this, we created an Android app that allows users to upload images and determine whether they are authentic or doctored. The model also indicates whether it is Authentic or Doctored by a percentage or a value.

We now trained the model on the CPU for 3 epochs, but with a high compute machine, we can train the model for more epochs, increasing the model's accuracy while lowering data loss throughout the training process.

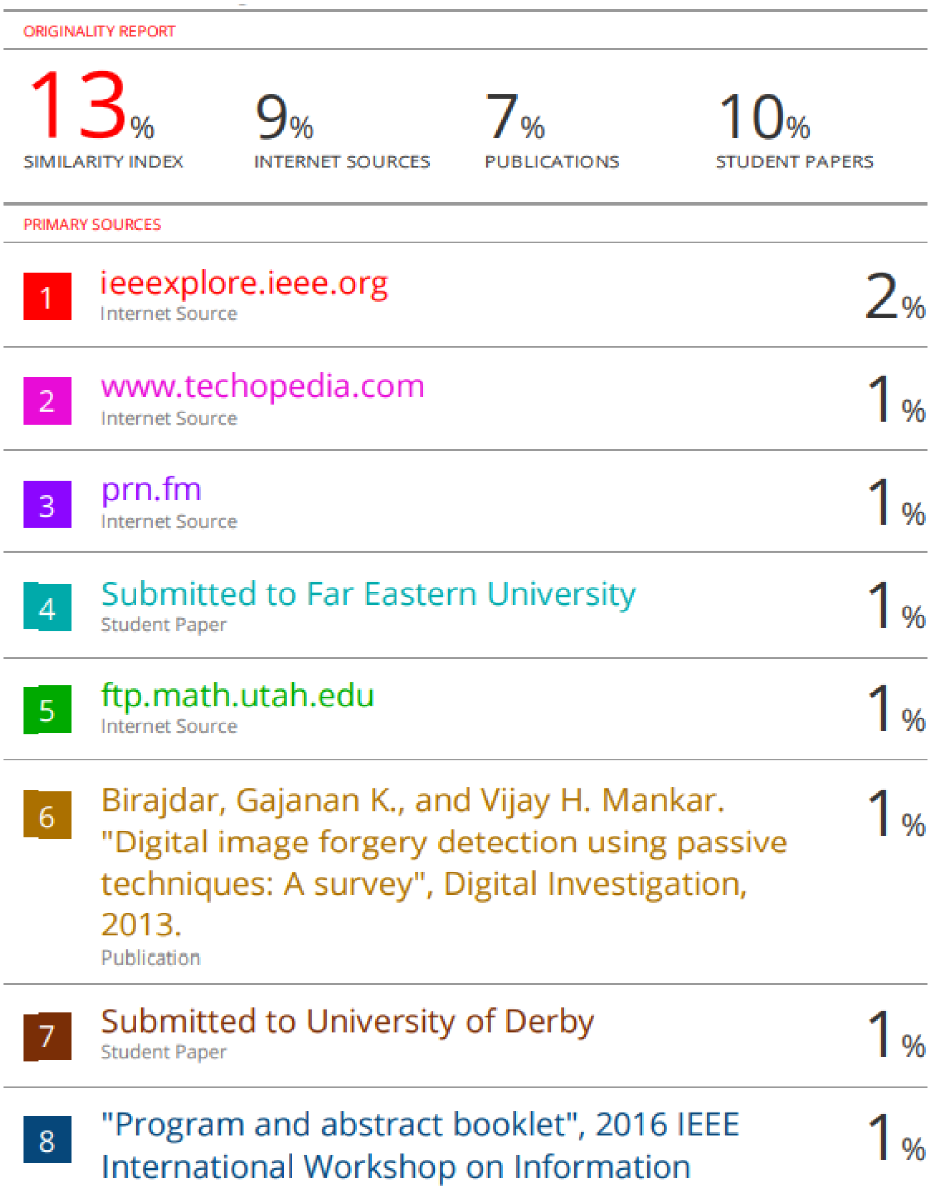
6.2 Future scope

The project can either be expanded to show the area that is being doctored, or it can be linked with well-known tools such as Adobe or Gimp as a tool to detect doctored photographs right away.

Project Title : Develop an Android Application to Detect Doctored Images.

Team No : D25.

Project Domain : Data Engineering.



ORIGINALITY REPORT

13%	9%	7%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ieeexplore.ieee.org Internet Source	2%
2	www.techopedia.com Internet Source	1 %
3	prn.fm Internet Source	1 %
4	Submitted to Far Eastern University Student Paper	1 %
5	ftp.math.utah.edu Internet Source	1 %
6	Birajdar, Gajanan K., and Vijay H. Mankar. "Digital image forgery detection using passive techniques: A survey", Digital Investigation, 2013. Publication	1 %
7	Submitted to University of Derby Student Paper	1 %
8	"Program and abstract booklet", 2016 IEEE International Workshop on Information	1 %

	Forensics and Security (WIFS), 2016 Publication	
9	Submitted to Universiti Teknologi MARA Student Paper	1 %
10	"Advances in Signal Processing and Intelligent Recognition Systems", Springer Science and Business Media LLC, 2016 Publication	1 %
11	Submitted to University of Bahrain Student Paper	1 %
12	expertsoftit.com Internet Source	1 %
13	Submitted to University of East London Student Paper	<1 %
14	Submitted to British University in Egypt Student Paper	<1 %
15	Submitted to Queen Mary and Westfield College Student Paper	<1 %
16	Yuecong Lai, Tianqiang Huang, Jing Lin, Henan Lu. "An improved block-based matching algorithm of copy-move forgery detection", Multimedia Tools and Applications, 2017 Publication	<1 %
17	Submitted to University of Greenwich Student Paper	<1 %

18	Submitted to University of Technology, Sydney Student Paper	<1 %
19	es.scribd.com Internet Source	<1 %
20	en.wikipedia.org Internet Source	<1 %
21	Birajdar, Gajanan K., and Vijay H. Mankar. "Blind method for rescaling detection and rescale factor estimation in digital images using periodic properties of interpolation", AEU - International Journal of Electronics and Communications, 2014. Publication	<1 %
22	Yuan Rao, Jiangqun Ni. "A deep learning approach to detection of splicing and copy-move forgeries in images", 2016 IEEE International Workshop on Information Forensics and Security (WIFS), 2016 Publication	<1 %
23	Vincent Christlein, Christian Riess, Elli Angelopoulou. "On rotation invariance in copy-move forgery detection", 2010 IEEE International Workshop on Information Forensics and Security, 2010 Publication	<1 %

Chapter 7

REFERENCES

1. U. Ananya and U. Mudenagudi, "Detection of doctored images using bispectral analysis," 2011 International Conference on Image Information Processing, 2011, pp. 1-6, doi: 10.1109/ICIIP.2011.6108975.

2. J. Ouyang, Y. Liu and M. Liao, "Copy-move forgery detection based on deep learning," 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 2017, pp. 1-5, doi: 10.1109/CISP-BMEI.2017.8301940.

3. Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," 2016 IEEE International Workshop on Information Forensics and Security (WIFS), 2016, pp. 1-6, doi: 10.1109/WIFS.2016.7823911.

4. Gajanan K. Birajdar, Vijay H. Mankar,Digital image forgery detection using passive techniques: A survey,Digital Investigation,Volume 10, Issue 3,2013,Pages 226-245,ISSN 1742-2876,https://doi.org/10.1016/j.diin.2013.04.007. (https://www.sciencedirect.com/science/arti

5. Huang, Hailing et al. “Detection of Copy-Move Forgery in Digital Images Using SIFT Algorithm.” 2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application 2 (2008): 272-276.

Appendix A

A.1 Glossary

- Adobe Photoshop

  Adobe Photoshop is software that is used for alter the image in raster way,image editing, graphic design, and digital art. It makes use of layering to allow for depth and flexibility in the design and editing process, as well as provide powerful editing tools, that when combined, are capable of just about anything.

- CNN

  A convolutional neural network (CNN) is a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data.CNN has their "neurons" arranged more like those of the frontal lobe, the area responsible for processing visual stimuli in humans and other animals.

- ELA

  Neal Krawetz created the concept of ELA, Error Level Analysis (ELA) permits identifying areas within an image that are at different compression levels.ELA highlights differences in the JPEG compression rate. With JPEG images, the entire picture should be at roughly the same level. If a section of the image is at a significantly different error level, then it likely indicates a digital modification.

- VGG16 Model

  The chosen model is the VGG-16 model more specifically a image classification model which gave an accuracy of 92.7% on the ImageNet dataset. This model will be used to classify 2 different classes, i.e. Authentic and Doctored.