



COMSATS University
Islamabad Sahiwal, Pakistan

Image Forgery Detection by using Machine Learning.

By

Ali Raza	CIIT/FA17-BCS-081/SWL
Husnain Bashir	CIIT/FA17-BCS-100/SWL
Hamza Siddiqui	CIIT/FA17-BCS-103/SWL

Supervisor

Dr. azhar Sadiq

Bachelor of Science in Computer Science (2017-2021)

The candidate confirms that the work submitted is their own and appropriate credit has been given where reference has been made to the work of others.



**COMSATS University Islamabad, Sahiwal
Pakistan**

Image Forgery Detection by using Machine Learning.

A project presented to

COMSATS Institute of Information Technology, Islamabad

In partial fulfillment

Of the requirement for the degree of

Bachelor of Science in Computer Science (2017-2021)

By

Ali Raza CIIT/FA17-BCS-081/SWL

Husnain Bashir CIIT/FA17-BCS-100/SWL

Hamza Siddiqui CIIT/FA17-BCS-103/SWL

DECLARATION

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software and accompanied report entirely based on our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Ali Raza

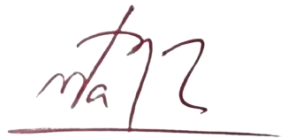
Husnain Bashir

Hamza Siddiqui

CERTIFICATE OF APPROVAL

It is to certify that the final year project of BS (CS) “Project title” was developed by.

Ali Raza (CIIT/FA17-BCS/081), Husnain Bashir (CIIT/FA17-BCS/100) and Hamza Siddiqui (CIIT/FA17-BCS/103) under the supervision of “Dr. Mazhar Sadiq” and co supervisor “Sir Waqar” and that in (their/his/her) opinion; it is fully adequate, in scope and quality for the degree of Bachelor of Science in Computer Sciences.



Supervisor

Dr. Mazhar Sadiq

Assistant Professor

External Examiner

Head of Department

(Department of Computer Science)

Executive Summary

In modern era we go through thousands of images in single day. Now a days communication considers more effective when it involves image stories more than the traditional wordings. Because a picture spoke more than the words. For making these pictures stories, all the social media and electronic media temper those images according to their need. So near about 80 to 90 % images on the social media and electronic media are considered as forged images. But where are the problems, there are the solution. Now there are number of the techniques that provide the efficient detection of the forged image. So, images forgery techniques provide the way to solve this problem. Images play a vital part in number of fields like medical, courts, journalism, in scientific experiments. In medical, digital images use as MRI and CT scan. it also helpful in field of education. As we know image tempering process is on its peak in these days, so it increases the importance of authentication of image specially in the field of law and court because there are many images that use as evidence for any case in courts. Forgery is also done by adding or hiding some useful and meaning information in the original image.

In the proposed system, we firstly take the tempered image (forged image) that is using to evaluate. In which code resize the image due to their large size. The resized image is converted into gray image (RGB to gray) because it is necessary to convert the image into gray. Segmentation is always done on the gray image. After that, the code calculates the intensity value of the segmentation. Threshold is set to detecting the forged area of the image. In last, an array is created that contain segmented image, seed points and threshold value and then calculate all the values to detect forged area.

The techniques that explain here it provide the more accuracy about the detection. The proposed solution provides the detection of the copy move. We try to make it short and easy to understand. It is in fast processing and using less memory. It based on the segmentation technique. Which make it easy to detect and locate the forged area? It detects forged area in less time and in effective way.

Acknowledgement

This Project is like a bridge between theoretical and practical work. Keeping this, I mind we start this project. First, we would like to thanks to ALLAH almighty who give us strength to achieve our goal. Without his blessing we are not able to do this task.

We complete this project under the guidance of our respected and honorable supervisor Dr. Mazhar Sadiq. We are heartily grateful to our supervisor for providing valuable support and guidance throughout the project. His beneficial awareness brings affective results for our project. He always helped us to settle all those problems that happened in process of project. He taught us how to deal with errors that are occurring in our project code. So, with the passage of time, we get familiar with those errors and now we can solve any type of problems related to our project.

So, we are extremely thankful to our supervisor Dr. Mazhar Sadiq for his empathy, humble behavior, attention, support, and valuable guidance. Without his support and guidance, we are not able to complete this project. He listens all our queries and problems with patience. We are very grateful for his acceptance and patience during all discussion about project. It is result of his supervision that we have done this project in smooth manners. We were also like to thank to all other teachers and people who join us and helped us in our task. They taught us how a project can be present in effective manners. They work with us` and they help a lot in many ways. Last but not the least we extremely thankful to our family and friend for their caring, praying and sacrifices for us. They not only pray for us but also understand our serious work situation so they support us in different ways.so we can say that their countless support brings effective result for this project.

Ali Raza

Husnain Bashir

Hamza Siddiqui

Abbreviations

ML	Machine learning
AI	Artificial intelligence
ICP	Introduction to computer programming
OOP	Object oriented programming
GUI	Graphical user interface
CV	Computer vision
SE	Software engineering
IFD	Image forgery detection

Table of Contents

1. Introduction.....	2
1.1. Brief.....	2
1.2. Relevance to Course Modules	3
1.3. Project Background	3
1.4. Literature Review	4
1.5. Analysis from Literature Review.....	5
1.6. Methodology and Software Lifecycle for This Project	5
1.6.1. Rationale behind Selected Methodology	6
2. Problem Definition.....	9
2.1. Problem Statement.....	9
2.2. Problem Solution for Proposed System	9
2.3. Deliverables and Development Requirements.....	10
3. Requirement Analysis.....	12
3.1. Requirements Gathering Techniques.....	12
3.1.1. Functional Requirements	12
3.1.2. Non-functional Requirements	13
3.2. Use Cases Diagram	14
4. Design and Architecture.....	17
4.1. System Architecture.....	17
4.2. Data Representation	17
4.3. Process Flow/Representation	19
4.4. Design Models.....	20
5. Implementation	23
5.1. User Interface	23
5.2. Algorithm.....	25
6. Testing and Evaluation.....	27
6.1. Manual Testing.....	27
6.2. System testing	27
6.2.1. Unit Testing	27
6.2.2. Functional Testing.....	28
6.3. Project Progress Overview	29
7. Conclusion and Future Work	32
7.1. Conclusion	32
7.2. Future Work.....	32
8. References.....	33

Table of Figures

Figure 1: Software Process Methodology	7
Figure 2: Use Case Diagram	14
Figure 3: Architectural Diagram	17
Figure 4: Forgery detection sequence	17
Figure 5: History Sequence	19
Figure 6: Accuracy Sequence.....	19
Figure 7: Flow chart.....	21
Figure 8: Class Diagram	21
Figure 9: Dashboard Screen	24
Figure 10: Start processing of the image.....	24
Figure 11: History Screen	25
Figure 12: Final Output Screen	25
Figure 13: User Help Screen.....	26
Figure 14: Feature Extraction.....	31
Figure 15: Image binary mask.....	31
Figure 16: Forged image	32
Figure 17: Result.....	34

Table of Tables

Table 1: System Comparison	4
Table 2: Use case 1	15
Table 3: Forgery detection sequence description	18
Table 4: History Sequence Description.....	19
Table 5: Accuracy Sequence Description	20
Table 6: Unit Testing - Insert image	28
Table 7: Unit testing - Process image	28
Table 8: Unit testing - Showing results after processing.	28
Table 9: Unit testing - Check image accuracy	29
Table 10: Unit testing - View history.....	29
Table 11: Functional testing - Insert image.....	29
Table 12: Functional testing - Accuracy check.....	30
Table 13: Functional testing - Complete history.....	30
Table 14: Functional testing - Output	30
Table 15: Single image accuracy checks	34
Table 16: Complete image folder accuracy checks.....	35

CHAPTER NO. 01

INRTODUCTION

1. Introduction

In these days, the manipulation of digital image has become easy due to availability of affordable and powerful image editing software such as adobe Photoshop. This thing is very difficult to understand for the human that the image is original or fake, now a day it will increase rapidly, it means image manipulation is common in the current days. These manipulated images can be uses to gain or make false propaganda. So, the verification of the integrity of digital image and authenticity of digital images became very important for this purpose we use image forgery detection to detect that the image is real or not. In courts of laws digital images are not accepted as the evidence without any forensic analysis. That is why, the forensic analysis of the digital images is very important because it can clarify the authentication of digital image. Image forgery detection is a process of proving that the image is real or fake. We can detect a type of passive approach in Image forgery detection. In passive approach there are two types of image splicing and copy move. In copy move one part of the image cut and paste in the same image. We can detect copy move by using machine learning techniques. In machine learning we use KNN algorithm to detect copy move in any copy move there is a correlation between the original image area and the forger image area. KNN can be used to detect the forged area in the image. With help of KNN we can detect copy move forgery in an image.

1.1. Brief

With the advancement of other technologies, digital world is also changing its behaviors. High resolution digital cameras are easily available in markets. Everyone wants to capture high quality pictures, for these purposes they use high quality cameras and other image editing software that makes their image more pleasurable and attractive for the viewers. This editing software contains numbers of filters for manipulating and modifying the image. These filters give a professional look to any images. These are easily purchasable so everyone in the world is using this editing software's. Where there are some benefits, but it has many disadvantages.it creates many social issues like doctoring of pictures of any person to improve it quality or making some fake pictures of that person by temper it with some other immoral pictures. Anybody forms anywhere can make a forged image of any type. So, these filtering techniques have put a question mark on the trustworthiness of the images. That is why it enhanced the demand of forgery detection techniques.

In modern era we go through thousands of images in single day. Now a days communication considers more effective when it involves image stories more than the traditional wordings. Because a picture spoke more than the words. For making these pictures stories, all the social media and electronic media temper those images according to their need. So near about 80 to 90 % Images on the social media and electronic media are considered as forged images. So, images forgery techniques provide the way to identify the authentication of any image. There are the many peoples who are working on different forgery detection technologies. All these technologies and techniques have different ways to detect and localized any tempering in image. These forensic analyses provide help to identify which image is real and which one is forged. There are also a wide range of image tempering techniques. It could be copy-move, image splicing, image resembling, retouching, image enhancing, morphing and computer generating etc. all these have separate methods to dealing with images it could result in number of unauthenticated images and create many issues and propagandas regarding trustworthiness of an

image. There are different approaches to detect the forgery in the image. In our system we are using Discrete Cosine Transform (DCT) for the block representation which is very helpful to detect the forgery in the image because it has high pass filtering and the result of DCT is more accurate as compared to any other. We are also using Principal Component Analysis (PCA) to detect the similarity between the duplicated regions we can use PCA for exploiting the feature of the image PCA uses the square blocks. The forger image consists of overlapping rectangular or circular blocks in forgery detection first we can divide the image overlapping rectangular blocks without considering circular blocks then we can reduce search area and make search unit as robust as possible for the post processing like scaling compression and Gaussian's noise after that all the pixels are sorted then we can compare these pixels and detect the duplicated region.

This system is very helpful there are a lot of benefits of this system. This interface of this system is user friendly and easy to use some of the most important benefits are given below.

Maintain history: When the user can detect the forgery in the image it can automatically save the image in history table so the user can access it later when he needs it again.

Safe from Blackmail: if a person can blackmail someone by tampering his image, we can detect that the image is real or fake and save the person from blackmail.

Court of law: In court of law image are presented as evidence so it is very difficult to check that the image is real or tamper. We can use this software to check that the image is real or fake and verify the evidence.

The result of this system is not 100%. Because it is very hard to detect smallest area of the digital image. The accuracy of this system is 80 to 85 percent. The software does not detect that the image is forger if there is natural duplication present in image it means the duplication not inserted by using any tool. This software only detects the forger part of the digital image if the image is tampering by using software tool. It is more difficult to detect the forger part in the digital image if there is a lot of noise in the image.

1.2. Relevance to Course Modules

The courses which I studied in my Bachelor of Science in computer science degree which is related to my project are ICP (introduction to programming), OOP (object-oriented programming), DATA STRUCTURE, DATA BASE, Software Design Methodology, Web Artificial Intelligence, and Machine Learning. We learn different algorithm of machine learning and artificial intelligence which helps us in making our project. In software design methodology we had learn iterative method which helps can be uses in our project. Through introduction to computer programming, we learn the basics of programming in languages that is data types, pre-processors directives, conditional statements, switch statement, loops, and pointers etc. We can use all of these in python and Django we had learn python in artificial intelligence lab and these basics help me in making our Web application. In object-oriented programming I learn the terms classes, inheritance, polymorphism and these all terms uses while coding in python. And these all terms help me a lot while doing my project in data structures. I learn all the data structures that are using in our project in making my application which are array, and matrix.

1.3. Project Background

In the present era millions of images are forger in a day we can see these types of images daily on different platform of social media some people can use these types of images for false

propaganda. These images can also destroy an innocent person life. To save someone life the detection of false images is very important. There are multiple types of false images we can make by using different editing tools. Now, these days a person can manipulate the images by using the mobile phone, so we must detect these images. There are two types of image forgery detection techniques that are active approach and passive approach. In active approach we embed watermark or digital signature with the image. It is very difficult to detect these types of forgery there is no certain software available in the market that can detect active forgery. The next technique of image forgery detection is passive approach. In passive approach we will discuss copy move forgery in detail.

Copy Move: In copy move we cut single and multiple part of the image and paste on the other part in the same Image. Our software can detect the copy move if the copy move can be done by using any editing tool. Our system will not detect the natural duplication it means that the copy move forgery does not exist in the picture.

There are many approaches that are used to detect the forgery in the digital image. There are multiple algorithms that are purposed by different scientists that helps to detect the temper area in the image. By using these algorithms, we can detect the temper area of the image. In image forgery detection first, we choose the image then pre-process it in pre-process we convert the image into grayscale image then we perform segmentation in segmentation we convert the in image into smaller parts then we perform feature extraction. In feature extract all the feature of the image and match all the part of the image. In matching all the pixels of the image should be match in matching if the pixels of the image cannot be match to some pixel, then the forgery is detected. The part of the forgery area can be highlighted. In this way we can detect the pixel base image forgery detection

1.4. Literature Review

There is multiple software that are available in the market for image forgery detection. The weakness of these software is not user friendly. Our software has good interface and very easy to handle the user just must upload the picture and detect the forgery in the image. Other systems do not save images as history our system can also store images as history. Other software just detect that the image is forgery or not our system can also detect the location of the forgery in image.

Table 1: System Comparison

Application Name	Weakness	Proposed Project Solution
Passive Copy-Move Detection System (htt)	<ul style="list-style-type: none"> • It only detects copy-move forgery. • It does not provide user friendly GUI. 	<ul style="list-style-type: none"> • It can detect copy-move with high accuracy. • A Graphical User Interface will be providing for ease of user.
A fuzzy approach to deal with uncertainty in image forensics	<ul style="list-style-type: none"> • It does not provide a complete system with GUI. 	<ul style="list-style-type: none"> • A complete forgery detection application will be provided.

(htt1)	<ul style="list-style-type: none"> • It does not use Artificial Intelligence. 	<ul style="list-style-type: none"> • Results of system will improve by using Artificial Intelligence
Region Duplication Forgery Detection (htt2)	<ul style="list-style-type: none"> • It only detects duplication of area. • It does not use Artificial intelligence. • It only accepts JPEG format images. 	<ul style="list-style-type: none"> • It also maintains history for user. • Artificial Intelligence will be used to detect forged area of image. • All format images will be accepted by system

1.5. Analysis from Literature Review

Fake images are increases day by day. People can manipulate fake images by using different editing tool and then they can upload these images on internet or can be uses to blackmail someone the identification of these fake images are very important so the person can verify that the image is real or fake. To detect the forgery in the image we use multiple software, but this software is not easy to use and does not have good interface or they can detect only copy move or splicing. The accuracy of some of the software is less than 65% they cannot properly highlight the forger area. We are going to develop the software which is more accurate and can detect copy move with more accuracy because for accuracy we can compare system generated masks and dataset mask, so the accuracy of our software is more than 90 percent. Our software has good interface that user can very conveniently use other software does not maintain the history of the images our software can maintain the history so; the user cannot have to load the image again and save his loading time which is very beneficial for the user. This type of software is very beneficial in court of law where proof is very important to check that the image is real or fake because people can forger images and represent as proof so the authentication of that kind of images is very important. Our software can check multiple image formats like PNG, JPG, JPEG and if the image is authentic, it does not highlight any forger area but if forgery exists in the image the software will detect that part and highlight that region.

1.6. Methodology and Software Lifecycle for This Project

We can get an image from the user, and then our software converts the original image into a grayscale image; the grayscale image is black and white. After getting the image from the user, our software resizes it according to by default size set in the software. Then convert the image into a segmented image, segmentation is the process that can divide the image into multiple segments or a set of pixels. After this process, we can move toward the next step, that is feature detection. In feature detection, the convenient features are extracting from the desired pixel of the image and then pixel behavior is analysis with its neighboring pixel. In the end, our software shows the picture is forged or not and highlights the forged part of the image. So, it can be used as base for searching duplicate region. This duplicated region is detected and localized by the following three steps.

- **Feature Extraction:**

In feature extraction the suitable features are extracting from the desired pixel of the

image and then pixel behavior is analysis with its neighboring pixel. In presence of the rotation invariant and scale features are extracting through some suitable algorithm, which is circular harmonic transform algorithm. This algorithm provides number of possibilities to be tested as, ZM, polar and cosine transform. For the scaling invariant FMT is done which is completely based on the log-polar sampling.

- **Matching:**

The best matching feature are selected for the further process. Matching is one of the most important and time requiring phase. So, reducing it time a fast technique is adopted which is nearest neighbor field technique. Matched part on the image is searched in the nearest field from the specific portion of the image.

- **Post-processing:**

In post-processing, an offset is created which link the pixel with its forged pixels. After that it is filtered so that the false alarm can be reduced.

1.6.1. Rationale behind Selected Methodology

We use Incremental model for the development phase of our project in the model the product is formed by increments. By using this model in our project, we can backtrack at any point with the help of this model we can enhance the quality of our product. The phase that we use in our project are given below:

Phases of model:

- Requirement specification
- Designing
- Implementation and coding
- Integration
- Testing
- Maintains and installing.

In this Model an overall architecture of the total system is developed first, the detailed increments and releases are planned. Each increment has its own complete life cycle. The increments may be built serially or in parallel depending on the nature of the dependencies among release and on availability of resource. Each increment adds additional or improved functionality to the system.

- It is easy to test and debug the product during iterations.
- Software released in increments over time is more likely to satisfy changing user requirements than if it were planned as a single overall release at the end of the same period.

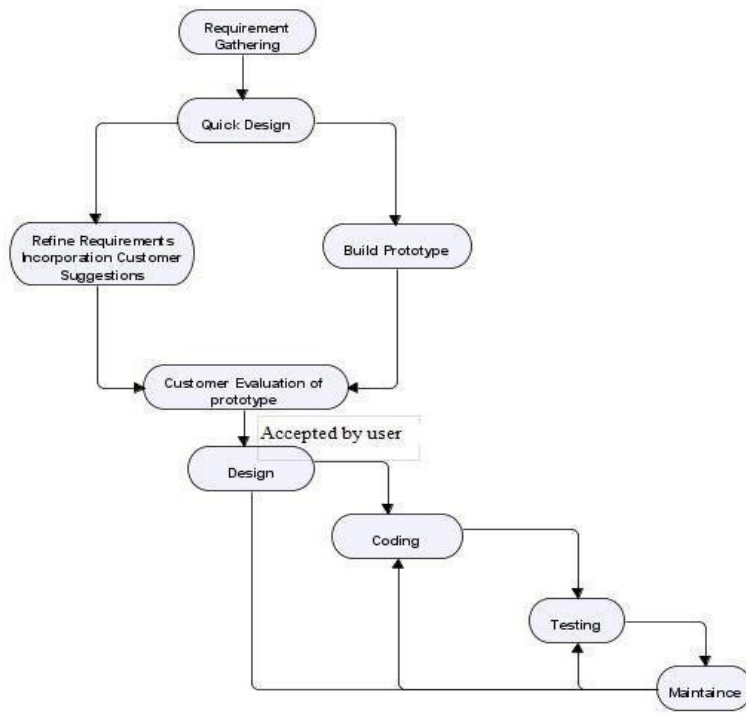


Figure 1: Software Process Methodology

CHAPTER NO. 02

PROBLEM DEFINITION

2. Problem Definition

In this chapter we can discuss about how we can solve this problem. If our problem statement is simple and any one can easily understand it, then our team member can understand the things easily and work together on developing the solution of the problem.

2.1. Problem Statement

Images play a vital part in number of fields like medical, courts, journalism, in scientific experiments. As we know image tempering process is on its peak in these days, so it increases the importance of authentication of image specially in the field of law and court because there are many images that use as evidence for any case in courts. Forgery is also done by adding or hiding some useful and meaning information in the original image. These forged images look like the original image. It may be very difficult to identify the difference between real and forged images. So, when these forged images are provided in courts as evidence its complete change the behavior of the case. Where there are the problems, there are the solutions. Now there are lots of technologies that are used to identify forged images. But it is still a problem to detect the exact location of forgery as well as type of forgery. There is need of such techniques that talk about which type of forgery is done in image like copy move, cloning. Morphing, retouching, and splicing. Moreover, it is also very important to provide user a complete interface on which he/she passed its forged image, and the system will provide him/her a complete solution to that forged image like forged area of the image.

There are different techniques that are used to detect image forgery. But another question arise here is which techniques is more effective and acceptable. Which will provide more authentic and quick result of a forged image? For example, each forgery has number of solutions like copy- move forgery detection can be done by the number of different technologies. Each has their different behavior. But everyone wants to adopt those solutions that provide result more accurately in less time and less effort.

2.2. Problem Solution for Proposed System

By using image forgery detection system, we can detect the forgery in the image. Our system also detects where the forgery exists in the image. Our systems are user friendly, and the interface of the system is very easy to use. We can give an image to the system then our system can detect the forgery in the image. The system has also the option of history in which we can store our forgery detected images if the user wants to see those images in which he has detect the forgery he can easily see these images in the history. We use a type of passive approach to detect image forgery. In passive approach we can detect copy move forgery. Copy move forgery is one of the most common used image forgery technique. In copy move we cut single and multiple part of the image and paste on the other part in the same Image. Our software can detect the copy move if the copy move can be done by using any editing tool. The system does not detect the forgery in the image if there is natural duplication present in image it means the duplication cannot be inserted by using any tool. Our system can also check accuracy of the forgery. For this purpose, we can compare to masks one mask is the original dataset mask and one mask is the system generated mask our system can compare these two masks and check the accuracy.

2.3. Deliverables and Development Requirements

- **Hardware requirements**

Hardware requires for the project is.

- SSD hard drive
- Graphic card
- Core i5 4th generation or above

- **Software requirements**

Any operating system some of them mentioned below.

- Windows
- Linux
- Database

CHAPTER NO. 03

REQUIREMENT ANALYSIS

3. Requirement Analysis

In this chapter we can discuss about the requirements and show to analyze those requirements. In simple word this chapter is all about the requirements analysis. The best way to get the proper requirements you can ask the questions directly from the clients and note down these question that will help you when you design the system as per your client requirements. Well, we can focus on both functional and non-functional requirements. To define project schedule and processing, different models and techniques also focused on this chapter.

3.1. Requirements Gathering Techniques

Most probably techniques that can be used for collecting requirements are as follows:

- By survey and interviews
- Use different software tools for this purpose.
- Use different techniques that can help for the decision making.
- We can also use prototyping for this purpose.
- By observations

There are many ways to collect the requirements like conduct meetings with the student, teachers, and different companies' managers. In requirement analysis portion we can only focus on the problem and its solution, and the most important thing is how to make it a proper application/system. Requirement's analysis is further divided into two parts:

1. Functional Requirements
2. Non-Functional Requirements

3.1.1. Functional Requirements

This system will have following modules:

1. Start Test
2. Gray Scaled Image
3. Segmented Image
4. Feature Detection
5. Tempered Image
6. Help
7. History/Reports
8. Clear/Reset

Start Test:

In this module, user will browse an image from his/her PC and upload to the system. After that user will start test.

Grey Scaled Image:

A grey scale image is colorless image. Its colors based on only black and white colors and their different shades. So, this module will display the grey scaled image of user input image. User can print it.

Segmented Image:

At this step of image forgery detection, an image divided into pixels or small blocks called segments. Segmented image will display in this module. User can print it.

Feature Detection:

In this module, all features of image will examine and divide them based on different signs or colors. After that display output to this module. User can print it.

Tempered Image:

After completing the test, the system will highlight the tempered area of image and then display in this module. This is the final output of the system. User can print it.

Help:

This module contains different user manuals for user help.

History/Reports:

This module is connected to database. All previous tests saved in database display in this module. User can print reports of the history.

Clear/Reset:

Results from all windows will be removing and system will be ready to take a new test.

Accuracy:

We can also add the accuracy module with the help of this module you can easily check the accuracy of the image mask, like we can compare system generated image and result image to check the accuracy of the image as well as we can also check the complete folder accuracy at the same time.

3.1.2. Non-functional Requirements

This system can be easily processing a high quality as well as low quality image. Attributes of a system is very important for maintaining its performance. So, high quality attributes are provided to system to match with user requirements and provide best results in most reliable and suitable manners. These attributes help system for generating best possible results, maintain system usability and performance to the maximum level to provide best services to users.

Usability

Usability mean how easy interface are to use. Usability requirements deal with ease of learning, ease of use, error avoidance and recovery, efficiency of interactions, and accessibility.

- The system allows the user to upload image and see results with a single interaction.
- The system has simple user interface user can easily understand and start working with on it.
- It will also keep the record of user activities.

Design for ease for user

The user interface of this system is completely simple and easily understandable. It is design for the ease-of-user. It is based on the buttons through which the user can add the forged image. By clicking in the button's user will be able to see the detail of the forged images and will be able to see the localized area where forgery occurs. In other words, we can say that this interface provides forgery detection just in one clicking. This thing is really will be helpful for the user.

Performance

The system shows the result within the reasonable time frame depending upon CPU time, image quality and checking quality. When user upload the image to the system than it will take some time to process the image because it will pass from different phases to detect the forgery from image.

Rigid translation can be measured according to image-level and pixel-level with respect to CPU time. Measurement will automatically reject when CPU time will get longs.

3.2. Use Cases Diagram

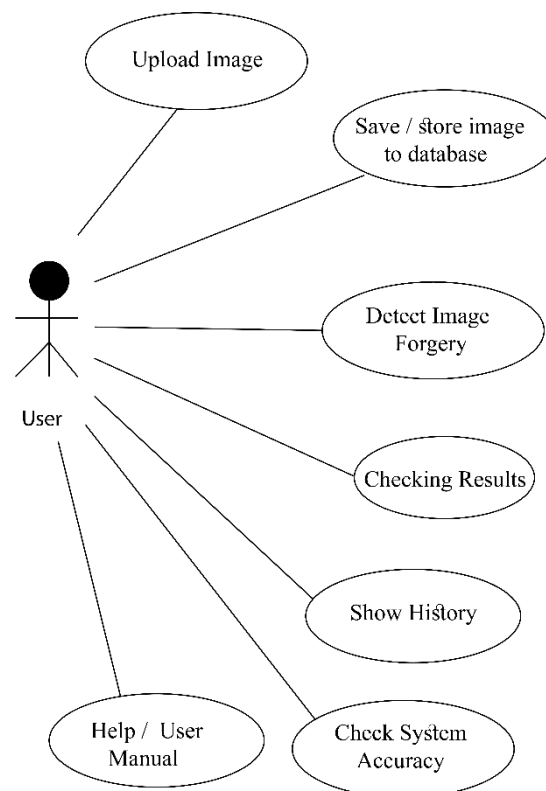


Figure 2: Use Case Diagram

Detailed Use Case

Following table illustrates our use case diagram.

Table 2: Use case 1

Use Case ID:	USE-1
Use Case Name:	User Interaction
Actors:	Primary Actor: User Secondary Actors: System
Description:	A user starts a test for image forgery, after completion of test; he/she can check the results. History will maintain as a sample test case. Moreover, he can also check accuracy of the system.
Trigger:	Image forgery detection needs in multiple fields like crime detection, image forensic department, etc. In security purposes, it would be very helpful to check forged area in an image.
Preconditions:	PRE-1 Model should have been properly trained and tested before being deployed.
Post conditions:	POST-1. User successfully gets his reports. POST-2. User gets result about uploaded image. POST-3. User gets some help in case for any query.
Assumptions:	ASUM-1: The system will be used by forensic department and professionals. ASUM-2: Image uploaded may have forgery type supported by system. ASUM-3: System will be used for forensic, security and research purposes.

CHAPTER NO. 04

DESIGN AND ARCHITECTURE

4. Design and Architecture

4.1. System Architecture

First, the input data/images pass through data preparation phase where all garbage images will remove, then applying normalization on them.

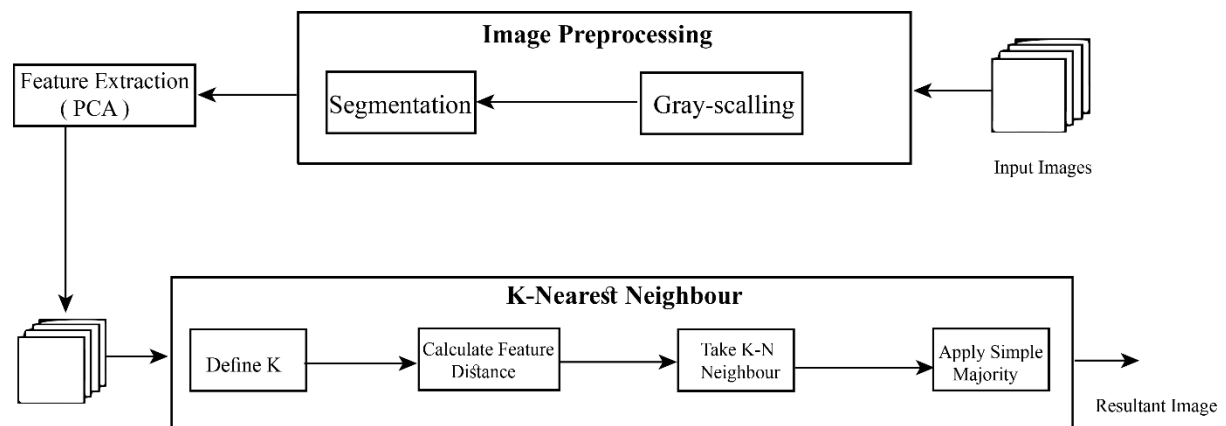


Figure 3: Architectural Diagram

After processing the images, the data will send to data models where KNN will apply on them to detect forged area. In KNN value of K-neighbors is decided, then calculate feature distance of. After that take K neighbors of the data and apply simple majority on that.

4.2. Data Representation

Sequence 1:

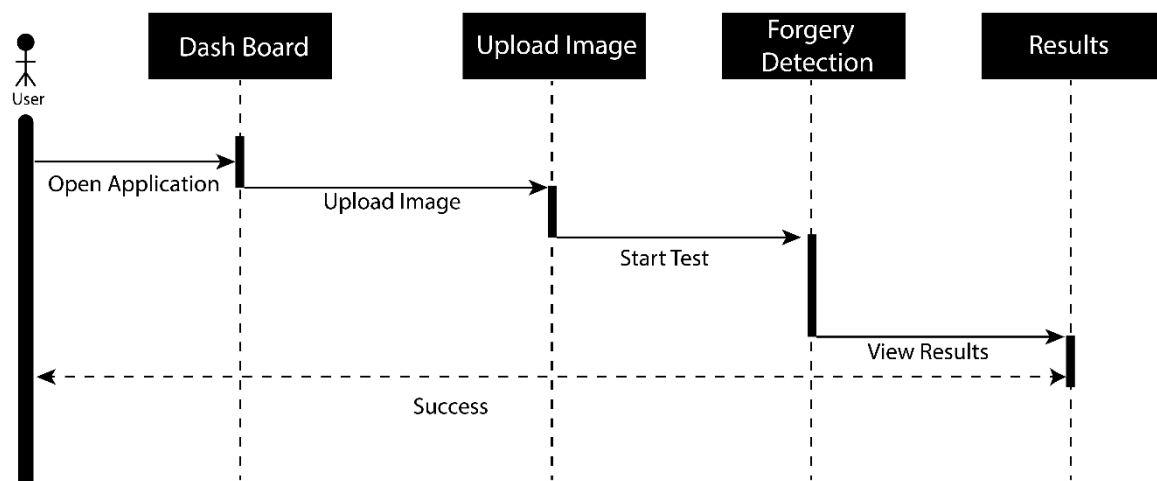


Figure 4: Forgery detection sequence

Description:**Table 3: Forgery detection sequence description**

Sequence ID	User Sequence
Sequence Name	Forgery Detection Sequence
Actor	User: Only singly primary actor accessing all part of system
Description	A user starts a test for image forgery by uploading different real and forged images to the database history and start processing to check forgery, after completion of test user can see the results.
Normal Flow	<p>Under normal circumstances flow will be:</p> <p>User will browse image for forgery test and upload it to database history. Click on “Next” button and then on “processing” button.</p> <p>System will Covert image to grey scaled image</p> <p>Convert to segmented image.</p> <p>All features will be detected.</p> <p>Result will be set on its screen.</p> <p>Check forged image (tempered image) Clear</p>
Alternative Flow	None

Sequence 2:

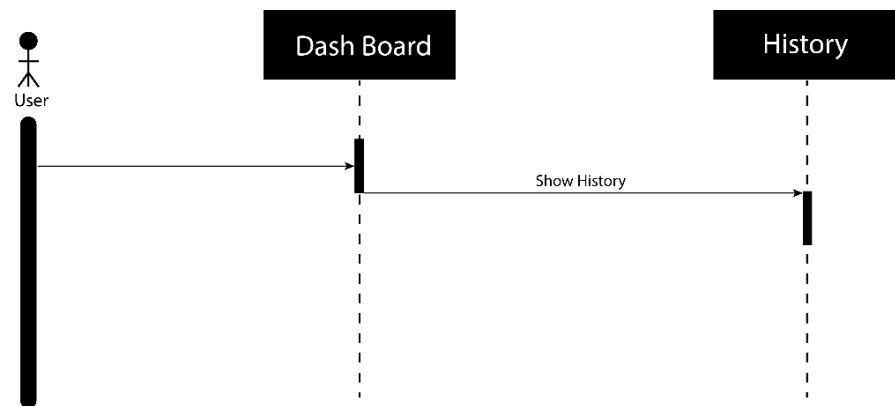


Figure 5: History Sequence

Description:

Table 4: History Sequence Description

Sequence ID	User Sequence
Sequence Name	History Sequence
Actor	User: Only singly primary actor accessing all part of system
Description	A user wants to view his history of previous checked images.
Normal Flow	<p>Under normal circumstances flow will be:</p> <ul style="list-style-type: none">• User will open the application or go to dashboard.• Click to history button and system will show him his history.• Load any results of image forgery Clear.• After that, user can print report of his history.
Alternative Flow	<ul style="list-style-type: none">• User will open the application or go to dashboard.• Click to history button and system will show him his history.

Sequence 3:

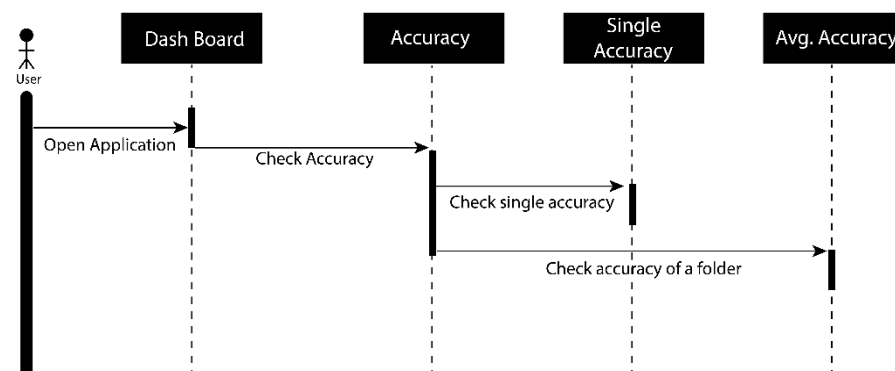


Figure 6: Accuracy Sequence

Description:

Table 5: Accuracy Sequence Description

Sequence ID	User Sequence
Sequence Name	Accuracy Sequence
Actor	User: Only singly primary actor accessing all part of system
Description	A user wants to check accuracy of the tested image or accuracy of the complete folder.
Normal Flow	<p>Under normal circumstances flow will be:</p> <ul style="list-style-type: none">• User will open the application or go to dashboard.• Click to accuracy button. <p>To check single accuracy.</p> <ul style="list-style-type: none">• Browse for two masks one for system generated results and second from sample masks.• Click on Results to check accuracy.
Alternative Flow	<ul style="list-style-type: none">• User will open the application or go to dashboard and click on accuracy button. <p>To check folder accuracy</p> <ul style="list-style-type: none">• Click on Results button under the average accuracy pane.

4.3. Process Flow/Representation

Flow of process will be such that, the user starts the application, goes to dashboard, upload image for test, preprocessing techniques like scaling, grey scaling, segmentation, and feature extraction, will apply to image. After that, image will pass through post processing techniques like features matching and forged area detection, after that forged area will be detected and results will show to the user. At last, the results will save to memory for next time. User can also select history directly from dashboard without tasting any image.

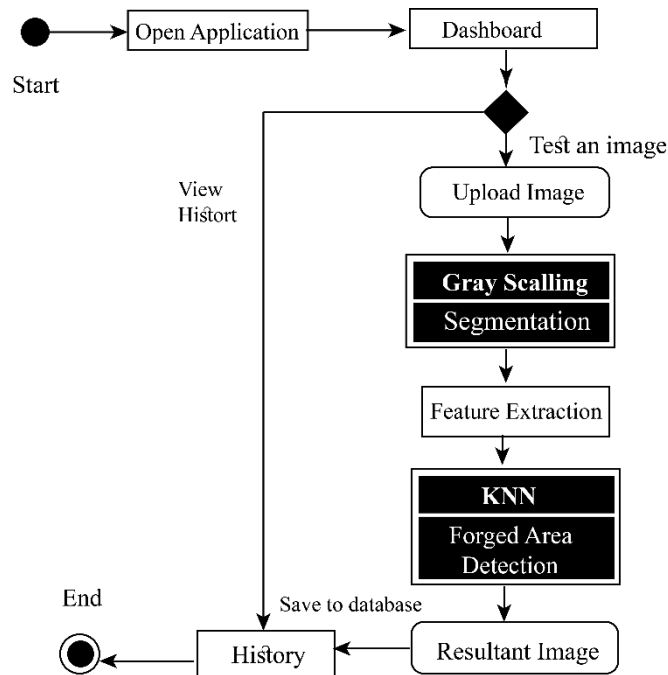


Figure 7: Flow chart

4.4. Design Models

There is following class diagram for purposed system that is following classes, attributes, operations, and their relationships.

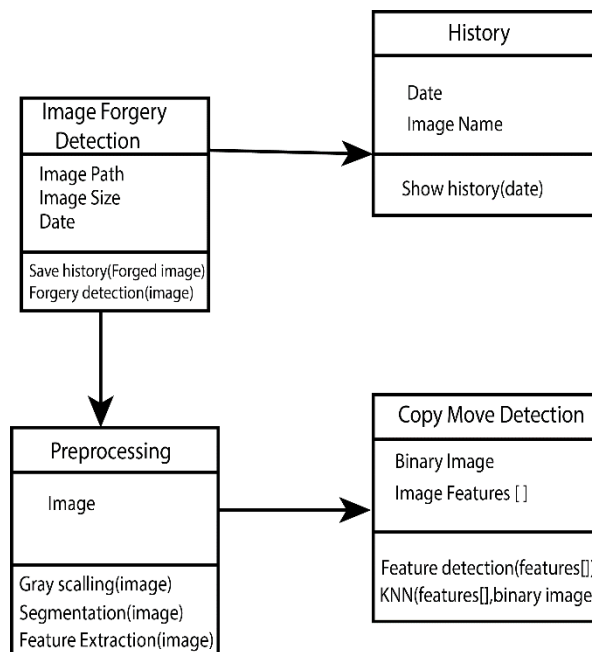


Figure 8: Class Diagram

Description:

Here is total four classes in the system.

1. Image Forgery detection
2. Preprocessing
3. Copy Move detection
4. History

1. Image forgery detection:

This is the main class which is starting the program.

Attributes: Image path, image size, date.

Function: Save history, forgery detection.

2. Preprocessing:

In this class parameters for preprocessing of an image are completed.

Attributes: Image.

Functions: Gray scaling, segmentation, feature extraction.

3. Copy move detection:

In this class, copy move forgery of the image is detected. It is belonging to post processing of image.

Attributes: Binary image, image features.

Functions: Feature detection, KNN.

4. History:

This class is used to show previous history of the user.

Attributes: Date, image name.

Function: Show history.

CHAPTER NO. 05

IMPLEMENTATION

5. Implementation

In this chapter, we can discuss the implementation of our system. In the implementation phase, we can also focus on the front-end and the backend of our project. For the front-end, we can use those languages HTML, CSS, JAVASCRIPT, and the last one is BOOTSTRAP to make our front-end responsive. Now finally, we can move on to the backend portion. We can use python for the backend. We can choose the python framework Django for this purpose. Django is the best framework that can be compatible with the python programming language that why we cab chooses it.

5.1. User Interface

The user interface is made as simple as possible so it can be easy for the users to use it. Some pages of the user interface are further explained along with images below:

Dashboard

This interface will be the first page. When any user can run the project, then this page will appear first. We can show the different user options that can check our tempered image quickly without facing any problem on the dashboard page. In the first step, the user can upload the image and hit on the execute button. When the user hits the execute button, then the machine learning algorithm starts working in the image. It means they can detect the forgery from the image.

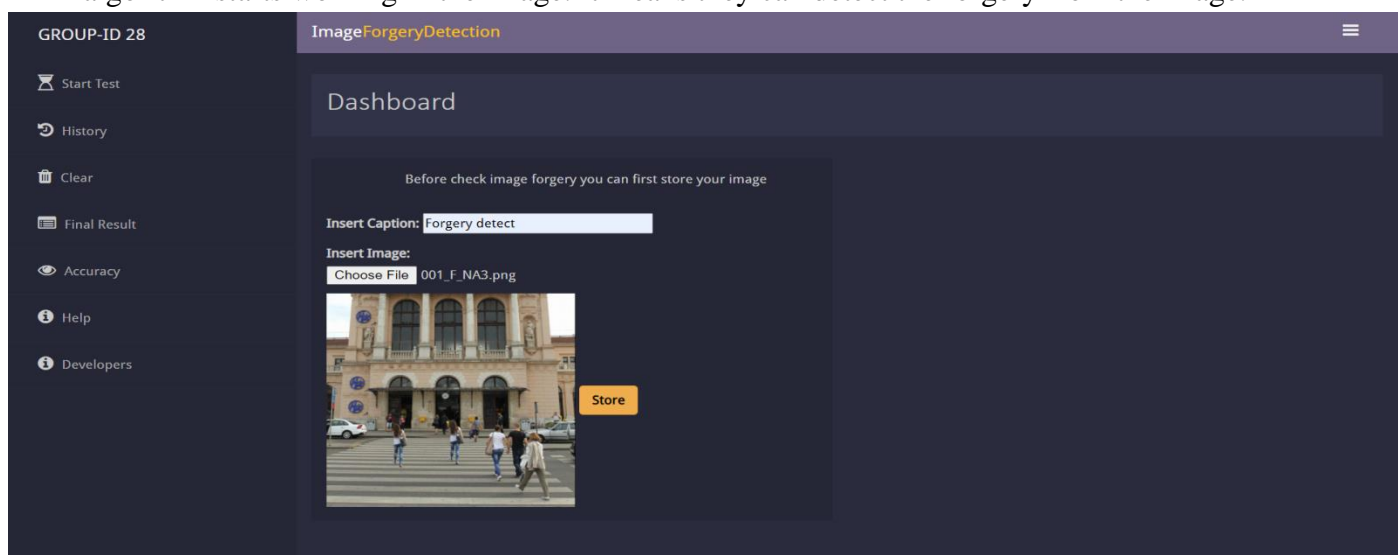


Figure 9: Dashboard Screen

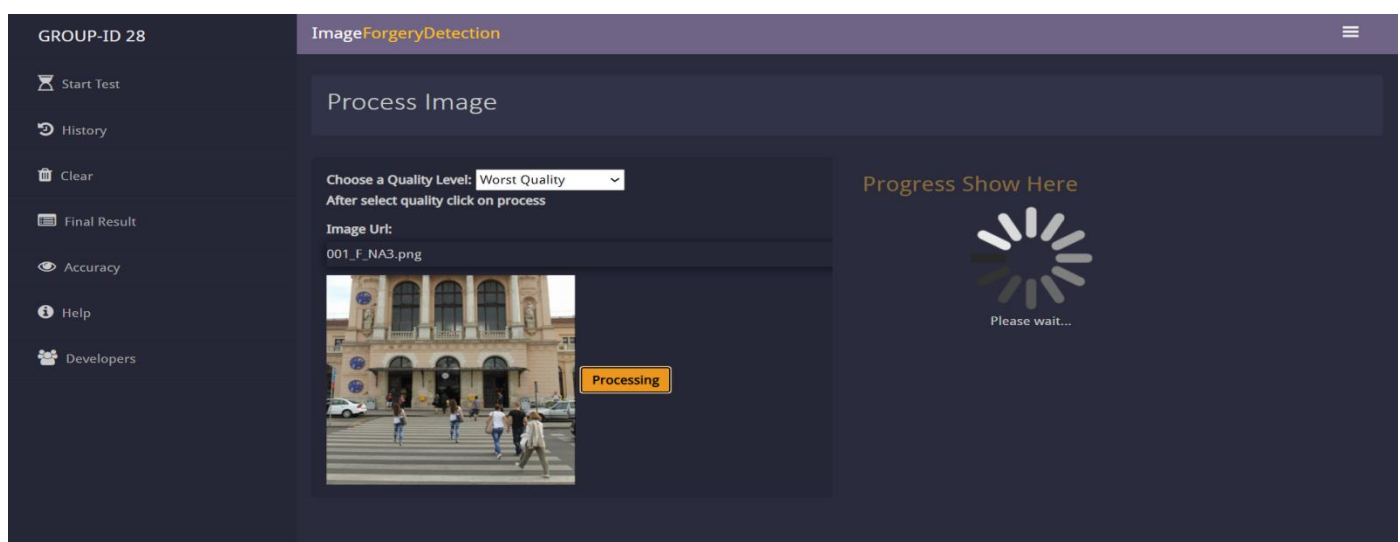
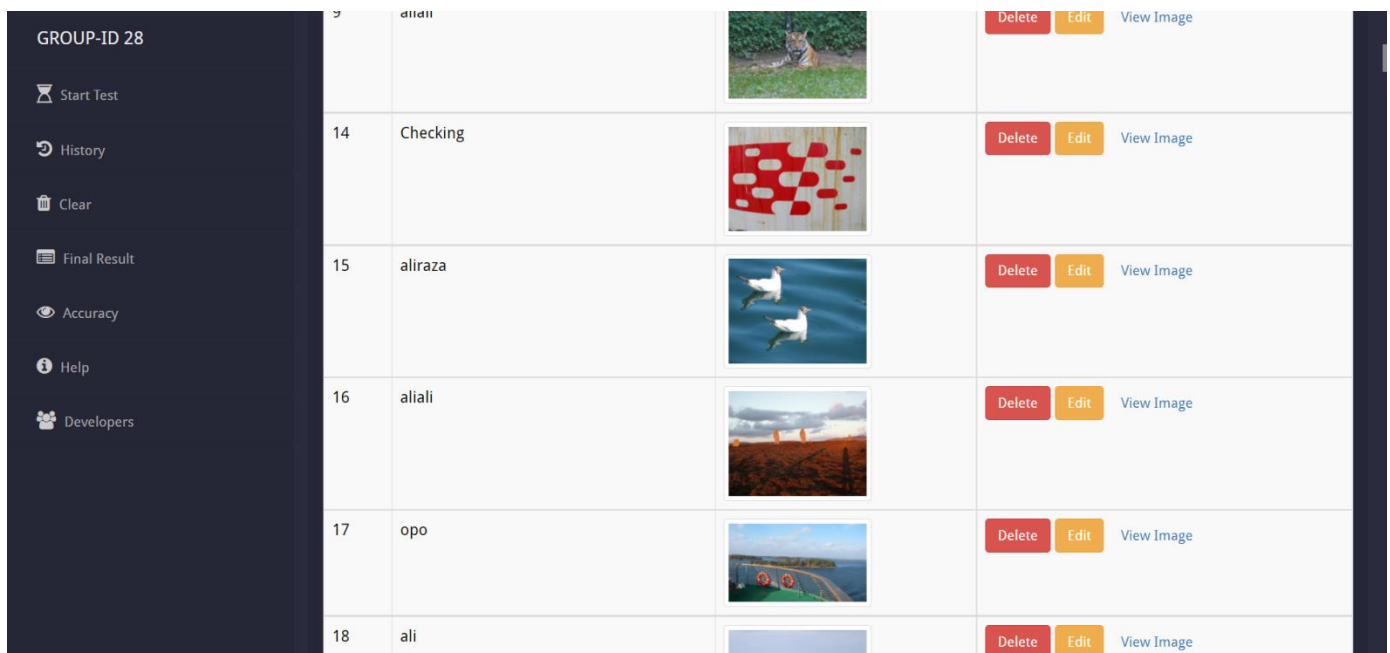


Figure 10: Start processing of the image.

Users can also watch his history; we can add some extra features to our project; the history option is one of them. In this portion, the user can check how many image images it can check in his past; in simple words, this option will make the record for the user; if they need any image in the future, they can check from the history portion.









ID	Name	Image	Actions
	alirazi		Delete Edit View Image
14	Checking		Delete Edit View Image
15	aliraza		Delete Edit View Image
16	aliali		Delete Edit View Image
17	opo		Delete Edit View Image
18	ali		Delete Edit View Image

Figure 11: History Screen

If the user clicks on the result button, our interface redirects the user to the result page. On this page, users can see the result of the forged image that they can enter. It will also highlight the area in which the forgery has appeared.

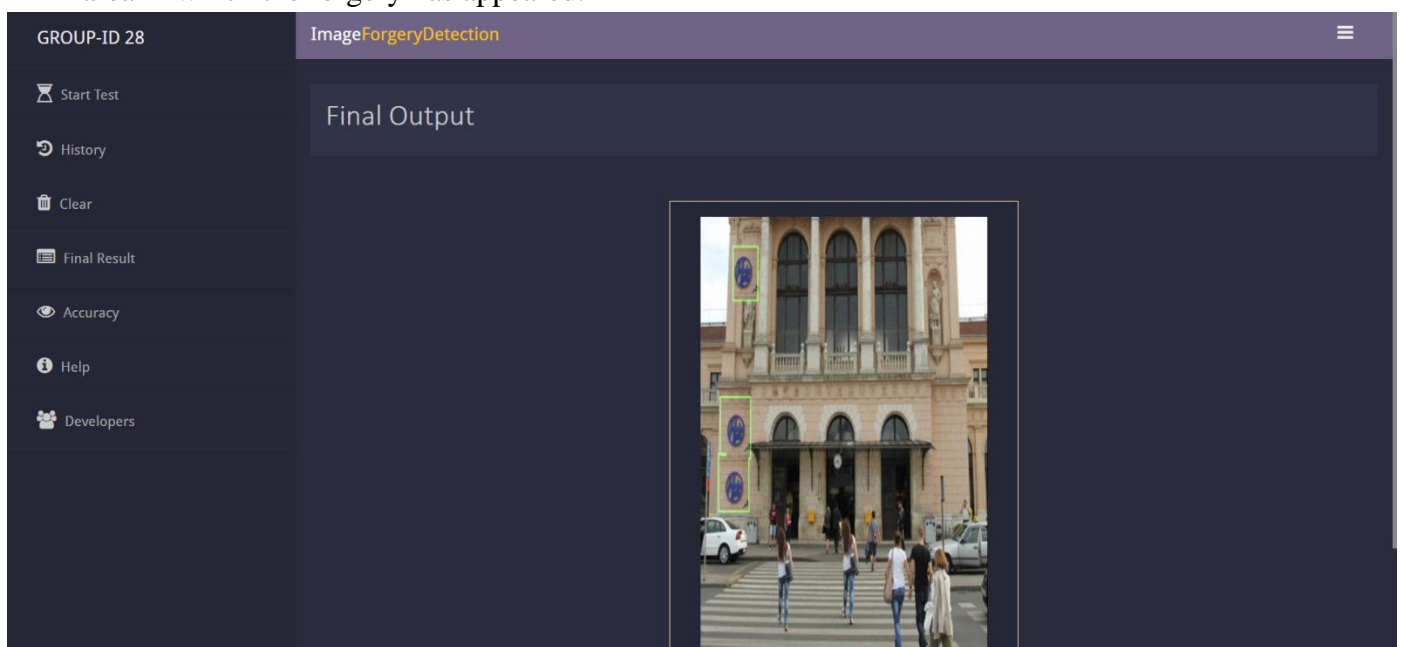


Figure 12: Final Output Screen

If the user cannot be aware of our system that they can click on the help button. When they can click on this button guide page will show. On this page, we can mention how a user can use our system and quickly done his required work.

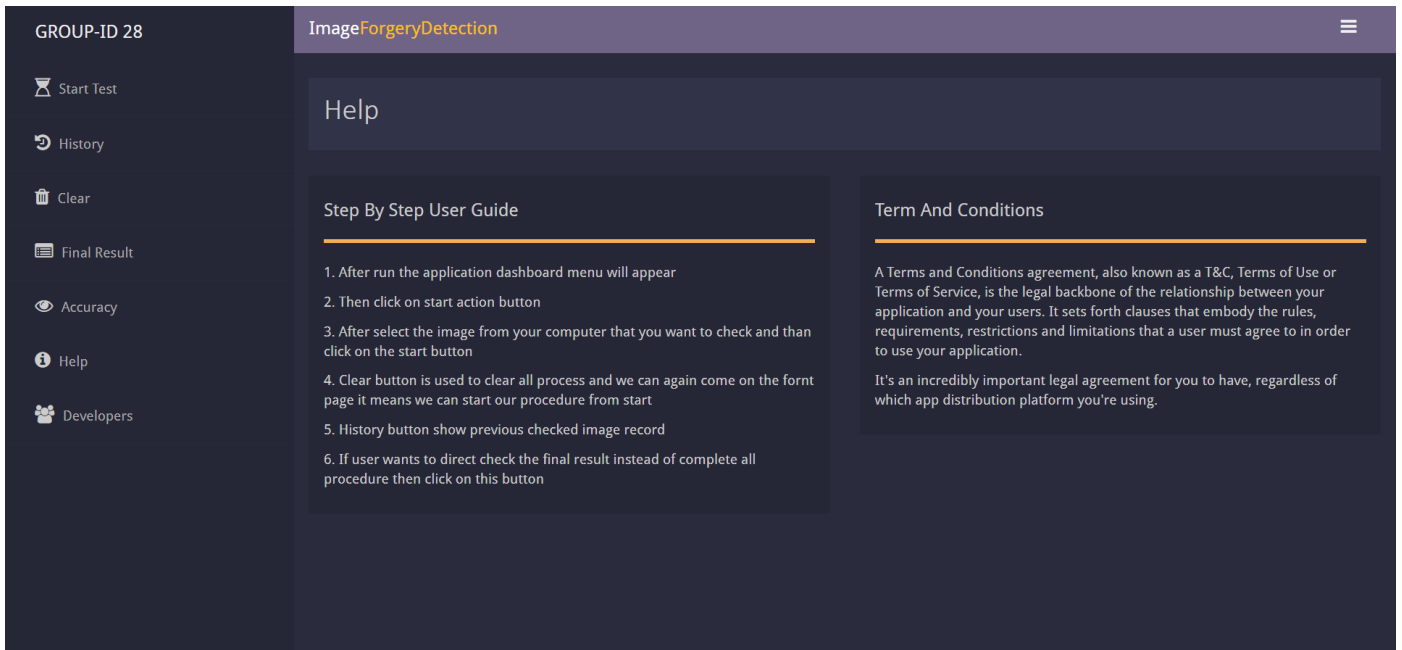


Figure 13: User Help Screen

5.2. Algorithm

To process on image and detect the forgery of the image we can use the KNN algorithm. In the 1st phase is image pre-processing in this phase resize the image, convert it into gray scale image and finally remove the noise from the image. 2nd phase is segmentation, in segmentation divide the image into different segments or pixels to identify the issue from the image. 3rd phase is to extract the feature it means software can check each pixel of the given image, if there is found any change in the image, they can highlight it. Because all editing done on the foreground of the original image, if there are any changes occur it will affect the pixels of the image and software easily identifies the changes and highlights it. Last phase is to highlight the forged part of the image that can help the user to check in which part of the image changes are occurring. It can highlight it with rectangular box. After highlight, the area user can easily check.

CHAPTER NO. 06

TESTING AND EVALUATION

5. Testing and Evaluation

5.1. Manual Testing

5.2. System testing

When our system is completely built, then our next step is to test the system. It will work properly or not as per our requirements because system testing will help determine the system's error and fix those errors. Here are the few types of the testing like unit testing, functional testing, and integration testing. We can complete the testing phase before delivering the product; we can ensure the thing is work properly.

5.2.1. Unit Testing

Unit Testing 1: Insert image into system.

Testing Objective: To ensure that image inserted successfully.

Table 6: Unit Testing - Insert image

No.	Test case/ Test Script	Attribute and Value	Expected result	Actual result
1.	We can check the image is inserted. successfully in the system or not	Image	Occur some errors in some. places	Pass

Unit Testing 2: Process image after insert.

Testing Objective: To ensure that system start working successfully.

Table 7: Unit testing - Process image

No.	Test case/ Test Script	Attribute and Value	Expected result	Actual result
1.	We can check processing on the image start or not?	Image	Occur some errors in some. places	Pass

Unit Testing 3: Show image results.

Testing Objective: To ensure that system shows the result correctly.

Table 8: Unit testing - Showing results after processing.

No.	Test case/ Test Script	Attribute and Value	Expected result	Actual result
1.	See the results of the image after processing	Image	Occur some errors in some places	Pass

Unit Testing 4: Check accuracy.

Testing Objective: To ensure that system show the image accuracy correctly.

Table 9: Unit testing - Check image accuracy

No.	Test case/ Test Script	Attribute and Value	Expected result	Actual result
1.	We can check the accuracy of the individual image as well as complete folder of the image. We can give the mask image to the system, and it can check the accuracy.	Image	Occur some errors in some places	Pass

Unit Testing 5: History

Testing Objective: To ensure that show history button is work correctly.

Table 10: Unit testing - View history

No.	Test case/ Test Script	Attribute and Value	Expected Result	Actual Result
1.	View history	History	Complete history show	Pass

5.2.2. Functional Testing

When unit testing is complete, the next step is to move on to functional testing. In the functional testing, we can check each module will work properly. We can also check that system will meet the requirement and specification.

Functional Testing 1: Insert image

Objective: To ensure that our image insert successfully, then system start working on it.

Table 11: Functional testing - Insert image.

No.	Test Case/Test Script	Attribute and Value	Expected Result	Actual Result
1.	Inset image into the system	Image	It will extract all the feature of the given image and then detect the forgery form the given image	Pass

Functional Testing 2: Check given image accuracy.

Objective: If process on the image complete show foraged areas as well as we can also check the image accuracy.

Table 12: Functional testing - Accuracy check

No.	Test Case/Test Script	Attribute and Value	Expected Result	Actual Result
1.	Inset image into the system	Image	It can compare the sample mask and the newly system generated mask to check the accuracy of the image. We can also check the complete image folder accuracy	Pass

Functional Testing 3: View history

Objective: To ensure that the upload image successfully in the database before start processing on it.

Table 13: Functional testing - Complete history

No.	Test Case/ Test Script	Attribute and Value	Expected Result	Actual Result
1.	User can view our history of image when click on history button they can also edit the image and delete the inserted image as well.	History	Preview detected images is shown in this portion	Pass

Functional Testing 3: Result

Objective: To ensure that you system shows the result.

Table 14: Functional testing - Output

No.	Test Case/ Test Script	Attribute and Value	Expected Result	Actual Result
1.	User can view the forgery result. directly	Output	Final output is shown	Pass

5.3. Project Progress Overview

Now we are going to show the actual implementation of our project with results. First, you can insert the image into our system, and then it will start processing on the given image; it will extract all possible features of the image before detecting the forgery from the image.

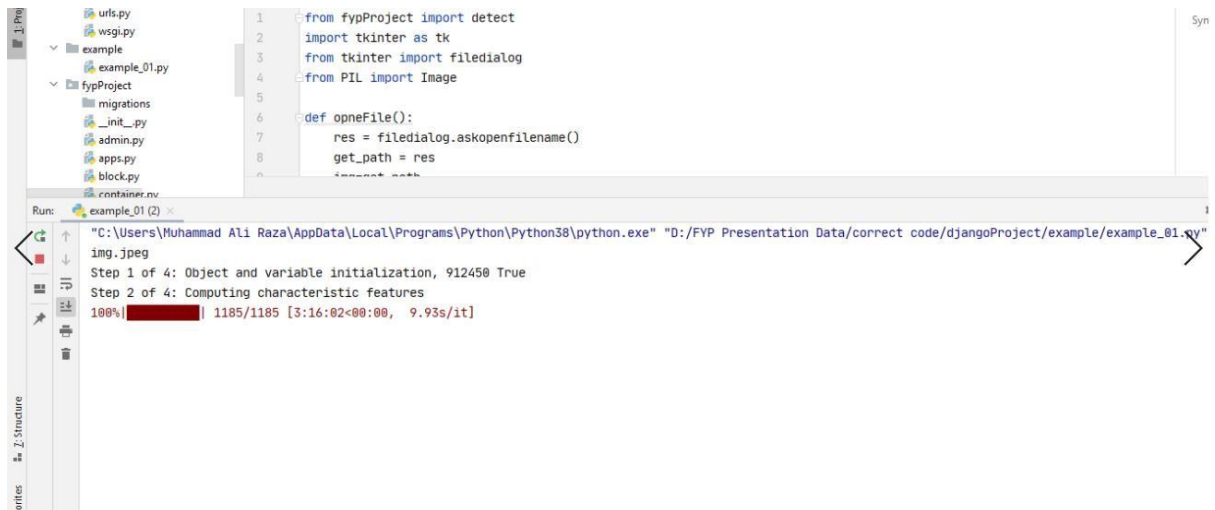


Figure 14: Feature Extraction

When the program completes our processing, it means complete the features extraction process, then it can detect where the forgery appears in the image, and then it will highlight those areas from the given image. But keep one thing in mind it can only detect the copy-move forgery. In the below image it will the mask of the image, it will show those areas in which forgery in exist.



Figure 15: Image binary mask

Finally, it will show the area of the image in which the forgery is exist, forged areas are highlight with the green rectangle.

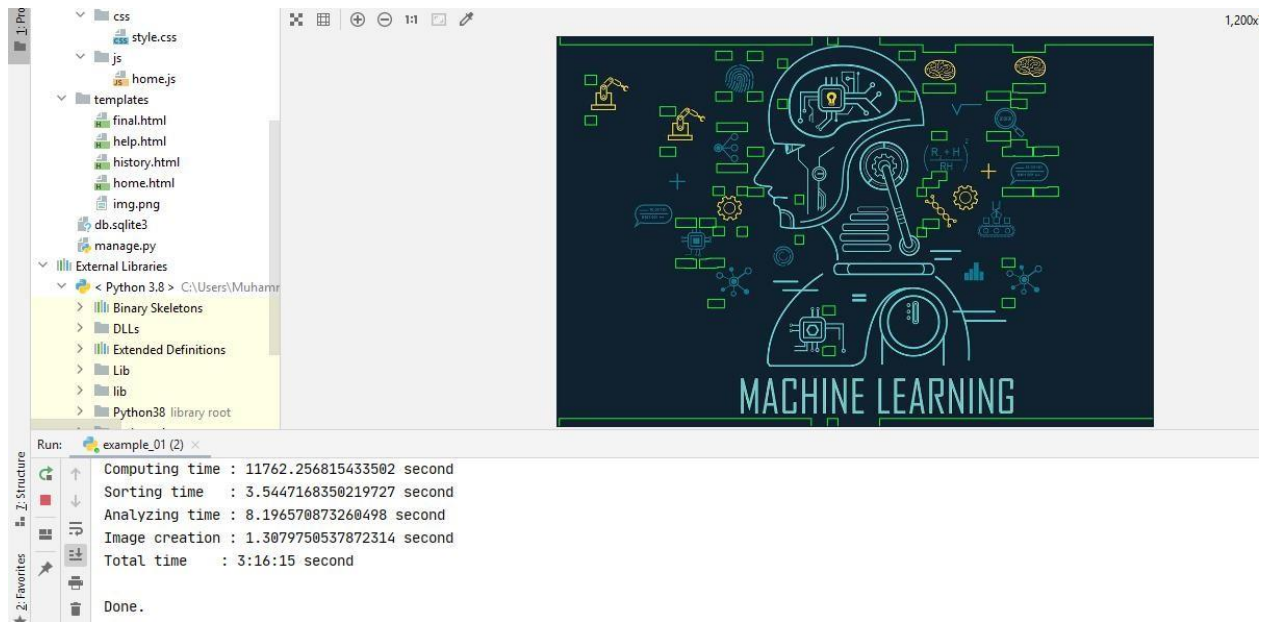


Figure 16: Forged image

CHAPTER NO. 07

Results

In this chapter we can discuss about the results of our system, we can categorize it into three different sub parts...

7.1. Image result

We can give the image to the system; system start processing on the image and fetch the forgery form the image. If system complete our processing, then it will show the any forgery exist or not in the image. We can see this in the result panel. Our system can also highlight the areas in which the forgery is exits.

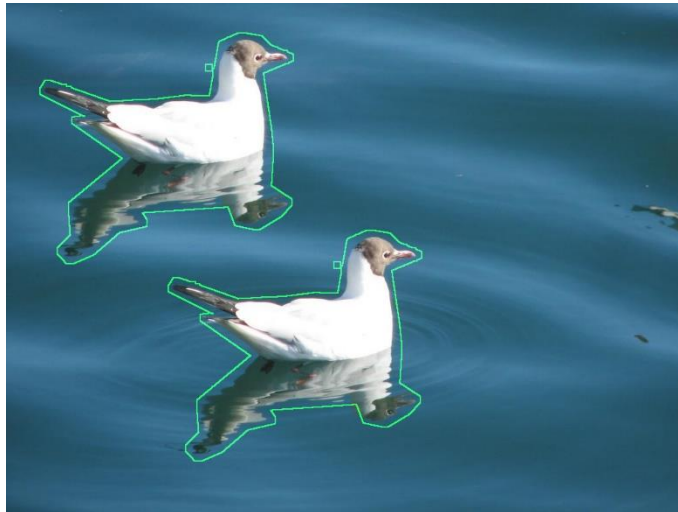


Figure 17: Result

7.2. Single image accuracy

We can also check the single image accuracy form our system, in single image accuracy we can give the two images to the system, one image is system generate result image and the second image is sample image. After insert both image our system will compare both images deeply and then show the accuracy result. It will check the system generate result is like the sample result; it will show the output after checking all these parameters.

Table 15: Single image accuracy checks

No. of images	Time	Accuracy
1	7-8 second	96.34749161142484
2	7-8 second	99.92177314211213
3	7-8 second	99.95188452285487
4	7-8 second	99.98197938441578

7.3. Complete image folder accuracy

We can also check the complete image folder accuracy at the same time. We can set both sample mask image folder and system generate mask image folder in the backend. User can simply add the image in both given folder and finally click on the result button our system compares each mask with the other mask and at the end when it will check all the image it will find the average of all of us and then show the accuracy on the screen.

Table 16: Complete image folder accuracy checks

Sample mask	System generated mask	Time	Accuracy
Sample mask folder	System generated mask folder	2 minutes (It will depend on the number of images in the folder)	99.31359444122413

CHAPTER NO. 09

CONCLUSION AND FUTURE WORK

6. Conclusion and Future Work

6.1. Conclusion

The proposed solution provides copy move forgery detection. Often image forgery detection software does not provide user friendly environment, but we also consider it because not all people are aware of new technology. And we also focus on history that a user can see his previous data. We try to make it simple and easy to understand. It based on the segmentation technique which makes it easy to detect and locate the forged area in image.

6.2. Future Work

Although, the proposed solution is very helpful for detection of faults in images. But this is supporting only PNG JPG, bmp image formats. In future, it may extend to other image formats like WEB, etc. Vector (SVG) images are more difficult to detect rather than raster images. Forgery in vector images may be detected by using another algorithm. Active forgery cannot be detected yet by any system in the world. Therefore, active forgery is very important to detect in future.

7. References

1. Al-Qershi, O. M., & KHoo, B. E. (2020). Passive detection of copy-move forgery in digital images. *Forensic Science International*, 284-295. Retrieved March 20, 2021, from www.elsevier.com/locate/forensic
2. Ansari, M. D., Gharera, S. P., & Tyagi, V. (2014, Aug 07). Pixel-Based Image Forgery Detection: A Review. *IETE Journal of Education*, 8. Retrieved Jan 29, 2021, from <https://doi.org/10.1080/09747338.2014.921415>
3. Ansari, M. D., Gharera, S. P., & Tyagi, V. (2018, August 7). Pixel-Based Image Forgery Detection: A Review. *IETE Journal of Education*, 37-41. Retrieved from <http://dx.doi.org/10.1080/09747338.2014.921415>
4. Birajdar, G. K., & Mankar, V. H. (2020). Digital image forgery detection using passive techniques. *Digital Investigation*, 1-20. Retrieved December 17, 2020, from www.elsevier.com/locate/di
5. Easow, S., & C., D. C. (2019). A Study on Image Forgery Detection Techniques. *International Journal of Computer (IJC)*, 8.
6. Farooq, S., Yousaf, M. H., & Hussain, F. (2020). A generic passive image forgery detection scheme using local binary pattern with rich models. *Computers and Electrical Engineering*, 1-14. Retrieved March 16, 2021, from www.elsevier.com/locate/compeleceng
7. Gahawar, G. K., Nath, V. V., & Gahawar, R. D. (2019, August 1). COMPREHENSIVE STUDY OF DIFFERENT TYPES IMAGE FORGERIES. *International Journal of Science Technology and Management*, 4, 6. Retrieved from www.ijstm.com
8. Kashyap, A., Parmar, R. S., Agarwal, M., & Gupta, H. (2017, November 15). An Evaluation of Digital Image Forgery Detection Approaches. *International Journal of Applied Engineering Research*, 12, 4747–4758. Retrieved Feb 22, 2021, from <http://www.ripublication.com>
9. Khan, S., Khan, K., Ali, F., & Kwak, K.-S. (2020, January 9). Forgery Detection and Localization of Modifications at the Pixel Level. *Symmetry*, 10. doi:10.3390/sym12010137
10. Mankar, S. K., & Gurjar, A. A. (2017, April). Image Forgery Types and Their Detection: A Review. *International Journal of Advanced Research in Computer Science and Software Engineering*, 5, 5. Retrieved 2021, from www.ijarcsse.com
11. Murali, S., Chittapur, G. B., S, P. H., & Anami, B. S. (2016). COMPARISON AND ANALYSIS OF PHOTO IMAGE FORGERY DETECTION TECHNIQUES. *International Journal on Computational Sciences & Applications (IJCSA)*, 12.
12. Nandi, G., & Sarma, B. (2016, November). A Study on Digital Image Forgery Detection. *International Journal of Advanced Research in Computer Science and Software Engineering*, 4(11), 6. Retrieved from www.ijarcsse.com

13. Rajini, N. H. (2019, june). Image Forgery Identification using Convolution Neural Network. *International Journal of Recent Technology and Engineering (IJRTE)*, 8(1S4), 10.
14. Xiao, B., Wei, Y., Bi, X., Li, W., & Ma, J. (2019, September 21). Image Splicing Forgery Detection Combining Coarse to Refined Convolutional Neural Network and Adaptive Clustering. *Information Science*, 39. doi:<https://doi.org/10.1016/j.ins.2019.09.038>