DIGITAL IMAGE FORGERY DETECTION USING CONVOLUTION NEURAL NETWORK

Minor project report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

S.MADHU SUDHAN REDDY	**(18UECS0774)**
P.BALA ADITHYA	**(18UECS0109)**
B.VEERA RAGHAVA REDDY	**(18UECS0114)**

Under the guidance of
Mr.SARAVANAKUMAR S,M.E.,
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING
VEL TECH RANGARAJAN Dr.SAGUNTHALA R&D
INSTITUTE OF SCIENCE AND TECHNOLOGY
(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A Grade
CHENNAI 600 062, TAMILNADU, INDIA

June,2021

CERTIFICATE

It is certified that the work contained in the project report titled ”DIGITAL IMAGE FORGERY DETECTION USING CONVOLUTION NEURAL NETWORK” by ”S.MADHU SUDHAN REDDY (18UECS0774) P.BALA ADITHYA (18UECS0109) B.VEERA RAGHAVA REDDY (18UECS0114)” has been carried out under my/our supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor
Mr.Saravanakumar S
Assistant Professor
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr.Sagunthala R&D
Institute of Science and Technology
June,2021

Signature of Head of the Department
Dr. V. Srinivasa Rao
Professor & Head
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr.Sagunthala R&D
Institute of Science and Technology
June,2021

DECLARATION

We declare that this written submission represents my ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)
(S.MADHU SUHAN REDDY)
Date: / /

(Signature)
(P.BALA ADITHYA)
Date: / /

(Signature)
(B.VEERA RAGHAVA REDDY)
Date: / /

APPROVAL SHEET

This project report entitled ”DIGITAL IMAGE FORGERY DETECTION USING CONVOLUTION NEURAL NETWORK” by S.MADHU SUDHAN REDDY (18UECS0774), P.BALA ADITHYA (18UECS0109), B.VEERA RAGHAVA REDDY (18UECS0114) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners **Supervisor**

Mr.Saravanakumar S,M.E.,

Date: / /

Place: Chennai

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO). DSc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Dean & Head, Department of Computer Science and Engineering Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We also take this opportunity to express a deep sense of gratitude to Our Internal Supervisor **Mr . SARAVANAKUMAR S ; M . E** for her cordial support, valuable information and guidance, he helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. C. SHYAMALA KUMARI, M.E., Ms.S.FLORENCE, M.Tech.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

S.MADHU SUDAN REDDY	**(18UECS0774)**
P.BALA ADITHYA	**(18UECS0109)**
B.VEERA RAGHAVA REDDY	**(18UECS0114)**

ABSTRACT

In present days images are being forged by using various softwares like adobe photoshop , canva , kine master , quicksort etc... With this project we are mainly focusing in images that are forged using image splicing techniques . Image forgeries is identified and detected by using convolutional neural network, and semantic segmentation. To teach the algorithm with two classes using deep convolutional neural network transfer learning approach is employed.The proposed system consists of two phases ; detection phase which is used to detect whether the image is forged or not , localization phase which is used to locate the forged part on the image . In this proposed system VGG16 and U-Net are the two algorithms used . VGG16 is a 16 layered CNN architecture it is used to classify whether image's pixels having any forgery or not , U-Net and VGG16 are being used separately as two different methods so that the images that are classified with color pixel label are trained using semantic segmentation to localize forged pixels on the image.Out of the two methods used for localization VGG16 works well with good results.

Keywords: Convolutional Neural Network , Semantic Segmentation , Pixels , Image Forgery , Image Splicing.

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

CNN	Convolutional Neural Network
VGG	Visual Geometry Group

TABLE OF CONTENTS

Page.No

Chapter 1

INTRODUCTION

1.1 Introduction

Images became an important part of our life in the current days with the common usage of smart acquisition devices like cameras and smartphones, then the ease of sharing over the internet. In parallel to the current fact, the number of image-processing software increased and are accessible, specified anyone may easily modify and share images online. Several techniques have emerged and one among them is image splicing which is foremost commonly used manipulations.In this technique of forgery an image is forged by implanting some other pixels of another image.Some of these manipulations are difficult to detect for non-expert user. Moreover, some manipulated images aim at delivering misleading information which could be a threat to society. Therefore , the forensic researcher community focused on developing tools which validate the integrity of a picture. Many approaches detecting the image authenticity and assessing the integrity of the photographs are proposed.

VGG 16 is a convolutional neural network that is 16 layers deep. The model is loaded with set of weights pre-trained on ImageNet .The model achieves an accuracy of 92.7 percent in ImageNet , which has a dataset of over 14 million labeled high-resolution images that belong to more than 1000 classes and 22,000 categories. The default image size for vgg16 is 224-by-244 pixels, with three channels of RGB image. It has convolution layers of 3-by-3 filter with a stride 1 and maxpool layer of 2-by-2 filter of stride 2 . Initially image is passed through a sequence of convolutional layers after which Spatial pooling is done by five max-pooling layers which are followed by convolutional layers. Then 3 fully connected dense layers follow a stack of convolutional layers. The first two layers have 4096 channels each whereas the third layer has 1000 channels one for each class . The configuration of these fully connected layers is the same in all networks.

U-Net is an architecture that is used for fast and precise image segmentation.The architecture is u-shaped which justifies the name .It consists of an expansive path and a contracting path and the bottleneck.The contraction section is made of several contraction blocks.Each block takes two 3 x 3 convolutinal layers followed by 2 x 2 max pooling.After each block the feature maps or kernals gets doubled.This is done so that architecture can learn the complex structures effectively .In contrast the expansive section is made up of several expansion blocks . Each block takes two 3 x 3 convolutinal layers followed by 2 x 2 unsampling layers.After each block the feature maps or kernals gets halved to maintain the symmetry . This is done to reconstruct the features that are learnt while contracting the image . The number of contraction blocks is same as the number of expansion blocks.The bottom most layer mediates between the expansion layer and the contraction layer .

1.2 Aim of the project

Digital images tampering has been made easy with widely available image editing softwares.The objective of this project is to identify the photographs that are forged or tampered and to localize the forged or tampered part on the image.

1.3 Project Domain

Image Processing

1.4 Scope of the Project

The scope of this project focuses on proposing a Image Splicing technique of image forgery , with Convolution Neural Network and checking its performance on accurately detecting tampered images and localizing the tampered part on the image .

1.5 Methodology

Initially VGG16 is trained with a dataset which has some pictures that belong to different classes and categories such that it should detect whether the image is forged or not .We then trained U-Net and two layers of VGG16 with the forged images and their mask images for localization. If the image is forged ,then it predicts the output in the form of a black and white image where the forged part is depicted in white colour and then we localize the part of forgery on the image which is depicted in white colour in the output.

Chapter 2

LITERATURE REVIEW

N.Hema Rajini et al , proposed Image Forgery Detection using Convolution Neural Network(CNN).This journal develops a model for detecting splicing and copy-move forgery concurrently on the similar dataset of CASIA v1.0 as well as CASIA v2.0 . The presented model will decide the presence of manipulated image among the provided images.When the image is found to be manipulated , CNN is employed for classifying the image into splicing or copy-move forgery.The major focus of copy-move forgery is to search the same area exit in input images , whereas finding the inconsistency feature in splicing detection . CNN accurately classifies with an accuracy of 99.03 whereas a maximum accuracy of 99.11 is attained for the spliced images [1] .

S.Prayla Shyry , Saranya Meka , Mahitha Moganti et al ; proposed Digital Image Forgery Detection.Image forgery detection which plays the role for replacement,insertion and removal of objects.SIFT algorithm technique is applied to mark the objects in the image.They used the SIFT algorithm where a whole image is scanned and from the scanned image objects are marked.To classify this type of objects techniques like block-based Key Point based techniques,shift key point can be used.Key-point based methods can distinguish foreground to background.Though the accuracy of detecting forgery images in image using traditional methods is attained to certain level,improvement in existing techniques is required for better accuracy . Combination of machine learning algorithm could be a better option to yields accuracy.In future , SVM(Support Vector Classifier) can be used for better results [2] .

Kanchan Jha , Kshitija Shirwadkar , Tejaswi Narvekar , Isha Kothari , Salabha Jacob et al ; proposed Digital Image Forgery Detection.Digital images are foremost important source of information.One of the main purpose of image forgery is to conceal an object in the image.The second purpose may be to replicate some object in an image.The copy-move forgery introduces a correlation between the original image object and the pasted one.This correlation can be used as a basis for a successful detection of this type of forgery.In this journal,a strategy that can efficiently detect and localise duplicated regions in an image developed from the simplest and seemingly obvious techniques.A good method for detecting copy-move forgery is to verify if a set of blocks of pixels in a region of the image matches with another in a different region of the image.The Exact Match Algorithm was implemented using MATLAB software and Duplicate regions found by matching blocks found in the forged image [3] .

B.Santhosh Kumar , S.Karthi , K.Karthika , Rajan Cristin et al ; proposed Systematic Study of Image Forgery Detection.The need for the image forgery detection is increasing because of the threatening situation offered by the sophisticated image modification tools which diminish the credibility and authenticity of the original image.At present different types of detection algorithms exist,but the general structure appears same with changes in concepts for detection with fake images.Pixel based techniques identify the forgery in the original image by analysing the pixels constituting the image.Format based techniques detect the forgery in images based on the changes in the image format.The shadow based detection is the most convenient forgery detection.Because,by analysing the shadow properties of the image,the tampering operation performed for the manipulation can be recognised.This results in revising the old saying “A picture is worth a thousand words” to “A picture unworthy a thousand true words” [4] .

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

In the existing system it is only able detect whether the image is forged or not. It does not show or localise the part of forgery on the image.And also the accuracy levels of the existing system are too low to determine .

3.2 Proposed System

This proposed system is mainly focusing on image splicing technique where we project a technique with two phases . In the first phase VGG16 is used to detect whether the image is forged or not and in the second phase U-Net and the first two layers of VGG16 are used to show or localize the forged part in the image.

Advantages

1 . It is efficient for real time applications like a forensic tool while investigating crimes involving manipulation of images .
2 . Forged images with a little variations in pixels can be detected .

3.3 Feasibility Study

Feasibility studies are handled within tight constraints of time and normally given in a written or oral feasibility report . We have taken a week in feasibility study with my project members . The contents and recommendations of the feasibility study helped as a sound basis for deciding how to precede with the project. It helped in taking decisions such as which software to use . This study investigated the feasibility of Image forgery detection by using convolutional neural networks . To find the best result and high accuracy we used VGG16 and U-Net algorithms . Thus we concluded the out model is far better than when compared to the existing systems of this project.

3.3.1 Economic Feasibility

Among the information contained in feasibility study cost benefit analysis is the most important aspect. That is, an assessment of economic estimation for computer-based system. Cost-benefit analysis decides the cost for development and weights them against tangible and intangible benefits in the system.

3.3.2 Technical Feasibility

Technical analysis evaluates technical merits and demerits of the system at the same time collecting additional information about productivity, maintainability,reliability and performance. In some cases, a small research and design is included in system analysis .

3.3.3 Social Feasibility

Operational feasibility measures how well the solution performs in the organization and how will end-user management feel about the system . The Proposed system allows to identify the forged image and localize it . The Classification process will also become faster with the use of various different techniques. So it is feasible to implement the system.

3.4 System Specification

3.4.1 Hardware Specification

- Laptop or desktop
- Ram consisting of 8GB(min) or more
- Available space should be more than 10GB
- Version of the processor should be i5 or more

3.4.2 Software Specification

- Any operating system should be loaded
- Python-3.7.X(IDLE) version should be installed
- Pip should be installed
- Using pip install required modules i.e., tensorflow-2.2.0 , keras-2.4.3, matplotlib, cv2, tkinter, pandas, numpy, PIL, OS.

3.4.3 Standards and Policies

Standards and Policies are the compulsions and limitations in any type of project . In the proposed project the user should maintain all the requirements in order to achieve the proper output . The process have to be followed in the mentioned sequence or else it would raise to errors.

Chapter 4

MODULE DESCRIPTION

4.1 General Architecture

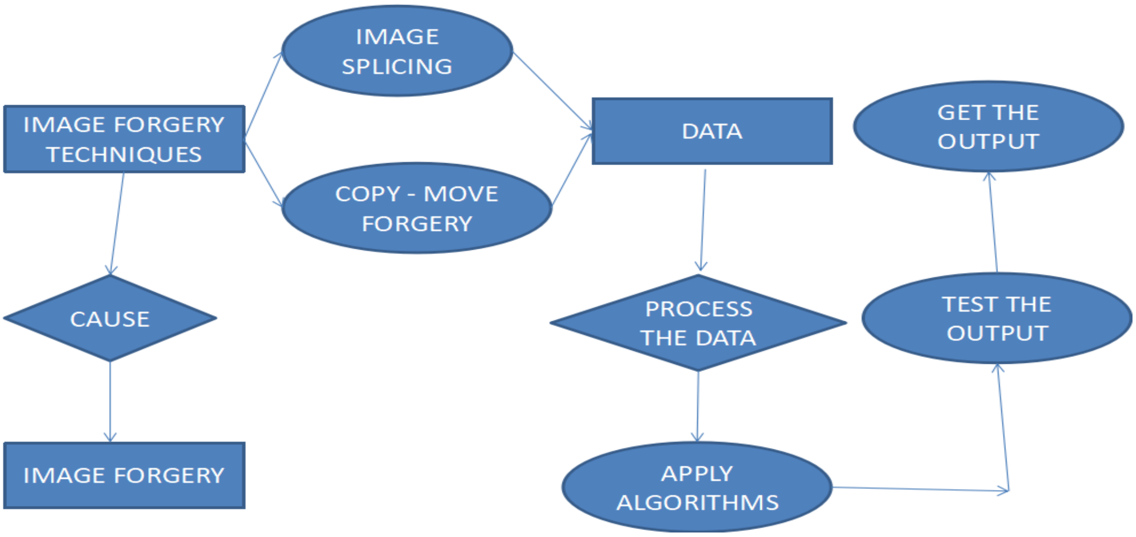


Figure 4.1: **Architecture Diagram**

Description

Image forgery is caused by different forgery techniques like image splicing , copy-move etc..To detect this forgery firstly dataset of different images is collected and processed and trained . When the input images is given it is trained and the algorithm is applied to get the output , the output is them tested as shown in figure 4.1 .

4.2 Design Phase

4.2.1 Data Flow Diagram

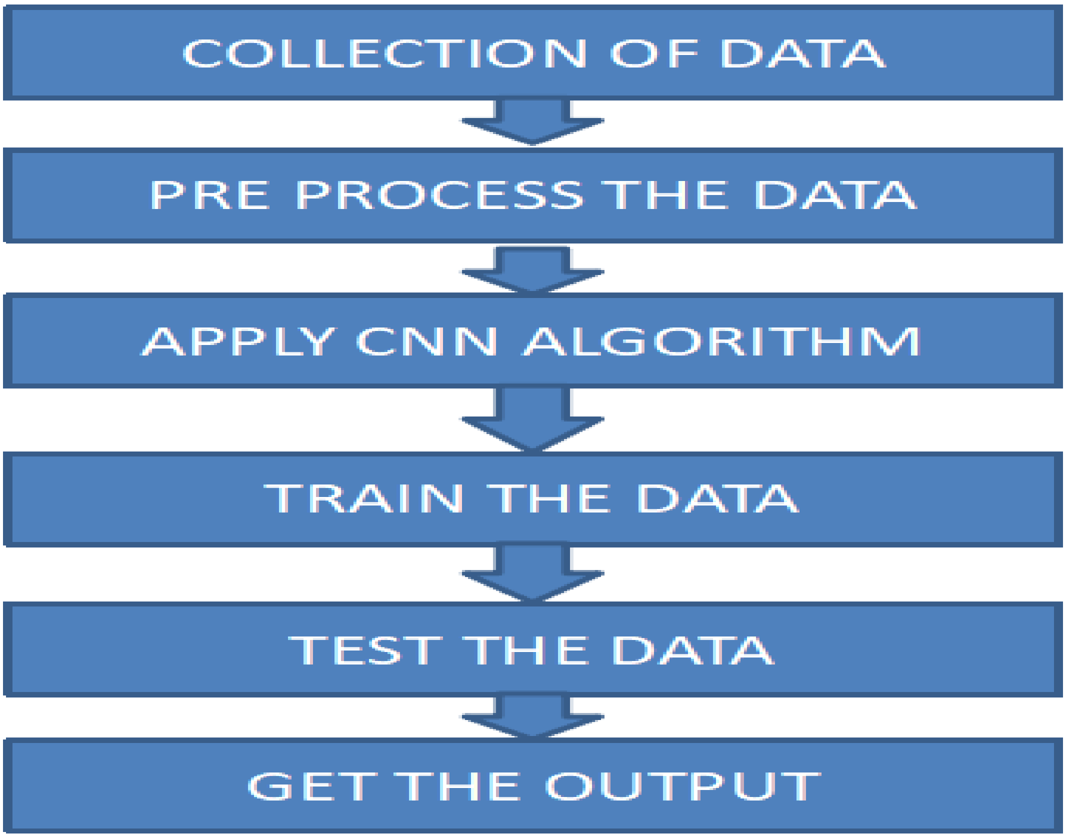


Figure 4.2: **Data Flow Diagram**

Description

It is the step by step process or the different stages present in the project as shown in figure 4.2. Here firstly the dataset is collected . After collection it is then preprocessed , After this CNN algorithm is applied , then the dataset is trained and tested to get the output .

4.2.2 UML Diagram

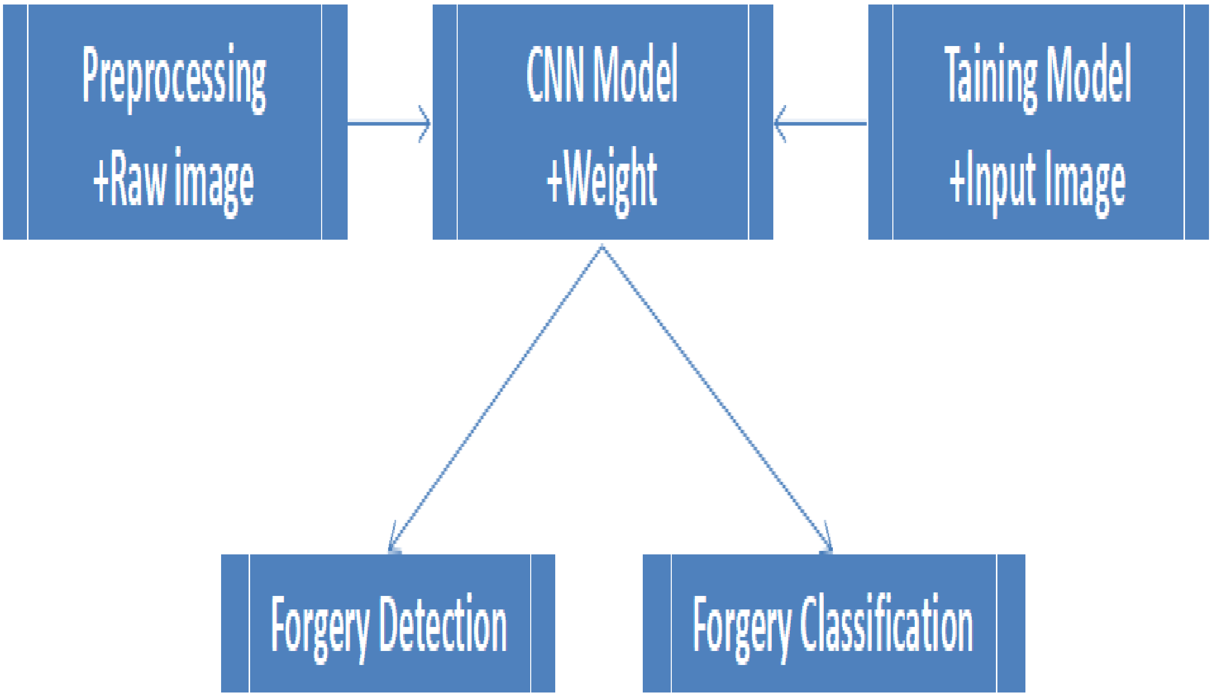


Figure 4.3: **UML Diagram**

Description

UML diagram is drawn to visualize the entire system at a glance . It represents the classes, actors, roles, actions, etc., in the proposed system .

4.2.3 Use Case Diagram

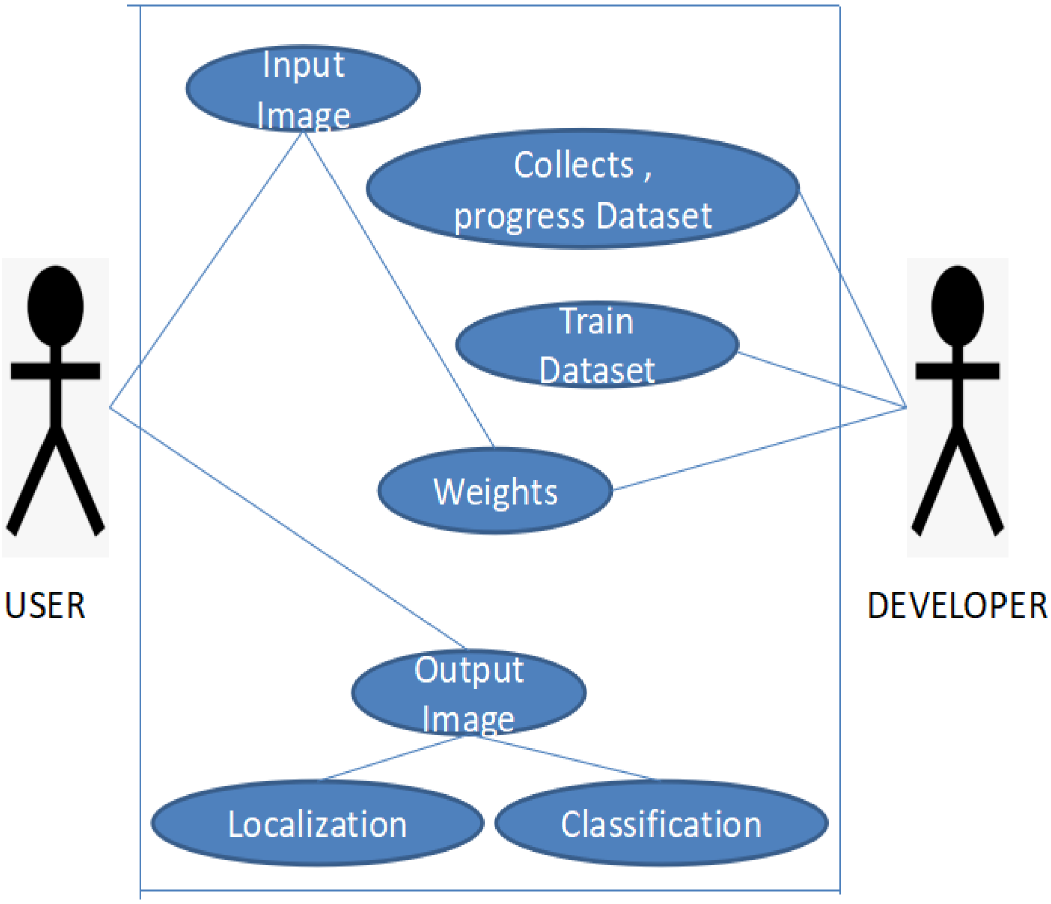


Figure 4.4: **Use Case Diagram**

Description

In the above shown figure 4.4 , the developer collects and processes the data and trains it . The user then gives an input image it is then trained and the trained results are stored in weights . After which the algorithm is applied to get the output of classification and localization of the forged part on the given input image .

4.2.4 Class Diagram

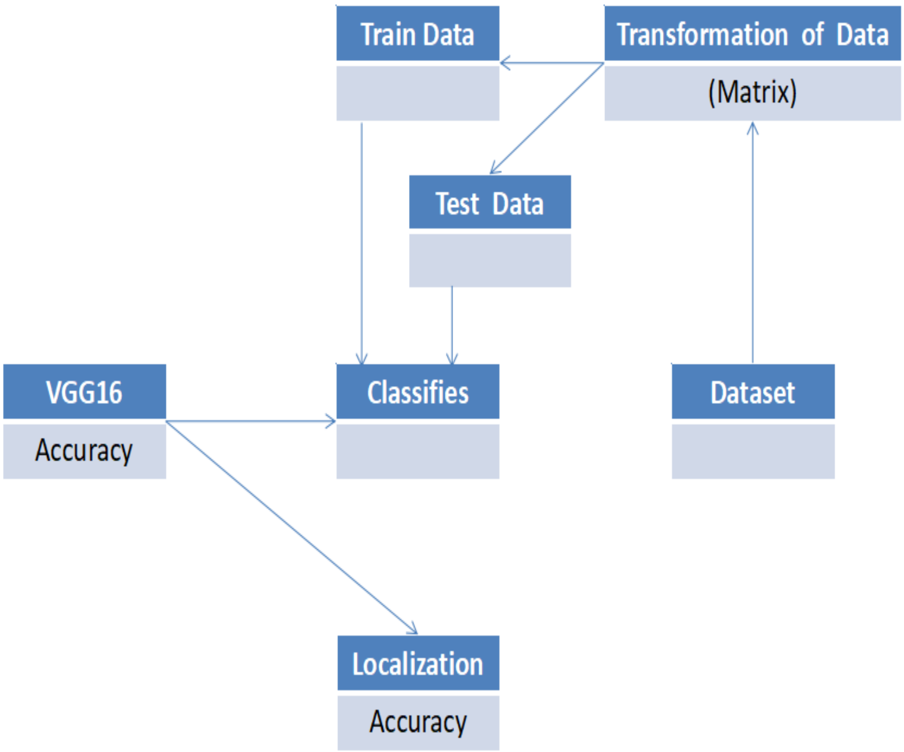


Figure 4.5: **Class Diagram**

Description

Class Diagram is used to describe the structure of the system and is one of the main building block of object oriented modelling. It is used for visualizing, describing and document the different aspects of the application.

4.2.5 Sequence Diagram

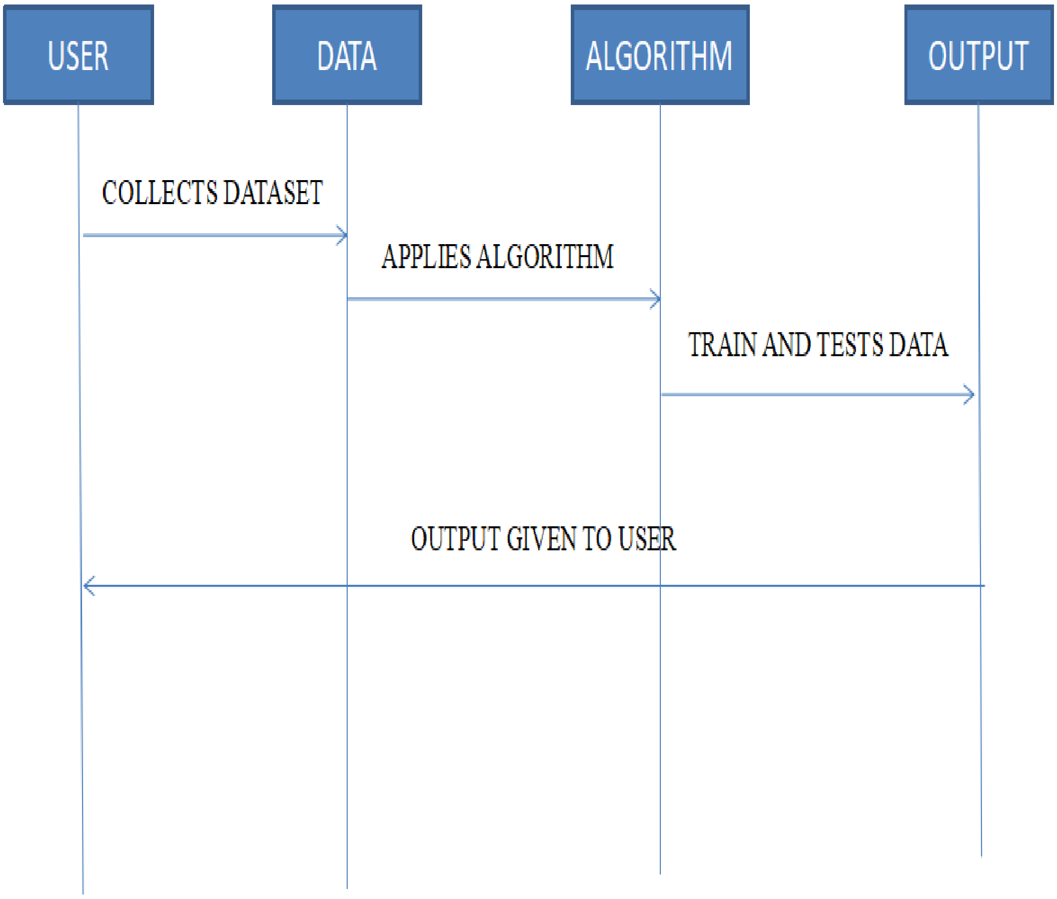


Figure 4.6: **Sequence Diagram**

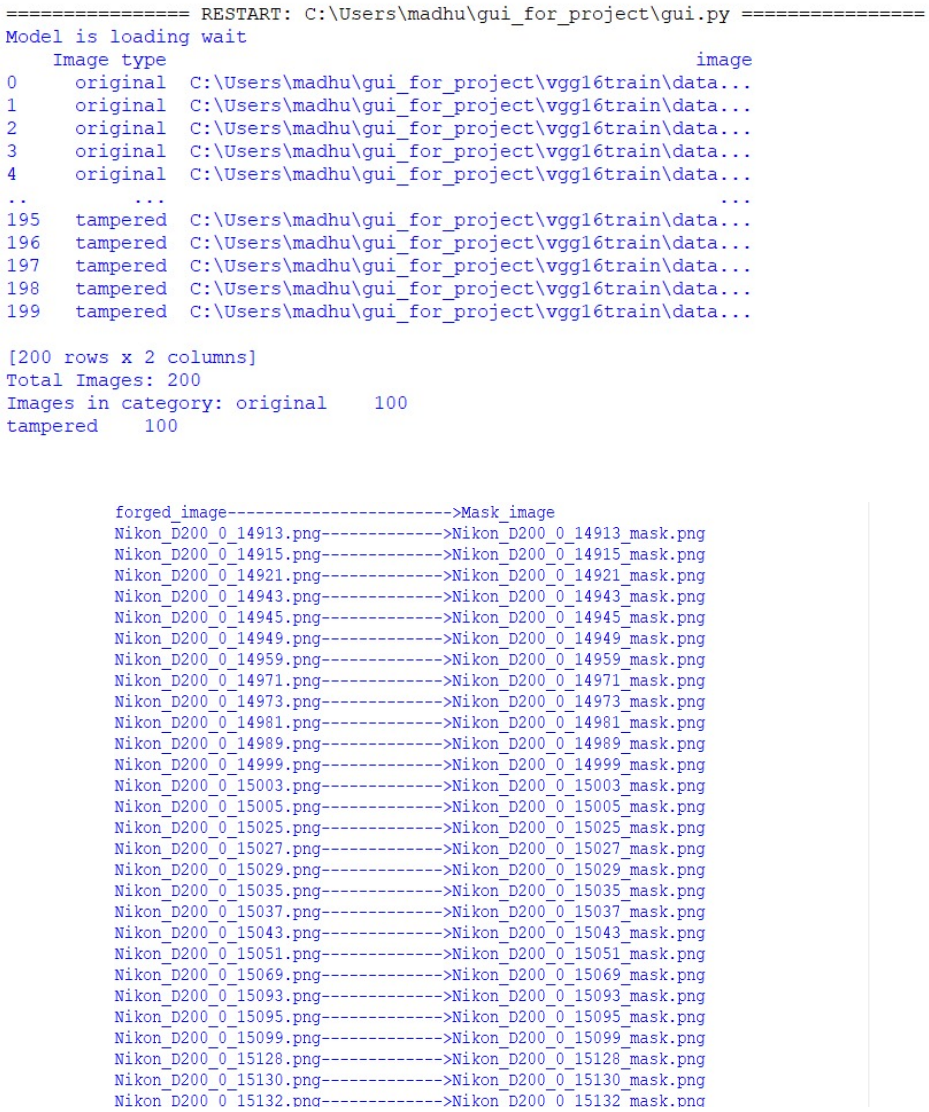
Description

This image depicts the sequence of steps that are encountered while doing this project . User collects the data , applies the algorithm , train and tests the data . After this he gives an input image to get the required output .

4.3 Module Description

4.3.1 Collection of Data

The initial stage for any analysis or classification using machine learning algorithms is to collect the data. There are various sources for data collection in the internet and some different organisations. The only thing we must ensure is that the data have the sufficient information enough to make the classification. While collecting information we should make sure it is 100% legit or not . Figure 4.7 shows the collected dataset .



```
================ RESTART: C:\Users\madhu\gui_for_project\gui.py ================
Model is loading wait
    Image type                                             image
0     original  C:\Users\madhu\gui_for_project\vgg16train\data...
1     original  C:\Users\madhu\gui_for_project\vgg16train\data...
2     original  C:\Users\madhu\gui_for_project\vgg16train\data...
3     original  C:\Users\madhu\gui_for_project\vgg16train\data...
4     original  C:\Users\madhu\gui_for_project\vgg16train\data...
..         ...                                               ...
195   tampered  C:\Users\madhu\gui_for_project\vgg16train\data...
196   tampered  C:\Users\madhu\gui_for_project\vgg16train\data...
197   tampered  C:\Users\madhu\gui_for_project\vgg16train\data...
198   tampered  C:\Users\madhu\gui_for_project\vgg16train\data...
199   tampered  C:\Users\madhu\gui_for_project\vgg16train\data...

[200 rows x 2 columns]
Total Images: 200
Images in category: original    100
tampered    100
```

```
forged_image------------------------>Mask_image
Nikon_D200_0_14913.png------------->Nikon_D200_0_14913_mask.png
Nikon_D200_0_14915.png------------->Nikon_D200_0_14915_mask.png
Nikon_D200_0_14921.png------------->Nikon_D200_0_14921_mask.png
Nikon_D200_0_14943.png------------->Nikon_D200_0_14943_mask.png
Nikon_D200_0_14945.png------------->Nikon_D200_0_14945_mask.png
Nikon_D200_0_14949.png------------->Nikon_D200_0_14949_mask.png
Nikon_D200_0_14959.png------------->Nikon_D200_0_14959_mask.png
Nikon_D200_0_14971.png------------->Nikon_D200_0_14971_mask.png
Nikon_D200_0_14973.png------------->Nikon_D200_0_14973_mask.png
Nikon_D200_0_14981.png------------->Nikon_D200_0_14981_mask.png
Nikon_D200_0_14989.png------------->Nikon_D200_0_14989_mask.png
Nikon_D200_0_14999.png------------->Nikon_D200_0_14999_mask.png
Nikon_D200_0_15003.png------------->Nikon_D200_0_15003_mask.png
Nikon_D200_0_15005.png------------->Nikon_D200_0_15005_mask.png
Nikon_D200_0_15025.png------------->Nikon_D200_0_15025_mask.png
Nikon_D200_0_15027.png------------->Nikon_D200_0_15027_mask.png
Nikon_D200_0_15029.png------------->Nikon_D200_0_15029_mask.png
Nikon_D200_0_15035.png------------->Nikon_D200_0_15035_mask.png
Nikon_D200_0_15037.png------------->Nikon_D200_0_15037_mask.png
Nikon_D200_0_15043.png------------->Nikon_D200_0_15043_mask.png
Nikon_D200_0_15051.png------------->Nikon_D200_0_15051_mask.png
Nikon_D200_0_15069.png------------->Nikon_D200_0_15069_mask.png
Nikon_D200_0_15093.png------------->Nikon_D200_0_15093_mask.png
Nikon_D200_0_15095.png------------->Nikon_D200_0_15095_mask.png
Nikon_D200_0_15099.png------------->Nikon_D200_0_15099_mask.png
Nikon_D200_0_15128.png------------->Nikon_D200_0_15128_mask.png
Nikon_D200_0_15130.png------------->Nikon_D200_0_15130_mask.png
Nikon_D200_0_15132.png------------->Nikon_D200_0_15132_mask.png
```

Figure 4.7: **Collected Dataset**

4.3.2 Visualization of Data

Make an analysis of data in the datasets provided by different sources and to classify them a original and tampered . The visualization of data can be of any type like graphs, maps, histogram,etc..

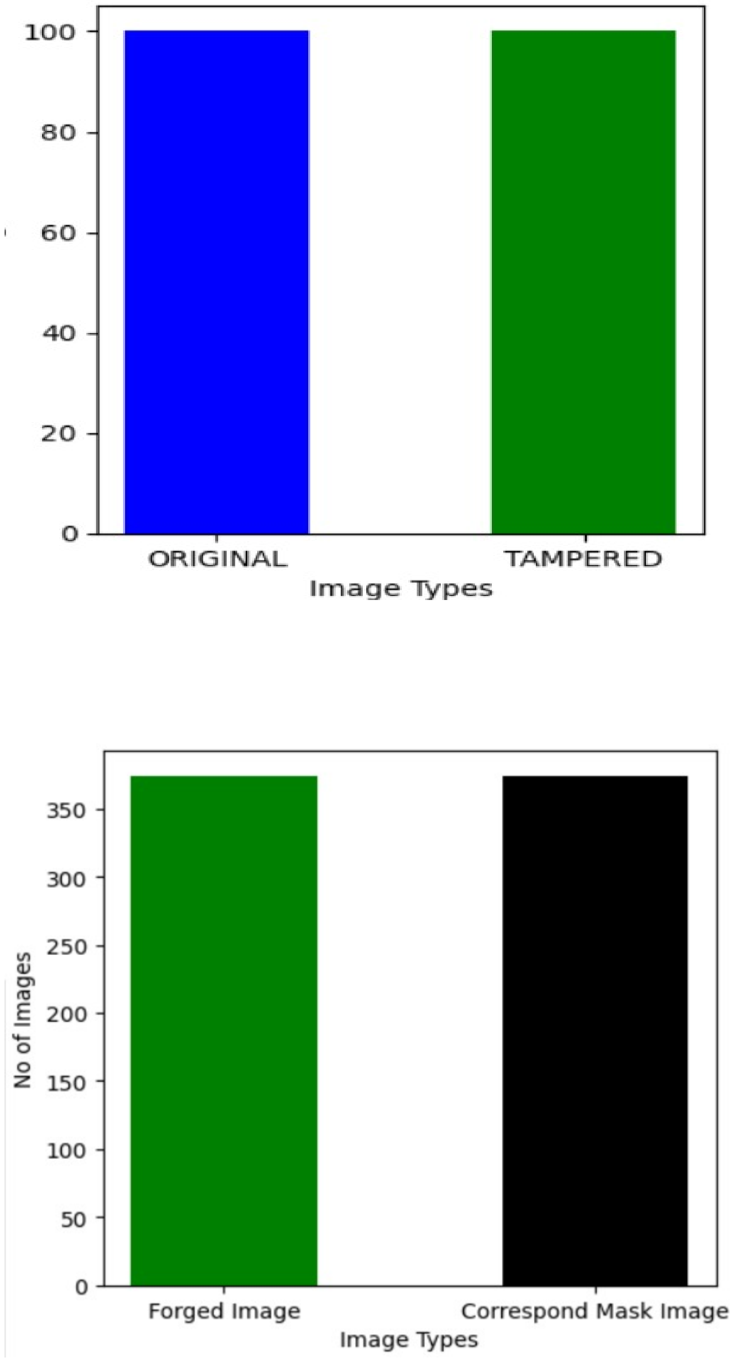
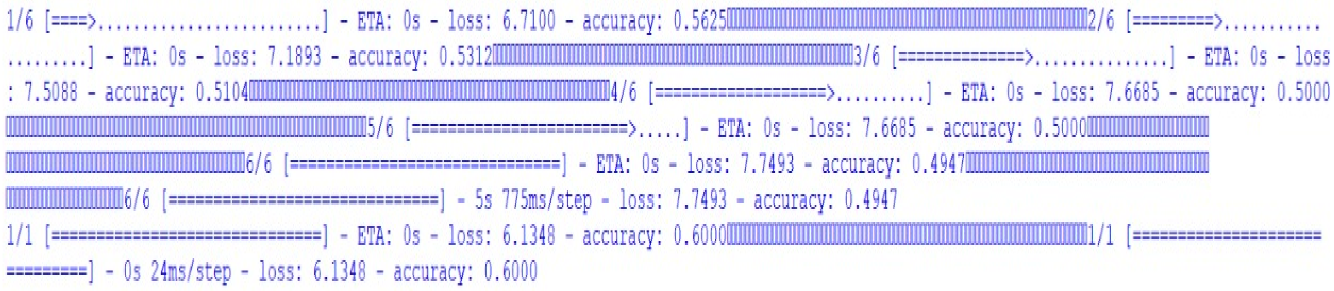


Figure 4.8: **Visualized Dataset**

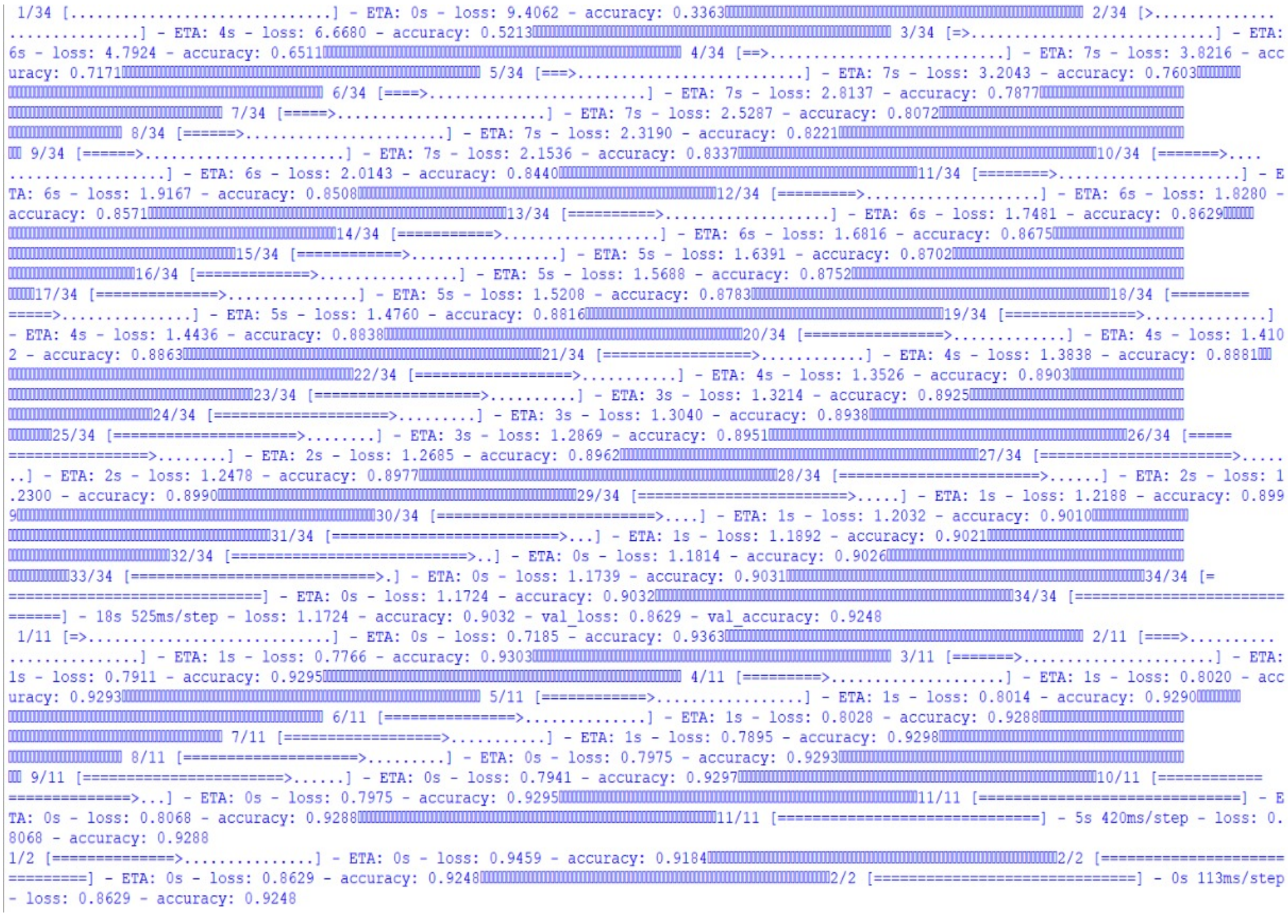
4.3.3 Training and Testing of Data

Training and Testing of the data is a very important step in the machine learning process because we know that we could not test all the data because it causes lots of trouble and could form errors in the prediction. So, in order to overcome all the errors we train and test the data in the ratio of 80:20. It gives 80% as training data and 20% as testing the data. This ensures the output to be accurate for a great result.



```
1/6 [====>........................] - ETA: 0s - loss: 6.7100 - accuracy: 0.56252/6 [=========>..........
.........] - ETA: 0s - loss: 7.1893 - accuracy: 0.53123/6 [==============>..............] - ETA: 0s - loss
: 7.5088 - accuracy: 0.51044/6 [===================>..........] - ETA: 0s - loss: 7.6685 - accuracy: 0.5000
5/6 [========================>.....] - ETA: 0s - loss: 7.6685 - accuracy: 0.5000
6/6 [==============================] - ETA: 0s - loss: 7.7493 - accuracy: 0.4947
6/6 [==============================] - 5s 775ms/step - loss: 7.7493 - accuracy: 0.4947
1/1 [==============================] - ETA: 0s - loss: 6.1348 - accuracy: 0.60001/1 [=====================
=========] - 0s 24ms/step - loss: 6.1348 - accuracy: 0.6000
```

Figure 4.9: **Train and Test for VGG16**



```
 1/34 [..............................] - ETA: 0s - loss: 9.4062 - accuracy: 0.3363 2/34 [>.............
..............] - ETA: 4s - loss: 6.6680 - accuracy: 0.5213 3/34 [=>............................] - ETA:
6s - loss: 4.7924 - accuracy: 0.6511 4/34 [==>...........................] - ETA: 7s - loss: 3.8216 - acc
uracy: 0.7171 5/34 [===>..........................] - ETA: 7s - loss: 3.2043 - accuracy: 0.7603
 6/34 [====>.........................] - ETA: 7s - loss: 2.8137 - accuracy: 0.7877
 7/34 [=====>........................] - ETA: 7s - loss: 2.5287 - accuracy: 0.8072
 8/34 [======>.......................] - ETA: 7s - loss: 2.3190 - accuracy: 0.8221
 9/34 [======>.......................] - ETA: 7s - loss: 2.1536 - accuracy: 0.833710/34 [=======>....
.................] - ETA: 6s - loss: 2.0143 - accuracy: 0.844011/34 [========>.....................] - E
TA: 6s - loss: 1.9167 - accuracy: 0.850812/34 [=========>....................] - ETA: 6s - loss: 1.8280 -
accuracy: 0.857113/34 [==========>...................] - ETA: 6s - loss: 1.7481 - accuracy: 0.8629
14/34 [===========>..................] - ETA: 6s - loss: 1.6816 - accuracy: 0.8675
15/34 [============>.................] - ETA: 5s - loss: 1.6391 - accuracy: 0.8702
16/34 [=============>................] - ETA: 5s - loss: 1.5688 - accuracy: 0.8752
17/34 [==============>...............] - ETA: 5s - loss: 1.5208 - accuracy: 0.878318/34 [=========
=====>...............] - ETA: 5s - loss: 1.4760 - accuracy: 0.881619/34 [===============>..............]
- ETA: 4s - loss: 1.4436 - accuracy: 0.883820/34 [================>.............] - ETA: 4s - loss: 1.410
2 - accuracy: 0.886321/34 [=================>............] - ETA: 4s - loss: 1.3838 - accuracy: 0.8881
22/34 [==================>...........] - ETA: 4s - loss: 1.3526 - accuracy: 0.8903
23/34 [===================>..........] - ETA: 3s - loss: 1.3214 - accuracy: 0.8925
24/34 [====================>.........] - ETA: 3s - loss: 1.3040 - accuracy: 0.8938
25/34 [=====================>........] - ETA: 3s - loss: 1.2869 - accuracy: 0.895126/34 [=====
=================>.......] - ETA: 2s - loss: 1.2685 - accuracy: 0.896227/34 [=======================>.....
..] - ETA: 2s - loss: 1.2478 - accuracy: 0.897728/34 [========================>......] - ETA: 2s - loss: 1
.2300 - accuracy: 0.899029/34 [=========================>.....] - ETA: 1s - loss: 1.2188 - accuracy: 0.899
930/34 [==========================>....] - ETA: 1s - loss: 1.2032 - accuracy: 0.9010
31/34 [===========================>...] - ETA: 1s - loss: 1.1892 - accuracy: 0.9021
32/34 [============================>..] - ETA: 0s - loss: 1.1814 - accuracy: 0.9026
33/34 [=============================>.] - ETA: 0s - loss: 1.1739 - accuracy: 0.903134/34 [=
=============================] - ETA: 0s - loss: 1.1724 - accuracy: 0.903234/34 [========================
======] - 18s 525ms/step - loss: 1.1724 - accuracy: 0.9032 - val_loss: 0.8629 - val_accuracy: 0.9248
 1/11 [=>............................] - ETA: 0s - loss: 0.7185 - accuracy: 0.9363 2/11 [====>.........
..............] - ETA: 1s - loss: 0.7766 - accuracy: 0.9303 3/11 [=======>......................] - ETA:
1s - loss: 0.7911 - accuracy: 0.9295 4/11 [=========>....................] - ETA: 1s - loss: 0.8020 - acc
uracy: 0.9293 5/11 [============>.................] - ETA: 1s - loss: 0.8014 - accuracy: 0.9290
 6/11 [===============>..............] - ETA: 1s - loss: 0.8028 - accuracy: 0.9288
 7/11 [==================>...........] - ETA: 1s - loss: 0.7895 - accuracy: 0.9298
 8/11 [=====================>........] - ETA: 0s - loss: 0.7975 - accuracy: 0.9293
 9/11 [========================>......] - ETA: 0s - loss: 0.7941 - accuracy: 0.929710/11 [============
==============>...] - ETA: 0s - loss: 0.7975 - accuracy: 0.929511/11 [==============================] - E
TA: 0s - loss: 0.8068 - accuracy: 0.928811/11 [==============================] - 5s 420ms/step - loss: 0.
8068 - accuracy: 0.9288
1/2 [==============>...............] - ETA: 0s - loss: 0.9459 - accuracy: 0.91842/2 [=====================
=========] - ETA: 0s - loss: 0.8629 - accuracy: 0.92482/2 [==============================] - 0s 113ms/step
- loss: 0.8629 - accuracy: 0.9248
```

Figure 4.10: **Test and train for U-Net**

4.3.4 Application of Machine Learning Algorithms

The applications of Machine Learning algorithms like VGG16 and U-Net will produce higher results with higher accuracy. VGG16 method is the standard method for classification and for producing the higher accuracy values but here we have even used U-Net as an alternative .

4.3.5 Creation of Graphical User Interface (GUI)

A simple GUI will be used to create a better impact for the user. The user interface for this project is created by using Tkinter module.

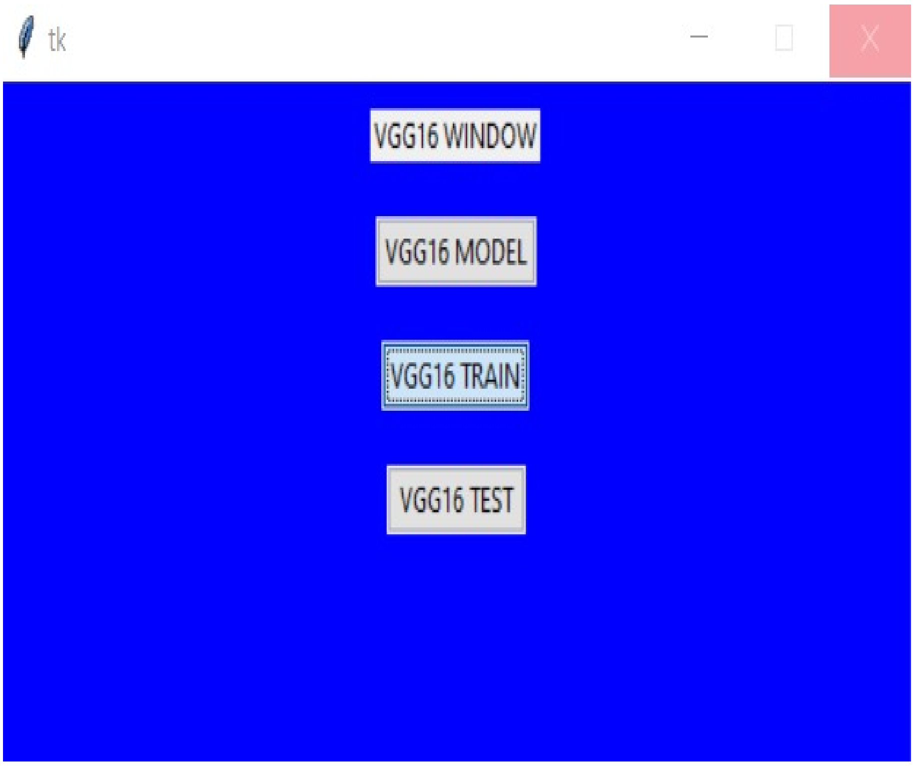
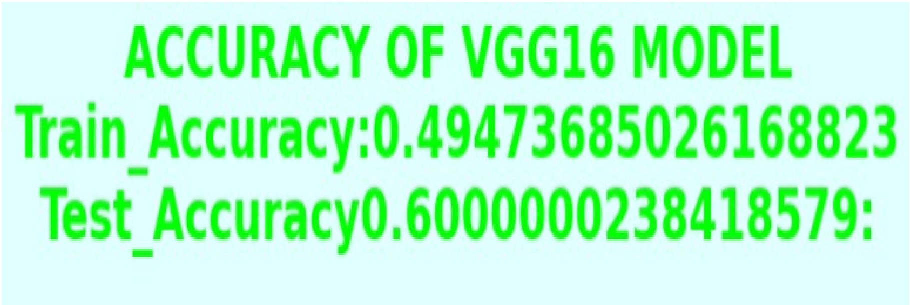


Figure 4.11: **GUI for VGG16**



```
ACCURACY OF VGG16 MODEL
Train_Accuracy:0.4947368502616882
Test_Accuracy0.6000000238418579:
```

Figure 4.12: **Output for VGG16**

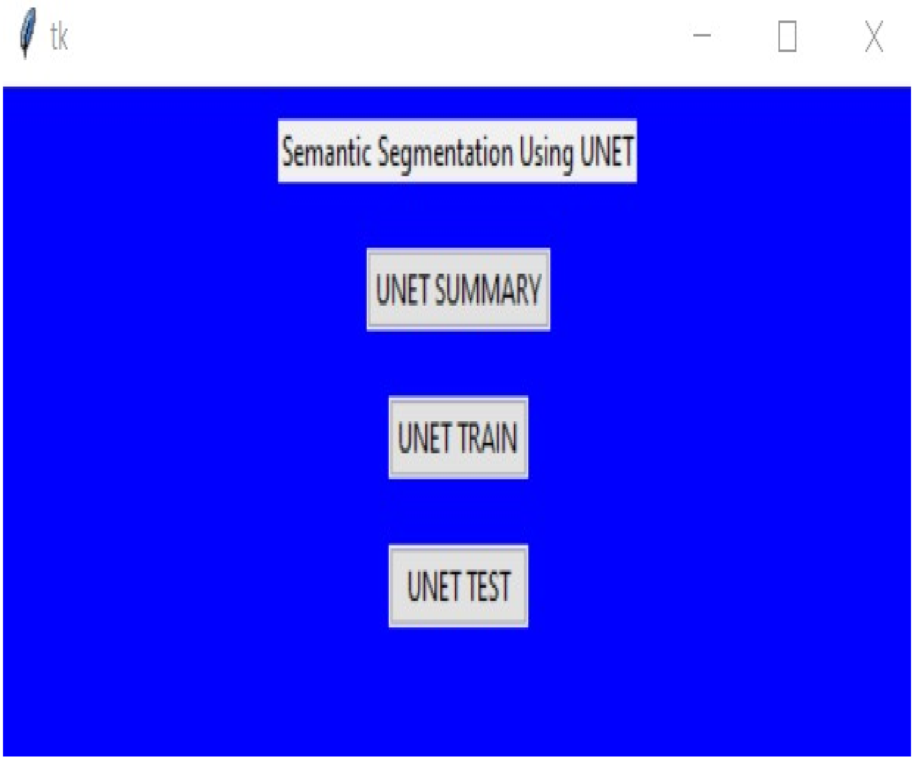
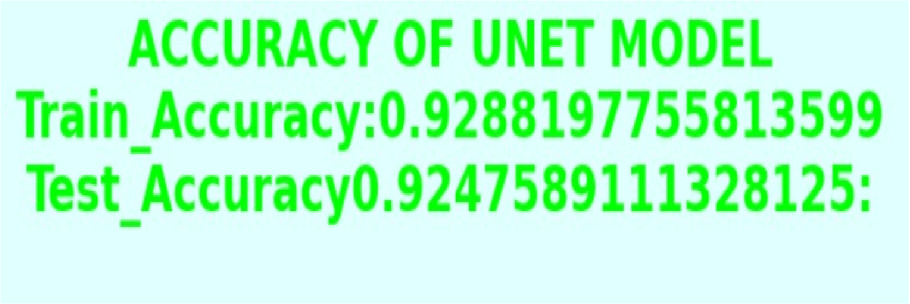


Figure 4.13: **GUI for U-Net**



```
ACCURACY OF UNET MODEL
Train_Accuracy:0.9288197755813599
Test_Accuracy0.9247589111328125:
```

Figure 4.14: **Output for U-Net**

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

The input here is an image. The output here is given whether the image is forged or not in the form of text in the first in the first output image and then it is segmented for localizing the forged part on the image if it is forged , this is depicted in the second output image else it does not localize .

5.1.1 Input Design

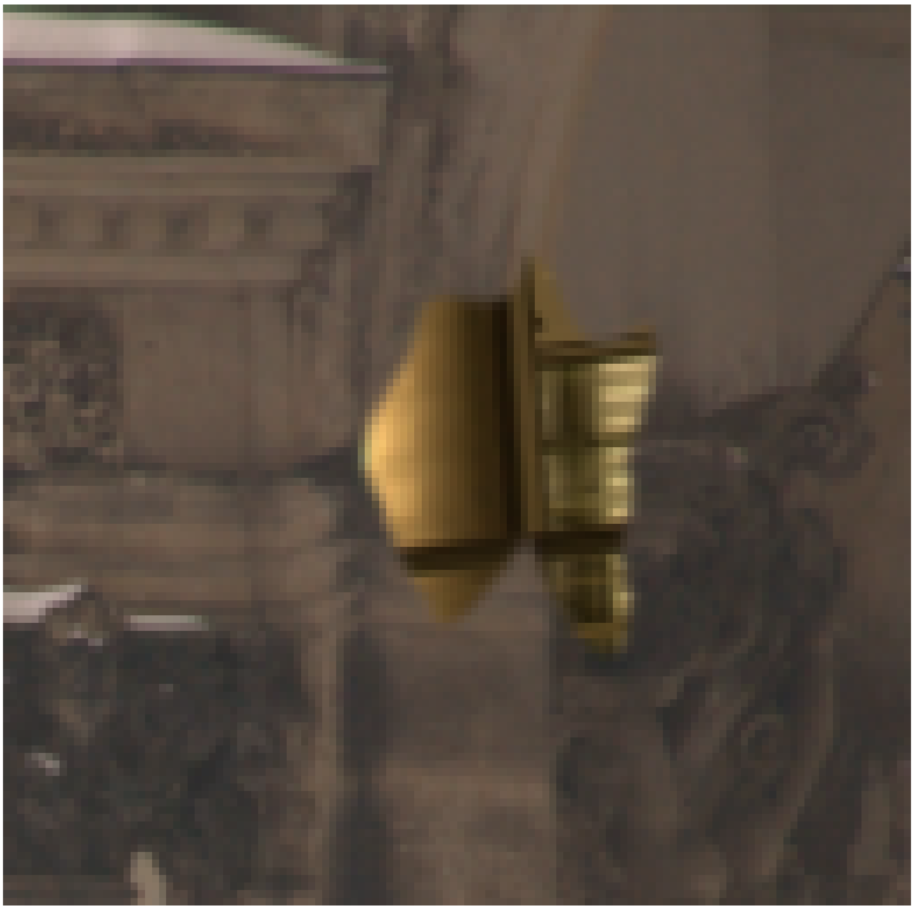
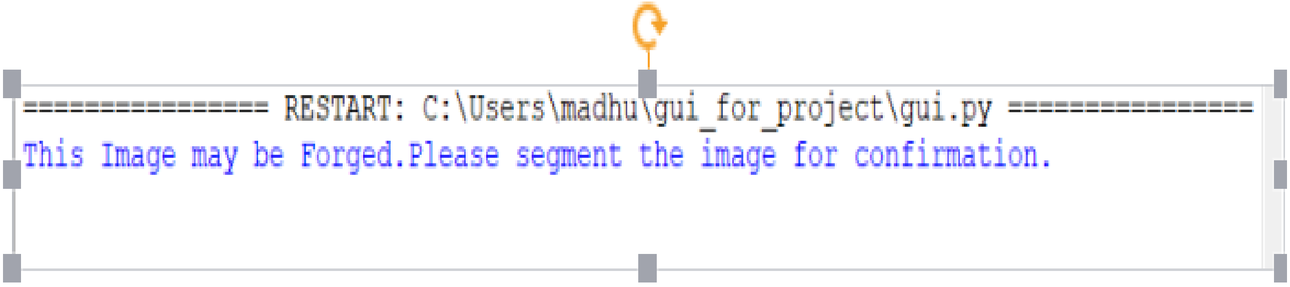


Figure 5.1: **Input 1**

5.1.2 Output Design



```
================ RESTART: C:\Users\madhu\gui_for_project\gui.py ================
This Image may be Forged.Please segment the image for confirmation.
```

Figure 5.2: **Output 1**

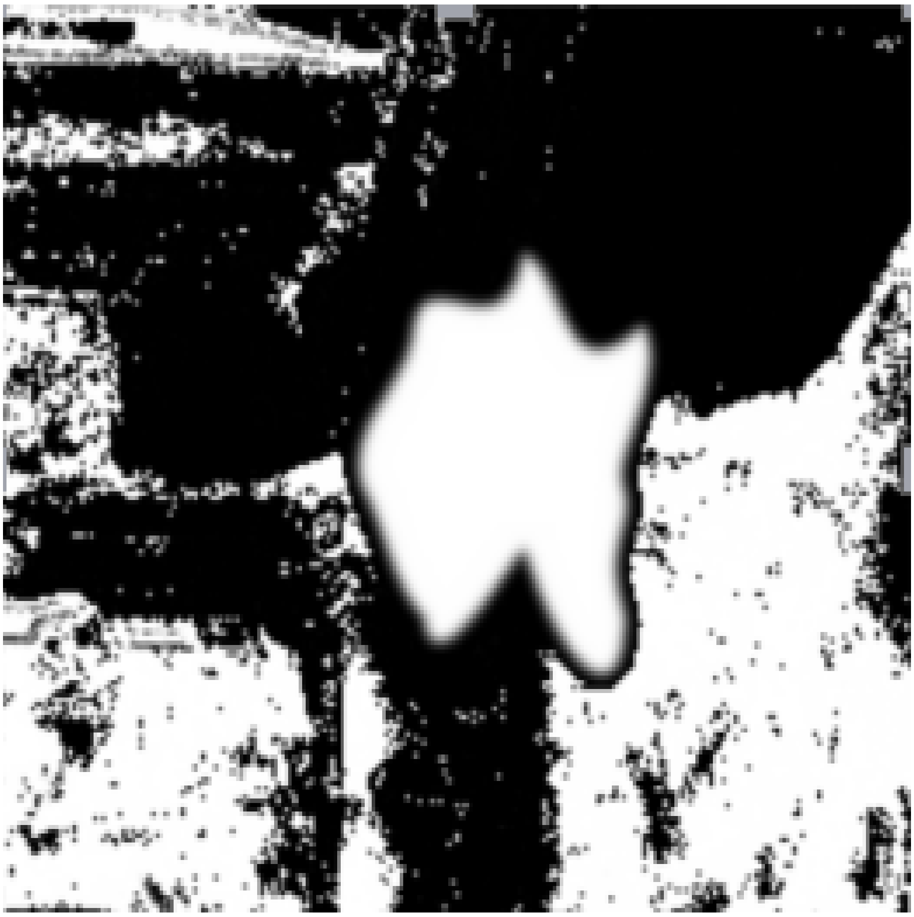


Figure 5.3: **Output 1**

5.2 Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or damage in a work product. It provides a way to check the functionality of components,assemblies, sub-assemblies or a finished product.It is the process of testing software to ensure that the Software system meets its requirements and expectations of user and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirements.

5.3 Types of Testing

5.3.1 Unit testing

Datasets are taken and we are doing the training and testing. Each and every part of code is seperately tested.

5.3.2 Integration testing

Assessment Test or Integration Test is a software test where different combinations are combined and tested as a group.The purpose of this test level is to extract communication from the interaction between the integrated units.Test test characters and test frequencies are used to assist in integration testing

5.3.3 White Box Testing

White Box Testing is a software testing system that tests structures or application performance, as opposed to its functionality. Here the testing strategy is known to the tester.

5.3.4 Black Box Testing

Black Box Testing is one of the types of software testing that examines the functionality of an application without peering into its workings or internal structures.This type of test can be applied virtually to every level of software testing:unit integration,system and acceptance. Here the testing strategies are unknown to the tester.

5.4 Testing Strategy

The developer should conduct the successful technical reviews to perform the testing successful.Testing starts with the component level and work from outside toward the integration of the whole computer based system.Different testing techniques are suitable at different point in time.Testing is organized by the developer of the software and by an independent test group.Debugging and testing are different activities,then also the debugging should be accommodated in any strategy of testing.

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The proposed system is CNN based and has the ability to detect digital image forgery in image splicing forgery technique of different formats efficiently . The system is tested using two algorithms VGG16 and U-Net and found that VGG16 algorithm produces the best results with an accuracy of 92 % in localizing the forged part on the image .

6.2 Comparison of Existing and Proposed System

This is a very effective algorithm but has a little high computational time when compared with other algorithms . This Proposed system works well only for images that are forged with image splicing techniques and not with other techniques .

6.3 Advantages of the Proposed System

1 . It is efficient for real time applications like a forensic tool while investigating crimes involving manipulation of images .

2 . Forged images with a little variations in pixels can be detected .

6.4 Sample Code

```
# This will import all the widgets
# and modules which are available in
# tkinter and ttk module
from tkinter import *
from tkinter.ttk import *
from tkinter import filedialog
import tkinter as tk
from PIL import ImageTk,Image
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import cv2

#imgname=filedialog.askopenfilename()
#imgname=Image.open(s)
#imgname=”2.png”
# creates a Tk() object
master = Tk()
# sets the geometry of main
# root window
master.geometry(”300x300”)

def vgggmodel():
    from vgg16modelsummary import ml
    mo=ml()
    print(”Wait model is loading”)
    mo.summary()

def vgggtrain():
    try:
        from vgg16train.train import vggtrain
        mo=vggtrain()
        if mo==True:
            print(”model trained”)
        else:
            print(”model is not trained”)
    except:
        print(”Model is not loaded Successfully. Please re run the program and train”)

def vgggtest():
    try:
        from vgg16test import t1
        imgname=filedialog.askopenfilename()
        result=t1(r”C:\Users\madhu\gui_for_project\trained_models\vgg16model\stmodel.h5”,imgname)
        if result[0][1]>=0.4619:
            a=’This Image may be Forged.Please segment the image for confirmation.’
        else:
            a=’Not-Forged’
        print(a)
    except:
        print(”Model is not loaded successfully. Please re run the program and test with Image.”)

def vggg16():
    newmaster1 = Tk()
    newmaster1.geometry(”500x500”)
    newmaster1.configure(bg=’blue’)
    label = Label(newmaster1,text =”VGG16 WINDOW”)
    label.pack(pady = 10)
    btn = Button(newmaster1,text =”VGG16 MODEL”,command = vgggmodel)
    btn.pack(pady = 10)
    btn = Button(newmaster1,text =”VGG16 TRAIN”,command =vgggtrain)
    btn.pack(pady = 10)
    btn = Button(newmaster1,text =”VGG16 TEST”,command = vgggtest)
    btn.pack(pady = 10)

def semanticsegmentationmodel():
    from simple_unet_model_summary import simple_unet_model
    smodel=simple_unet_model(256,256,3)
    print(”Model is loading”)
    smodel.summary()
```

```

def semanticsegmentationtrain():
    print(”model is training”)
    from unet.unettrain import unettrain
    result=unettrain()
    #print(result)

def semanticsegmentationtest():
    global testno
    imgname=filedialog.askopenfilename()
    from unettest import unettest
    result=unettest(imgname)
    #print(result)

    g=plt.figure(figsize=(8, 5))
    g.patch.set_facecolor('lightcyan')
    plt.subplot(121)
    x=500
    plt.title('Selected Image')


    #imgname1=mpimg.imread(imgname)
    imgname1=cv2.imread(imgname,cv2.COLOR_BGR2RGB)
    imgname1=cv2.cvtColor(imgname1,cv2.COLOR_BGR2RGB)
    imgname1=cv2.resize(imgname1,(x,x))
    plt.imshow(imgname1)
    plt.subplot(122)
    plt.title('Prediction of selected Image')
    result=cv2.resize(result ,(x,x))
    plt.imshow(result , cmap='gray')
    plt.imsave(r”C:\Users\madhu\gui_for_project\output_images\unetmaskoutput{}.png”.format(testno), result , cmap='gray')

    #plt.imsave(r”C:\Users\madhu\gui_for_project\output_images\orginalimage{}.png”.format(testno),imgname, cmap='gray')
    plt.show()
    print(”semanticsegmentationtest”)

def semanticsegmentation():
    print(”Semantic Segmentation using UNET”)
    newmaster2 = Tk()
    newmaster2.geometry(”500x500”)
    newmaster2.configure(bg=’blue’)
    label = Label(newmaster2,text =”Semantic Segmentation Using UNET”)
    label.pack(pady = 10)
    btn = Button(newmaster2,text =”UNET SUMMARY”,command = semanticsegmentationmodel)
    btn.pack(pady = 10)
    btn = Button(newmaster2,text =”UNET TRAIN”,command = semanticsegmentationtrain)
    btn.pack(pady = 10)
    btn = Button(newmaster2,text =”UNET TEST”,command = semanticsegmentationtest)
    btn.pack(pady = 10)

def vgggsemanticsegmentationmodel():
    from semvgg16model import semanticvgg16model
    print(”Modified vgg16 with 2layers for segmentation.”)
    semanticvgg16model()

def vgggsemanticsegmentationtrain():
    from semanticvgg.vggsemsegtrain import vggsemsegtrain
    result=vggsemsegtrain()
    print(”Training Completed”)

def vgggsemanticsegmentationtest():
    from semtest import semvgg16
    imgname=filedialog.askopenfilename()
    result=semvgg16(imgname)
    #print(result)
    g=plt.figure(figsize=(8, 5))
    g.patch.set_facecolor(’lightcyan’)
    plt.subplot(121)
    x=500
    plt.title(’Selected Image’)

    #imgname1=mpimg.imread(imgname)
    imgname1=cv2.imread(imgname,cv2.COLOR_BGR2RGB)
    imgname1=cv2.cvtColor(imgname1,cv2.COLOR_BGR2RGB)
    imgname1=cv2.resize(imgname1,(x,x))
    plt.imshow(imgname1)
    plt.subplot(122)
    plt.title('Prediction of selected Image')
    result=cv2.resize(result,(x,x))
    plt.imshow(result, cmap='gray')
    plt.imsave(r”C:\Users\madhu\gui_for_project\output_images\vgg16segmaskoutput{}.png”.format(testno), result, cmap='gray')

    #plt.imsave(r”C:\Users\madhu\gui_for_project\output_images\orginalimage{}.png”.format(testno),imgname, cmap='gray')
    #cv2.imwrite(r”C:\Users\madhu\gui_for_project\output_images\vgg16segmaskoutput”,result)
    plt.show()

def vgggsemanticsegmentation():
    newmaster3 = Tk()
    newmaster3.configure(bg='blue')
    newmaster3.geometry(”500x500”)
    label = Label(newmaster3,text =”VGG16 SEMANTIC SEGMENTATION WINDOW”)
    label.pack(pady = 10)
    btn = Button(newmaster3,text =”MODEL SUMMARY”,command = vgggsemanticsegmentationmodel)
    btn.pack(pady = 10)
    btn = Button(newmaster3,text =”MODEL TRAINING”,command = vgggsemanticsegmentationtrain)
    btn.pack(pady = 10)
    btn = Button(newmaster3,text =”MODEL TEST”,command = vgggsemanticsegmentationtest)
    btn.pack(pady = 10)

def openNewWindow():
    newmaster = Tk()
    newmaster.geometry(”500x500”)
    newmaster.configure(bg='blue')
    label = Label(newmaster,text =”ALGORITHMS WINDOW”)
    label.pack(pady = 10)
    btn = Button(newmaster,text =”VGG16”,command = vggg16)
    btn.pack(pady = 10)
    btn = Button(newmaster,text =”SEMANTIC SEGMENTATION USING UNET”,command = semanticsegmentation)
    btn.pack(pady = 10)
    btn = Button(newmaster,text =”SEMANTIC SEGMENTATION USING VGG16”,command = vgggsemanticsegmentation)
    btn.pack(pady = 10)

def total():
    from vgg16test import t1
    imgname=filedialog.askopenfilename()
    result1=t1(r”C:\Users\madhu\gui_for_project\trained_models\vgg16model\stmodel.h5”,imgname)
    if result1[0][1]>=0.4619:
        a='FORGED'
    else:
        a='NOT−FORGED'

    from semtest import semvgg16
    result=semvgg16(imgname)
    g=plt.figure(figsize=(8, 5))
    g.patch.set_facecolor('lightcyan')
    plt.subplot(121)
    x=500
    plt.title('Selected Image')

    #imgname1=mpimg.imread(imgname)
    imgname1=cv2.imread(imgname,cv2.COLOR_BGR2RGB)
    imgname1=cv2.cvtColor(imgname1,cv2.COLOR_BGR2RGB)
    imgname1=cv2.resize(imgname1,(x,x))
    plt.imshow(imgname1)
    plt.subplot(122)
    plt.title('Prediction of selected Image as {}'.format(a))
    result=cv2.resize(result,(x,x))
    plt.imshow(result, cmap='gray')
    plt.show()
#testno=int(input(”Please Enter Test no for saving images in folder with different number”))
testno=1
```

```
label = Label(master,text =”MAIN WINDOW”)
master.configure(bg='blue')
label.pack(pady = 10)

# a button widget which will open a
# new window on button click

btn = Button(master,text =”TEST WITH EACH ALGORITHM”,command = openNewWindow)
btn.pack(pady = 10)
btn = Button(master,text =”FINAL TEST ”,command =total)
btn.pack(pady = 10)

# mainloop, runs infinitely
mainloop()
```

Output

Figure 6.1: **Output 1**

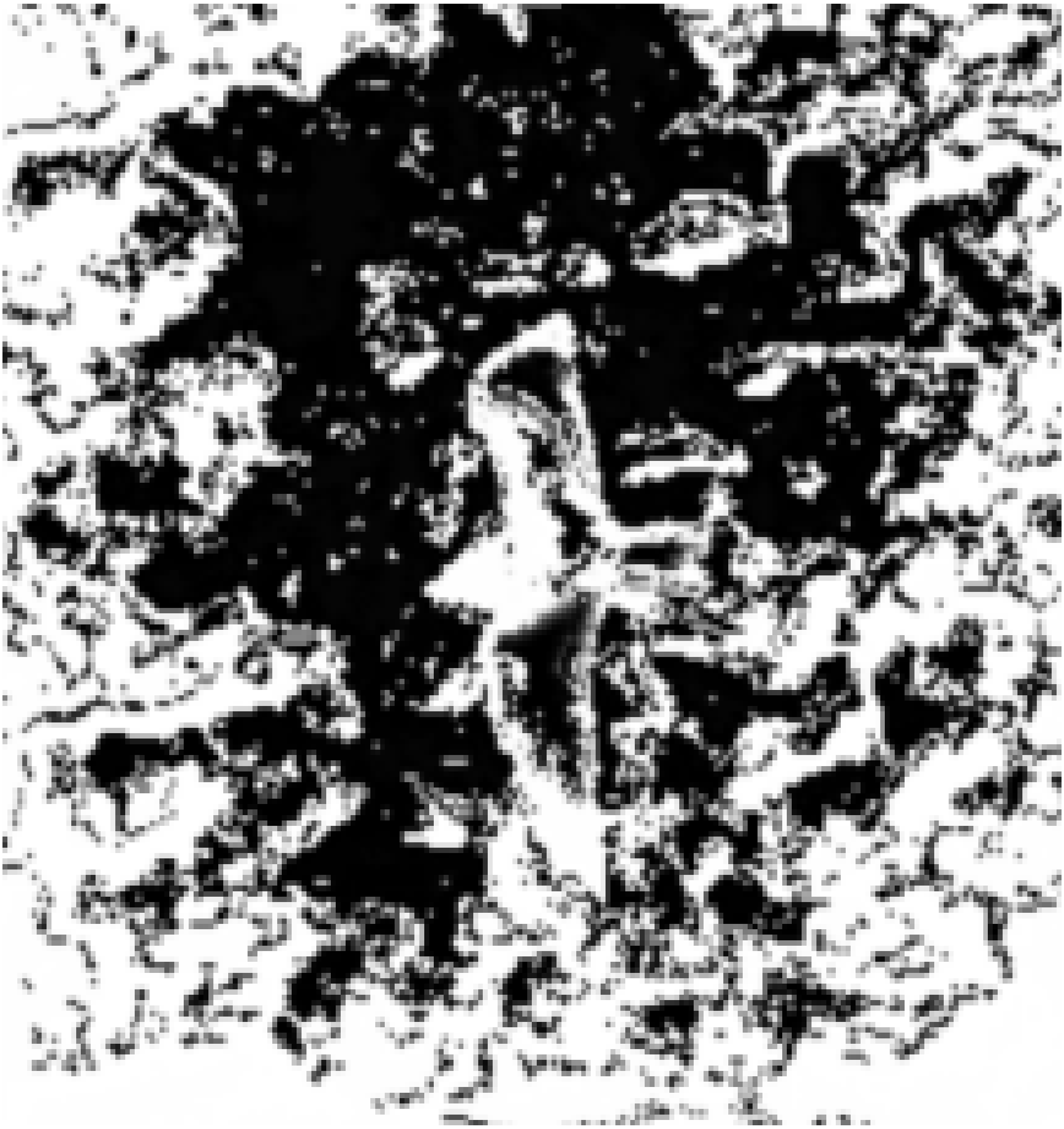


Figure 6.2: **Output 2**

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

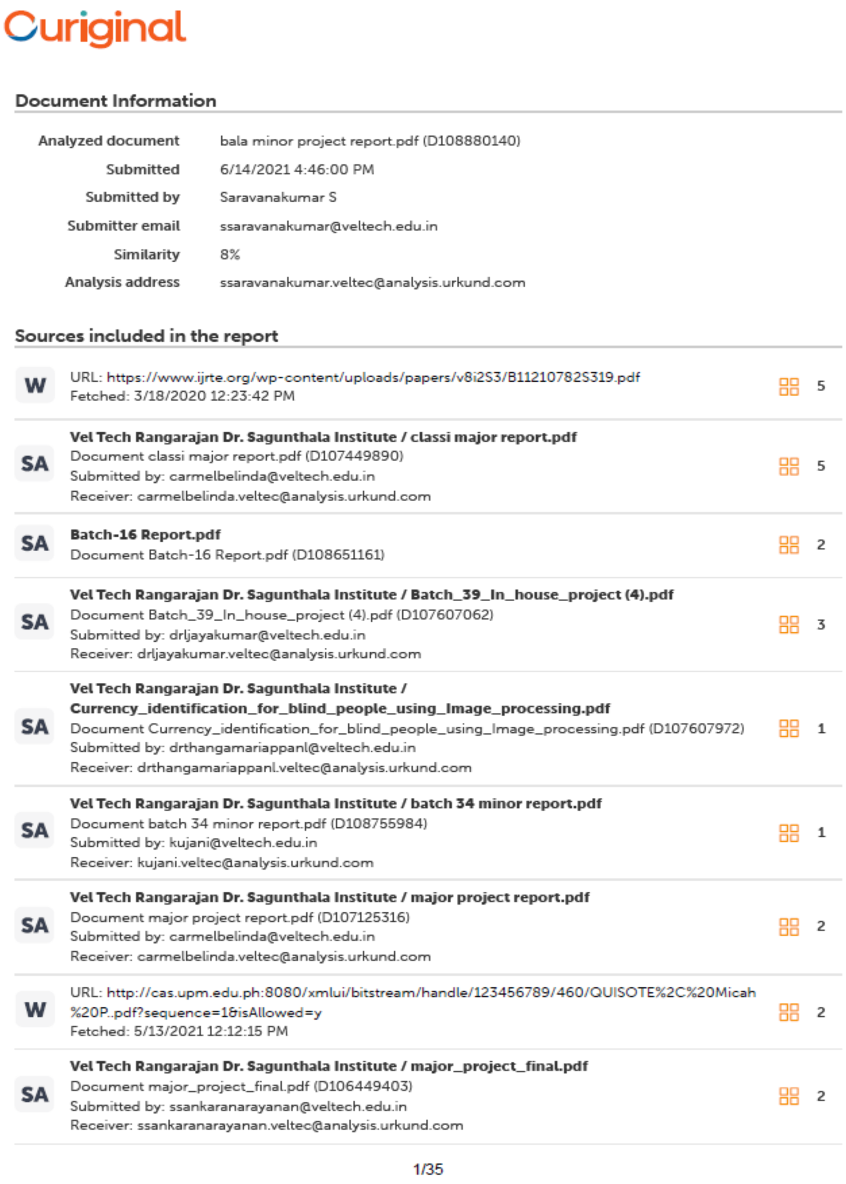
This project describes a new method of detecting forged images and localizing the forged part on the image that are forged by image splicing technique . The output is given using 1 x 2 array to know whether the image is forged or not and the output for localization is given by a black and white image where the white part depicts the area of forgery on the image .

7.2 Future Enhancements

As a future scope this project can be enhanced by gathering and training the algorithm with more number of high resolution images as dataset of different classes and categories and also the accuracy of predicting the forgery and the accuracy of localization of forged part on the output image can be increased .

Chapter 8

PLAGIARISM REPORT



Ouriginal

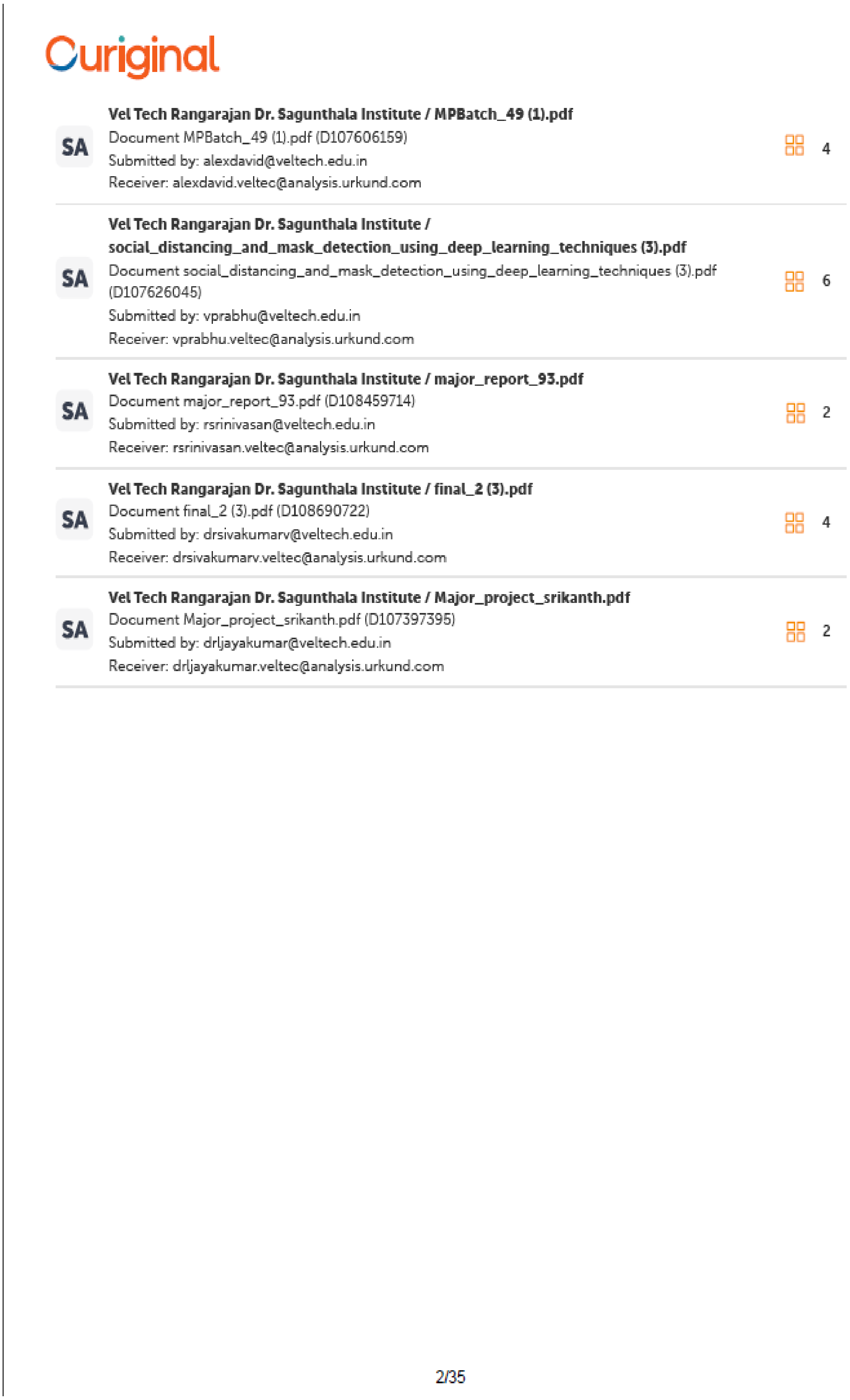
Document Information

Analyzed document	bala minor project report.pdf (D108880140)
Submitted	6/14/2021 4:46:00 PM
Submitted by	Saravanakumar S
Submitter email	ssaravanakumar@veltech.edu.in
Similarity	8%
Analysis address	ssaravanakumar.veltec@analysis.urkund.com

Sources included in the report

W	URL: https://www.ijrte.org/wp-content/uploads/papers/v8i2S3/B11210782S319.pdf Fetched: 3/18/2020 12:23:42 PM	⊞	5
SA	**Vel Tech Rangarajan Dr. Sagunthala Institute / classi major report.pdf** Document classi major report.pdf (D107449890) Submitted by: carmelbelinda@veltech.edu.in Receiver: carmelbelinda.veltec@analysis.urkund.com	⊞	5
SA	**Batch-16 Report.pdf** Document Batch-16 Report.pdf (D108651161)	⊞	2
SA	**Vel Tech Rangarajan Dr. Sagunthala Institute / Batch_39_In_house_project (4).pdf** Document Batch_39_In_house_project (4).pdf (D107607062) Submitted by: drljayakumar@veltech.edu.in Receiver: drljayakumar.veltec@analysis.urkund.com	⊞	3
SA	**Vel Tech Rangarajan Dr. Sagunthala Institute / Currency_identification_for_blind_people_using_Image_processing.pdf** Document Currency_identification_for_blind_people_using_Image_processing.pdf (D107607972) Submitted by: drthangamariappanl@veltech.edu.in Receiver: drthangamariappanl.veltec@analysis.urkund.com	⊞	1
SA	**Vel Tech Rangarajan Dr. Sagunthala Institute / batch 34 minor report.pdf** Document batch 34 minor report.pdf (D108755984) Submitted by: kujani@veltech.edu.in Receiver: kujani.veltec@analysis.urkund.com	⊞	1
SA	**Vel Tech Rangarajan Dr. Sagunthala Institute / major project report.pdf** Document major project report.pdf (D107125316) Submitted by: carmelbelinda@veltech.edu.in Receiver: carmelbelinda.veltec@analysis.urkund.com	⊞	2
W	URL: http://cas.upm.edu.ph:8080/xmlui/bitstream/handle/123456789/460/QUISOTE%2C%20Micah%20P..pdf?sequence=1&isAllowed=y Fetched: 5/13/2021 12:12:15 PM	⊞	2
SA	**Vel Tech Rangarajan Dr. Sagunthala Institute / major_project_final.pdf** Document major_project_final.pdf (D106449403) Submitted by: ssankaranarayanan@veltech.edu.in Receiver: ssankaranarayanan.veltec@analysis.urkund.com	⊞	2

1/35



Ouriginal

SA
Vel Tech Rangarajan Dr. Sagunthala Institute / MPBatch_49 (1).pdf
Document MPBatch_49 (1).pdf (D107606159)
Submitted by: alexdavid@veltech.edu.in
Receiver: alexdavid.veltec@analysis.urkund.com
4

SA
Vel Tech Rangarajan Dr. Sagunthala Institute / social_distancing_and_mask_detection_using_deep_learning_techniques (3).pdf
Document social_distancing_and_mask_detection_using_deep_learning_techniques (3).pdf (D107626045)
Submitted by: vprabhu@veltech.edu.in
Receiver: vprabhu.veltec@analysis.urkund.com
6

SA
Vel Tech Rangarajan Dr. Sagunthala Institute / major_report_93.pdf
Document major_report_93.pdf (D108459714)
Submitted by: rsrinivasan@veltech.edu.in
Receiver: rsrinivasan.veltec@analysis.urkund.com
2

SA
Vel Tech Rangarajan Dr. Sagunthala Institute / final_2 (3).pdf
Document final_2 (3).pdf (D108690722)
Submitted by: drsivakumarv@veltech.edu.in
Receiver: drsivakumarv.veltec@analysis.urkund.com
4

SA
Vel Tech Rangarajan Dr. Sagunthala Institute / Major_project_srikanth.pdf
Document Major_project_srikanth.pdf (D107397395)
Submitted by: drljayakumar@veltech.edu.in
Receiver: drljayakumar.veltec@analysis.urkund.com
2

2/35

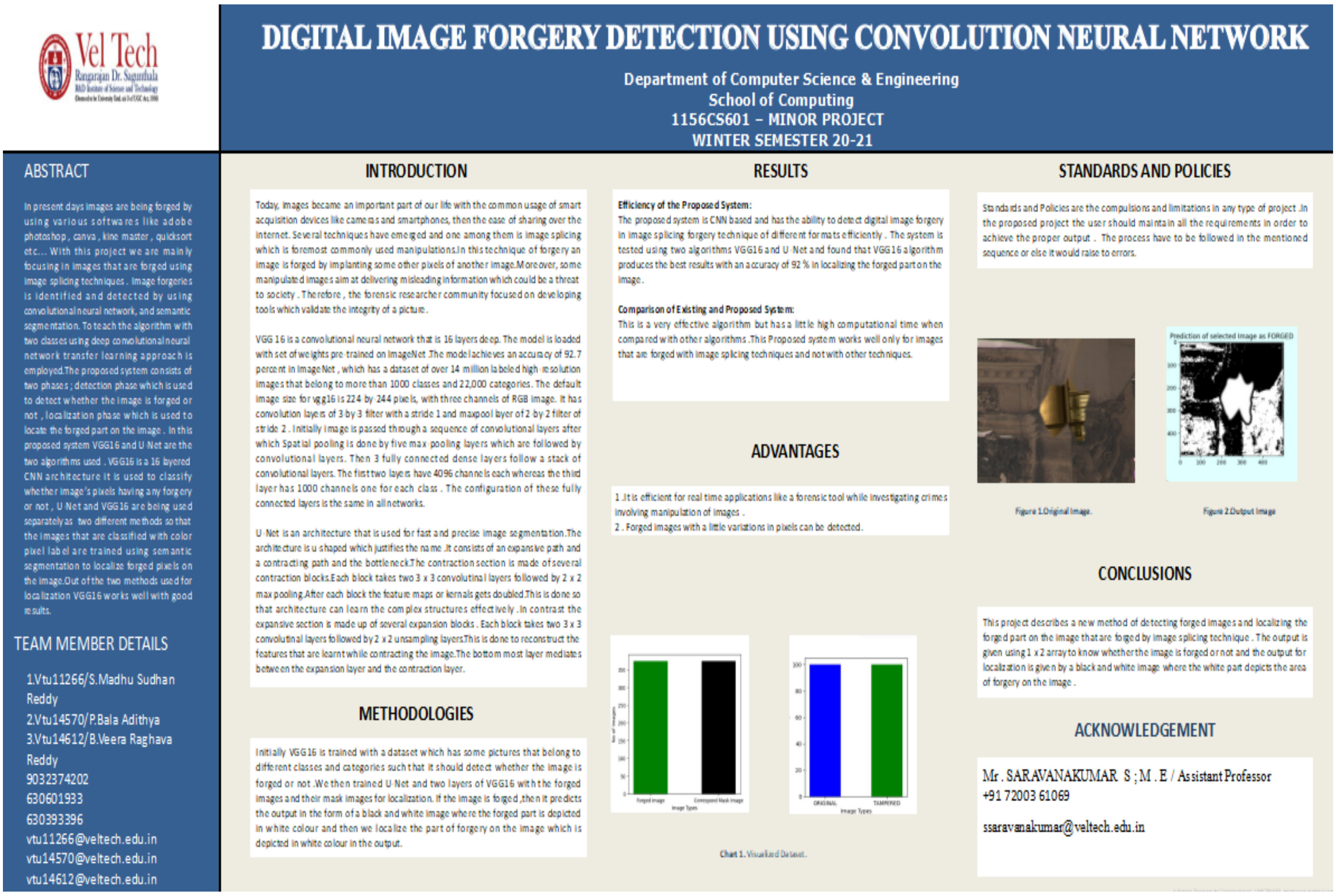
Chapter 9

SOURCE CODE & POSTER PRESENTATION

9.1 Source code

```
# This will import all the widgets
# and modules which are available in
# tkinter and ttk module
from tkinter import *
from tkinter.ttk import *
from tkinter import filedialog
import tkinter as tk
from PIL import ImageTk,Image
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import cv2

#imgname=filedialog.askopenfilename()
#imgname=Image.open(s)
#imgname="2.png"
# creates a Tk() object
master = Tk()
# sets the geometry of main
# root window
master.geometry("300x300")

def vgggmodel():
    from vgg16modelsummary import m1
    mo=m1()
    print("Wait model is loading")
    mo.summary()

def vgggtrain():
    try:
        from vgg16train.train import vggtrain
        mo=vggtrain()
        if mo==True:
            print("model trained")
        else:
            print("model is not trained")
    except:
        print("Model is not loaded Successfully. Please re run the program and train")

def vgggtest():
    try:
        from vgg16test import t1
        imgname=filedialog.askopenfilename()
        result=t1(r"C:\Users\madhu\gui_for_project\trained_models\vgg16model\stmodel.h5",imgname)
        if result[0][1]>=0.4619:
            a='This Image may be Forged.Please segment the image for confirmation.'
        else:
            a='Not-Forged'
        print(a)
    except:
        print("Model is not loaded successfully. Please re run the program and test with Image.")

def vggg16():
    newmaster1 = Tk()
    newmaster1.geometry("500x500")
    newmaster1.configure(bg='blue')
    label = Label(newmaster1,text ="VGG16 WINDOW")
    label.pack(pady = 10)
    btn = Button(newmaster1,text =”VGG16 MODEL”,command = vgggmodel)
    btn.pack(pady = 10)
    btn = Button(newmaster1,text =”VGG16 TRAIN”,command =vgggtrain)
    btn.pack(pady = 10)
    btn = Button(newmaster1,text =”VGG16 TEST”,command = vgggtest)
    btn.pack(pady = 10)

def semanticsegmentationmodel():
    from simple_unet_model_summary import simple_unet_model
    smodel=simple_unet_model(256,256,3)
    print(”Model is loading”)
    smodel.summary()

def semanticsegmentationtrain():
    print(”model is training”)
    from unet.unettrain import unettrain
    result=unettrain()
    #print(result)

def semanticsegmentationtest():
    global testno
    imgname=filedialog.askopenfilename()
    from unettest import unettest
    result=unettest(imgname)
    #print(result)

    g=plt.figure(figsize=(8, 5))
    g.patch.set_facecolor('lightcyan')
    plt.subplot(121)
    x=500
    plt.title('Selected Image')


    #imgname1=mpimg.imread(imgname)
    imgname1=cv2.imread(imgname,cv2.COLOR_BGR2RGB)
    imgname1=cv2.cvtColor(imgname1,cv2.COLOR_BGR2RGB)
    imgname1=cv2.resize(imgname1,(x,x))
    plt.imshow(imgname1)
    plt.subplot(122)
    plt.title(’Prediction of selected Image’)
    result=cv2.resize(result,(x,x))
    plt.imshow(result, cmap=’gray’)
    plt.imsave(r”C:\Users\madhu\gui_for_project\output_images\unetmaskoutput{}.png”.format(testno), result, cmap=’gray’)

   #plt.imsave(r”C:\Users\madhu\gui_for_project\output_images\orginalimage{}.png”.format(testno),imgname, cmap=’gray’)
    plt.show()
    print(”semanticsegmentationtest”)

def semanticsegmentation():
    print(”Semantic Segmentation using UNET”)
    newmaster2 = Tk()
    newmaster2.geometry(”500x500”)
    newmaster2.configure(bg=’blue’)
    label = Label(newmaster2,text =”Semantic Segmentation Using UNET”)
    label.pack(pady = 10)
    btn = Button(newmaster2,text =”UNET SUMMARY”,command = semanticsegmentationmodel)
    btn.pack(pady = 10)
    btn = Button(newmaster2,text =”UNET TRAIN”,command = semanticsegmentationtrain)
    btn.pack(pady = 10)
    btn = Button(newmaster2,text =”UNET TEST”,command = semanticsegmentationtest)
    btn.pack(pady = 10)

def vgggsemanticsegmentationmodel():
    from semvgg16model import semanticvgg16model
    print(”Modified vgg16 with 2layers for segmentation.”)
    semanticvgg16model()

def vgggsemanticsegmentationtrain():
    from semanticvgg.vggsemsegtrain import vggsemsegtrain
    result=vggsemsegtrain()
    print(”Training Completed”)

def vgggsemanticsegmentationtest():
    from semtest import semvgg16
    imgname=filedialog.askopenfilename()
    result=semvgg16(imgname)
    #print(result)
    g=plt.figure(figsize=(8, 5))
    g.patch.set_facecolor('lightcyan')
    plt.subplot(121)
    x=500
    plt.title('Selected Image')

    #imgname1=mpimg.imread(imgname)
    imgname1=cv2.imread(imgname,cv2.COLOR_BGR2RGB)
    imgname1=cv2.cvtColor(imgname1,cv2.COLOR_BGR2RGB)
    imgname1=cv2.resize(imgname1,(x,x))
    plt.imshow(imgname1)
    plt.subplot(122)
    plt.title('Prediction of selected Image')
    result=cv2.resize(result,(x,x))
    plt.imshow(result, cmap='gray')
    plt.imsave(r”C:\Users\madhu\gui_for_project\output_images\vgg16segmaskoutput{}.png”.format(testno), result, cmap='gray')

    #plt.imsave(r”C:\Users\madhu\gui_for_project\output_images\orginalimage{}.png”.format(testno),imgname, cmap='gray')
    #cv2.imwrite(r”C:\Users\madhu\gui_for_project\output_images\vgg16segmaskoutput”,result)
    plt.show()

def vgggsemanticsegmentation():
    newmaster3 = Tk()
    newmaster3.configure(bg='blue')
    newmaster3.geometry(”500x500”)
    label = Label(newmaster3,text =”VGG16 SEMANTIC SEGMENTATION WINDOW”)
    label.pack(pady = 10)
    btn = Button(newmaster3,text =”MODEL SUMMARY”,command = vgggsemanticsegmentationmodel)
    btn.pack(pady = 10)
    btn = Button(newmaster3,text =”MODEL TRAINING”,command = vgggsemanticsegmentationtrain)
    btn.pack(pady = 10)
    btn = Button(newmaster3,text ="MODEL TEST",command = vgggsemanticsegmentationtest)
    btn.pack(pady = 10)

def openNewWindow():
    newmaster = Tk()
    newmaster.geometry("500x500")
    newmaster.configure(bg='blue')
    label = Label(newmaster,text ="ALGORITHMS WINDOW")
    label.pack(pady = 10)
    btn = Button(newmaster,text ="VGG16",command = vggg16)
    btn.pack(pady = 10)
    btn = Button(newmaster,text ="SEMANTIC SEGMENTATION USING UNET",command = semanticsegmentation)
    btn.pack(pady = 10)
    btn = Button(newmaster,text ="SEMANTIC SEGMENTATION USING VGG16",command = vgggsemanticsegmentation)
    btn.pack(pady = 10)

def total():
    from vgg16test import t1
    imgname=filedialog.askopenfilename()
    result1=t1(r"C:\Users\madhu\gui_for_project\trained_models\vgg16model\stmodel.h5",imgname)
    if result1[0][1]>=0.4619:
        a='FORGED'
    else:
        a='NOT–FORGED'

    from semtest import semvgg16
    result=semvgg16(imgname)
    g=plt.figure(figsize=(8, 5))
    g.patch.set_facecolor('lightcyan')
    plt.subplot(121)
    x=500
    plt.title('Selected Image')

    #imgname1=mpimg.imread(imgname)
    imgname1=cv2.imread(imgname,cv2.COLOR_BGR2RGB)
    imgname1=cv2.cvtColor(imgname1,cv2.COLOR_BGR2RGB)
    imgname1=cv2.resize(imgname1,(x,x))
    plt.imshow(imgname1)
    plt.subplot(122)
    plt.title('Prediction of selected Image as {}'.format(a))
    result=cv2.resize(result,(x,x))
    plt.imshow(result, cmap='gray')
    plt.show()
#testno=int(input(”Please Enter Test no for saving images in folder with different number”))
testno=1
label = Label(master,text =”MAIN WINDOW”)
master.configure(bg='blue')
label.pack(pady = 10)

# a button widget which will open a
# new window on button click

btn = Button(master,text =”TEST WITH EACH ALGORITHM”,command = openNewWindow)
btn.pack(pady = 10)
btn = Button(master,text =”FINAL TEST ”,command =total)
btn.pack(pady = 10)

# mainloop, runs infinitely
mainloop()
```

9.2 Poster Presentation



Vel Tech
Rangarajan Dr. Sagunthala
R&D Institute of Science and Technology

DIGITAL IMAGE FORGERY DETECTION USING CONVOLUTION NEURAL NETWORK

Department of Computer Science & Engineering
School of Computing
1156CS601 – MINOR PROJECT
WINTER SEMESTER 20-21

ABSTRACT

In present days images are being forged by using various softwares like adobe photoshop , canva , kine master , quicksort etc... With this project we are mainly focusing in images that are forged using image splicing techniques . Image forgeries is identified and detected by using convolutional neural network, and semantic segmentation. To teach the algorithm with two classes using deep convolutional neural network transfer learning approach is employed.The proposed system consists of two phases ; detection phase which is used to detect whether the image is forged or not , localization phase which is used to locate the forged part on the image . In this proposed system VGG16 and U-Net are the two algorithms used . VGG16 is a 16 layered CNN architecture it is used to classify whether image's pixels having any forgery or not , U-Net and VGG16 are being used separately as two different methods so that the images that are classified with color pixel label are trained using semantic segmentation to localize forged pixels on the image.Out of the two methods used for localization VGG16 works well with good results.

TEAM MEMBER DETAILS

1.Vtu11266/S.Madhu Sudhan Reddy
2.Vtu14570/P.Bala Adithya
3.Vtu14612/B.Veera Raghava Reddy
9032374202
630601933
630393396
vtu11266@veltech.edu.in
vtu14570@veltech.edu.in
vtu14612@veltech.edu.in

INTRODUCTION

Today, images became an important part of our life with the common usage of smart acquisition devices like cameras and smartphones, then the ease of sharing over the internet. Several techniques have emerged and one among them is image splicing which is foremost commonly used manipulations.In this technique of forgery an image is forged by implanting some other pixels of another image.Moreover, some manipulated images aim at delivering misleading information which could be a threat to society . Therefore , the forensic researcher community focused on developing tools which validate the integrity of a picture.

VGG 16 is a convolutional neural network that is 16 layers deep. The model is loaded with set of weights pre-trained on ImageNet .The model achieves an accuracy of 92.7 percent in ImageNet , which has a dataset of over 14 million labeled high-resolution images that belong to more than 1000 classes and 22,000 categories. The default image size for vgg16 is 224-by-244 pixels, with three channels of RGB image. It has convolution layers of 3-by-3 filter with a stride 1 and maxpool layer of 2-by-2 filter of stride 2 . Initially image is passed through a sequence of convolutional layers after which Spatial pooling is done by five max-pooling layers which are followed by convolutional layers. Then 3 fully connected dense layers follow a stack of convolutional layers. The first two layers have 4096 channels each whereas the third layer has 1000 channels one for each class . The configuration of these fully connected layers is the same in all networks.

U-Net is an architecture that is used for fast and precise image segmentation.The architecture is u-shaped which justifies the name .It consists of an expansive path and a contracting path and the bottleneck.The contraction section is made of several contraction blocks.Each block takes two 3 x 3 convolutinal layers followed by 2 x 2 max pooling.After each block the feature maps or kernals gets doubled.This is done so that architecture can learn the complex structures effectively .In contrast the expansive section is made up of several expansion blocks . Each block takes two 3 x 3 convolutinal layers followed by 2 x 2 unsampling layers.This is done to reconstruct the features that are learnt while contracting the image.The bottom most layer mediates between the expansion layer and the contraction layer.

METHODOLOGIES

Initially VGG16 is trained with a dataset which has some pictures that belong to different classes and categories such that it should detect whether the image is forged or not .We then trained U-Net and two layers of VGG16 with the forged images and their mask images for localization. If the image is forged ,then it predicts the output in the form of a black and white image where the forged part is depicted in white colour and then we localize the part of forgery on the image which is depicted in white colour in the output.

RESULTS

Efficiency of the Proposed System:
The proposed system is CNN based and has the ability to detect digital image forgery in image splicing forgery technique of different formats efficiently . The system is tested using two algorithms VGG16 and U-Net and found that VGG16 algorithm produces the best results with an accuracy of 92 % in localizing the forged part on the image .

Comparison of Existing and Proposed System:
This is a very effective algorithm but has a little high computational time when compared with other algorithms .This Proposed system works well only for images that are forged with image splicing techniques and not with other techniques.

ADVANTAGES

1 .It is efficient for real time applications like a forensic tool while investigating crimes involving manipulation of images .
2 . Forged images with a little variations in pixels can be detected.

Chart 1. Visualized Dataset.

STANDARDS AND POLICIES

Standards and Policies are the compulsions and limitations in any type of project .In the proposed project the user should maintain all the requirements in order to achieve the proper output . The process have to be followed in the mentioned sequence or else it would raise to errors.

Figure 1.Original Image.

Figure 2.Output Image

CONCLUSIONS

This project describes a new method of detecting forged images and localizing the forged part on the image that are forged by image splicing technique . The output is given using 1 x 2 array to know whether the image is forged or not and the output for localization is given by a black and white image where the white part depicts the area of forgery on the image .

ACKNOWLEDGEMENT

Mr . SARAVANAKUMAR S ; M . E / Assistant Professor
+91 72003 61069

ssaravanakumar@veltech.edu.in

References

[1] N.Hema Rajini , June(2019) , Image Forgery Identification using Convolutional Neural Network , International Journal of Recent Technology and Engineering (IJRTE) , 8(1S4) , 1421-1429.

[2] S.Prayla Shyry , Saranya Meka , Mahitha Moganti, (2019). Digital Image Forgery Detection , International Journal of Recent Technology and Engineering (IJRTE) , 8(2S3) , 255-263.

[3] Kanchan Jha , Kshitija Shirwadkar , Tejaswi Narvekar , Isha Kothari , Salabha Jacob , May(2020) , DIGITAL IMAGE FORGERY DETECTION , International Research Journal of Engineering and Technology (IRJET), 7(5) , 782-788.

[4] B. Santhosh Kumar1 ,S.Karthi , K.Karthika , Rajan Cristin , 2018 , A Systematic Study of Image Forgery Detection , Journal ofComputational and Theoretical Nanoscience , 15(7) , 2560-2564.

[5] Amit Doegara, Maitreyee Duttaa, Gaurav Kumarb , “CNN based Image Forgery Detection using pre-trained AlexNet Model ” , Proceedings of International Conference on Computational Intelligence IoT (ICCIIoT) , 2018.

[6] Yuan Rao, Jiangqun Ni,"A Deep Learning Approach to Detection of "Splicing and Copy-Move Forgeries in Images",IEEE International Workshop on Information Forensics and Security , 2018.

[7] Gul Muzaffer , Guzin Ulutas , "A new deep learning-based method to detection of copy-move forgery in digital images" , IEEE International Workshop , 2019.

[8] Boubacar Diallo , Thierry Urruty, Pascal Bourdon, Christine Fernandez-Maloigne , " Robust forgery detection for compressed images using CNN supervision " , Forensic Science International: Reports , 2020.

[9] Davide Cozzolino, Luisa Verdoliva., "Camera-based Image Forgery Localization using Convolutional Neural Networks" Department of Electrical Engineering and Information Technology University Federico II of Naples, Naples, Italy name.surname@unina.it , 2018.

[10] Franseco Marra , Diego Gragnaniello , Luisa Verdoliva , Giovanni Poggi,"A Full-Image Full-Resolution End-to-End-Trainable CNN Framework for Image Forgery Detection" , Ieee , 2020.

[11] Xinyi Wang, He Wang, Shaozhang Niu and Jiwei Zhang , "Detection and localization of image forgeries using improved mask regional convolutional neural network" Beijing Key Lab of Intelligent Telecommunication Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing, China, 2019.