A

Mini Project

On

FAKE IMAGES DETECTION

(Submitted in partial fulfilment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

SAYAM AKSHAYA (207R1A05N9)

AMBATI NAGENDER (207R1A05J5)

PASHAM DEEPTHIKA (207R1A05N3)

Under the Guidance of

Najeema Afrin

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

2020-2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

This is to certify that the project entitled **"FAKE IMAGES DETECTION "** being submitted by **S.AKSHAYA(207R1A05N9), A.NAGENDER(207R1A05J5) & P. DEEPTHIKA(207R1A05N3)** in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2023-24.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Najeema Afrin
(Assistant Professor)
INTERNAL GUIDE

Dr. A. RajiReddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINAR

Submitted for viva voice Examination held on _____

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **Najeema Afrin** Assistant Professor for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by her shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **G. Vinesh Shanker, Dr. J. Narasimha Rao, Ms. Shilpa, & Dr. K. Maheswari** for their cordial support, valuable information, and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

SAYAM AKSHAYA (207R1A05N9)
AMBATI NAGENDER (207R1A05J5)
PASHAM DEEPTHIKA (207R1A05N3)

ABSTRACT

In this paper, we investigate whether robust hashing has a possibility to robustly detect fake-images even when multiple manipulation techniques such as JPEG compression are applied to images for the first time. In an experiment, the proposed fake image detection with robust hashing is demonstrated to outperform state-of-the-art one under the use of various datasets including fake images generated with GANs.

These days, the availability of image processing software, such as Adobe Photoshop or GIMP have made image manipulation so common. Detecting such fake images is unavoidable for unveiling of the image-based cybercrimes. An image taken by digital camera or smartphone is usually saved in the JPEG format due to its popularity. JPEG algorithm works on image grids, compressed independently, with a size of 8x8 pixels. While unmodified images, have a similar error level. For resaving operation, each block should degrade at around same rate due to similar amount of errors across the whole image. The compression ratio of this fake image is different from that of the original image and is detected using Error Level Analysis. The objective of our paper is to develop a photo forensics algorithm which can detect any photo manipulation. The error level analysis was then enhanced using vertical and horizontal histograms of error level analysis image to pinpoint the location of modification. Results show that the proposed algorithm could identify the modified image while showing the exact location of modification.

LIST OF FIGURES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1.INTRODUCTION

1. INTRODUCTI ON

1.1 PROJECT SCOPE

This project is titled “FAKE IMAGES DETECTION”. Models can be trained detect various types of image manipulations, such as cloning, splicing, retouching, and more. Detecting fake or manipulated images shared on social media platforms to prevent the spread of misinformation. Developing real-time systems that can quickly identify fake images as they are uploaded or shared online.

1.2 PROJECT PURPOSE

The purpose of a fake image detection project is to develop technology automatically identify manipulated or fraudulent images. This endeavor is crucial for combating the spread of misinformation, preserving trust in images used as evidence, protecting individuals' privacy, enhancing security through authentication, and maintaining the authenticity of visual content in various domains. By leveraging advanced techniques like computer vision and machine learning, these projects strive to ensure the accuracy and trustworthiness of images in an age where image manipulation and misinformation are increasingly prevalent.

1.2 PROJECT FEATURES

A fake image detection project using deep learning typically incorporates several key features to effectively identify manipulated or counterfeit images. Firstly, it relies on convolutional neural networks (CNNs) for image analysis, allowing the model to learn intricate visual patterns and irregularities. Dataset curation is crucial, including both authentic and manipulated images for training and validation. Preprocessing techniques such as image resizing and normalization are employed to ensure data consistency. To enhance accuracy, ensemble methods like combining multiple deep learning models can be used. Post-processing steps, such as thresholding and image forensics, may further refine results. Real-time or batch processing modes can be implemented depending on the project's requirements. with evolving manipulation techniques, making it a robust tool for detecting fake images.

2.SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

SYSTEM ANALYSIS

1. Stage 1: In this step, the features are found in a series of two or more images. If carried out perfectly, with no overhead cost; it may work efficiently with less overload and reduce the extraneous information to be processed.

2. Stage 2: Features found in stage 1 are matched between the frames. Under most common scenarios, two frames are used and two sets of features are matched to a resultant single set of motion vectors. These features in one frame are used as seed points which use other techniques to determine the flow.

   Despite this, both these stages of feature-based modelling possess drawbacks. In the stage of detecting features, it is necessary that features are located with precision and good reliability. This is proved to be of immense significance and research is performed on feature detectors. This feature holds an ambiguity of possible matches to occur as well; unless it is priorly known that image displacement less than the distance between features.

3. Frame Differencing and motion-based methods: Frame differencing is a method of finding the difference between two consecutive images from a sequence of images to segregate the moving object (vehicle) from the background. If there is a change in pixel values, it implies that there was a change in position in the two image frames. The motion rectification step of detecting a vehicle in a trail of images by alienating the moving objects, also known as blobs based on its speed, movement and orientation.

2.1 PROBLEM DEFINITION

The main challenge phase during Fake Images Detection is to detect the Fake Image. It identifies the fake images by using the Robust hashing techniques. It gives clarity about the image weather it is a fake or a Real Image.

2.2 EXISTING SYSTEM

Fake images are manually generated by using image editing tools such as Photoshop. Splicing, copy-move, and deletion are also carried out under the use of such a tool. Similarly, resizing, rotating, blurring, and changing the color of an image can be manually carried out. In addition, recent rapid advances in deep image synthesis techniques such as GANs have automatically generated fake images. Cycle GAN and Star GAN are typical image synthesis techniques with GANs. Cycle GAN is a GAN that performs one-to-one transformations, e.g. changing apples to oranges, while Star GAN is a GAN that performs many-to- many transformations, such as changing a person's facial expression or hair color (. Furthermore, fake videos created using deep learning are called Deepfake, and various tampering methods have emerged, such as those using autoencoders, Face2Face.

2.2.1 DISADVANTAGES OF EXISTING SYSTEM

- LESS ACCURACY
- LOW EFFICIENCY

2.3 PROPOSED SYSTEM

How an overview of the proposed method. In the framework , robust hash value is computed from easy reference image by using a robust hash method, and stored in a database. Similar to reference images, a robust hash value is computed from a query one by using the same hash method. The hash value of the query is compared with those stored the database. Finally, the query image is judged whether it is real or fake in accordance with the distance between two hash values

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

- HIGH ACCURACY
- HIGH EFFICIEN

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are,

- **ECONOMICAL FEASIBILITY**
- **TECHNICAL FEASIBILITY**
- **SOCIAL FEASIBILITY**

2.4.1 ECONOMIC FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

For developing the application the following are the Hardware Requirement

System	:	i3 or above.
Ram	:	4 GB.
Hard Disk	:	40 GB

2.5.2 SOFTWARE REQUIREMENTS:

Operating system	:	Windows8 or Above.
Coding Language	:	python 3.7

3.ARCHITECTURE

3.ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for classification, starting from input to final prediction.

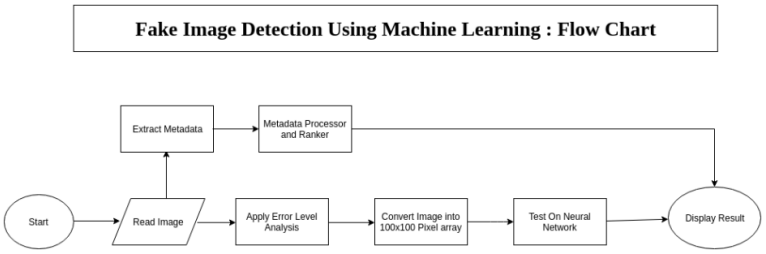


Figure 3.1: Project Architecture of Fake Images Recognition

3.2 DESCRIPTION

This project is totally based upon identifying the Fake Images . The model is built to recognize faces as the image is either real image or fake image by using the error level analysis and then Convert the image into pixel and then by testing it in the neural network to predict the Image.

3.3 USE CASE DIAGRAM

In the use case diagram, we have basically one actor who is the user in the trained model.

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.

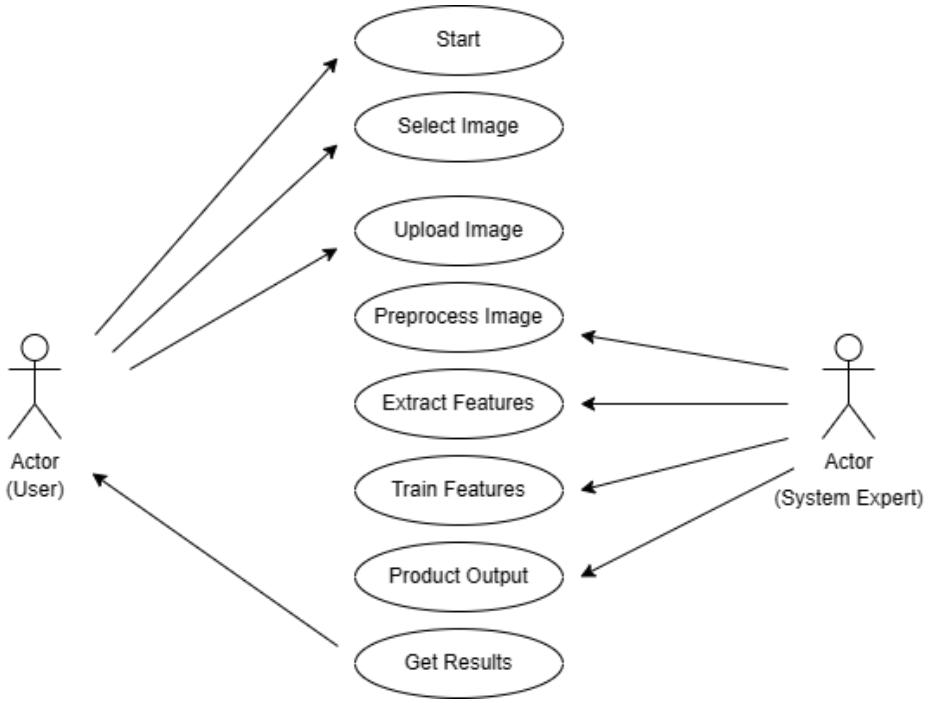


Figure 3.2: Use Case Diagram of Fake Images Detection

3.4 CLASS DIAGRAM

Class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations(or methods), and the relationships among objects.

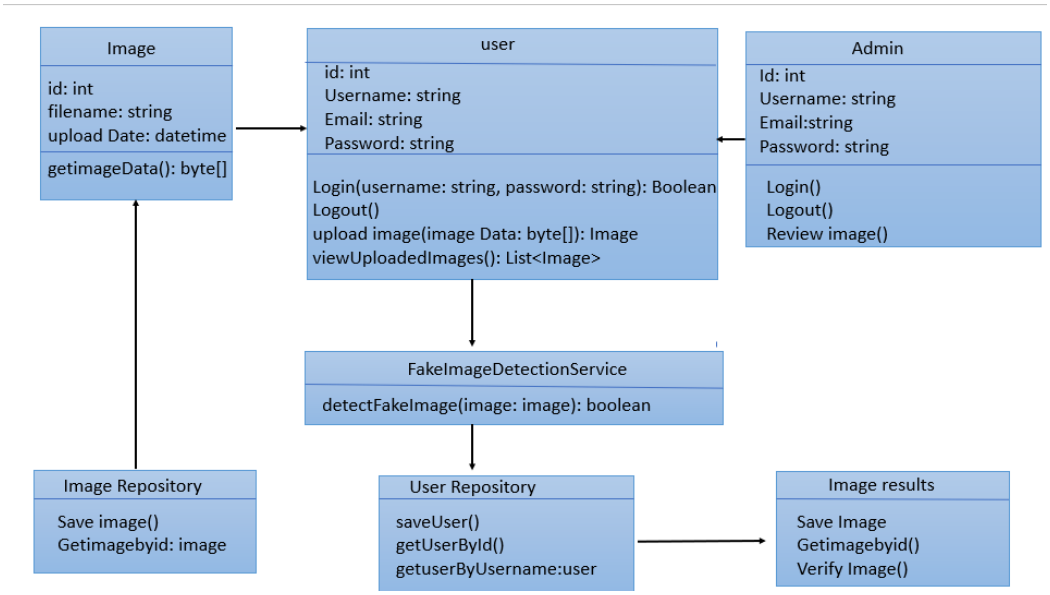


Figure 3.3: Class Diagram of Fake Image Detection

3.5 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development .

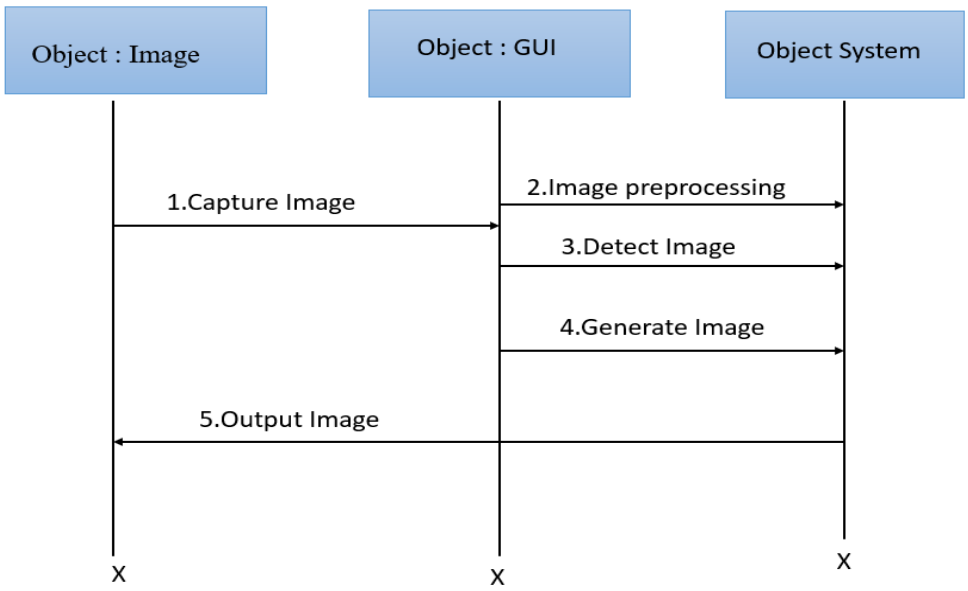


Figure 3.4: Sequence Diagram of Fake Images Detection

3.6 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. They can also include elements showing the flow of data between activities through one or more data stores.

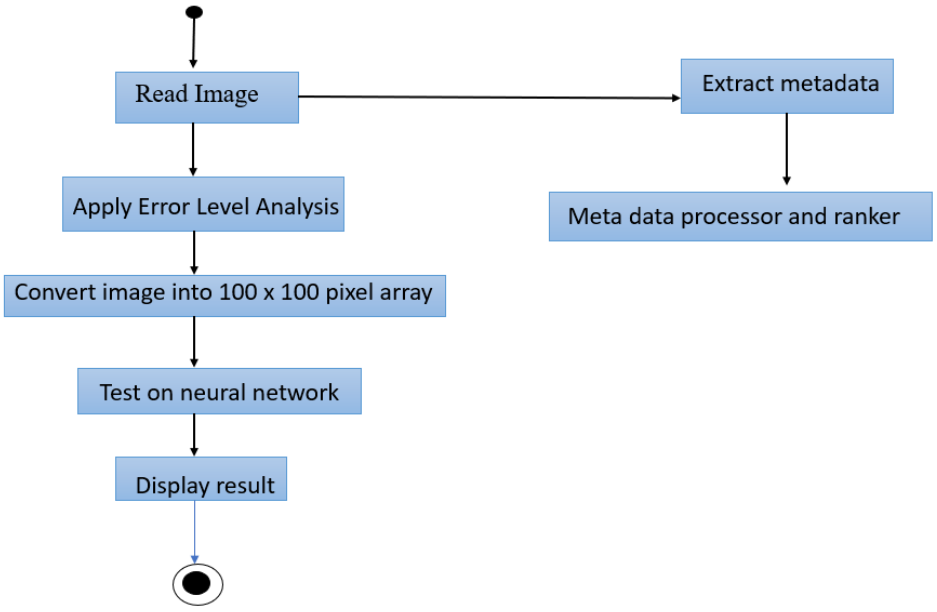


Figure 3.5: Activity Diagram of Fake Images Detection

4.IMPLEMENTATION

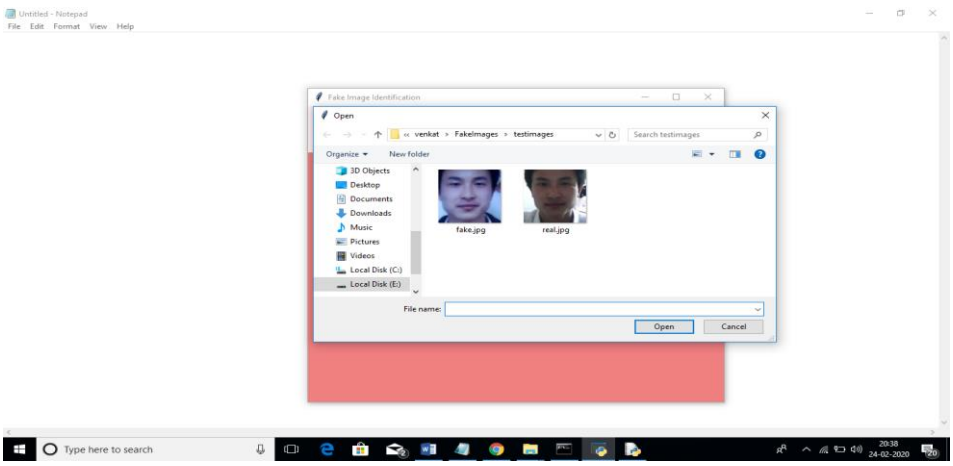
4.1 SAMPLE CODE

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
import os
    def get_pixel(img, center, x, y):
      new_value = 0
        try:
          if img[x][y] >= center:
                new_value = 1
         except:
            pass
        return new_value

def lbp_calculated_pixel(img, x, y):
center = img[x][y]
val_ar = []
val_ar.append(get_pixel(img, center, x-1, y+1))     # top_right

val_ar. append(get_pixel(img, center, x, y+1))       # right

 val_ar.append(get_pixel(img, center, x+1, y+1))     # bottom_right
val_ar.append(get_pixel(img, center, x+1, y))       # bottom
val_ar.append(get_pixel(img, center, x+1, y-1))     # bottom_left
val_ar.append(get_pixel(img, center, x, y-1))       # left
val_ar.append(get_pixel(img, center, x-1, y-1))     # top_left
val_ar.append(get_pixel(img, center, x-1, y))       # top
    power_val = [1, 2, 4, 8, 16, 32, 64, 128]
        val = 0
        for i in range(len(val_ar)):
              val += val_ar[i] * power_val[i]
        return val
def main():
    filename = 'data/Fake'
        for root, dirs, files in os.walk(filename):
          for fdata in files:
             image_file = root+"/"+fdata;
              img_bgr = cv2.imread(image_file)
              height, width, channel = img_bgr.shape
              img_gray = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2GRAY)
               img_lbp = np.zeros((height, width,3), np.uint8
```

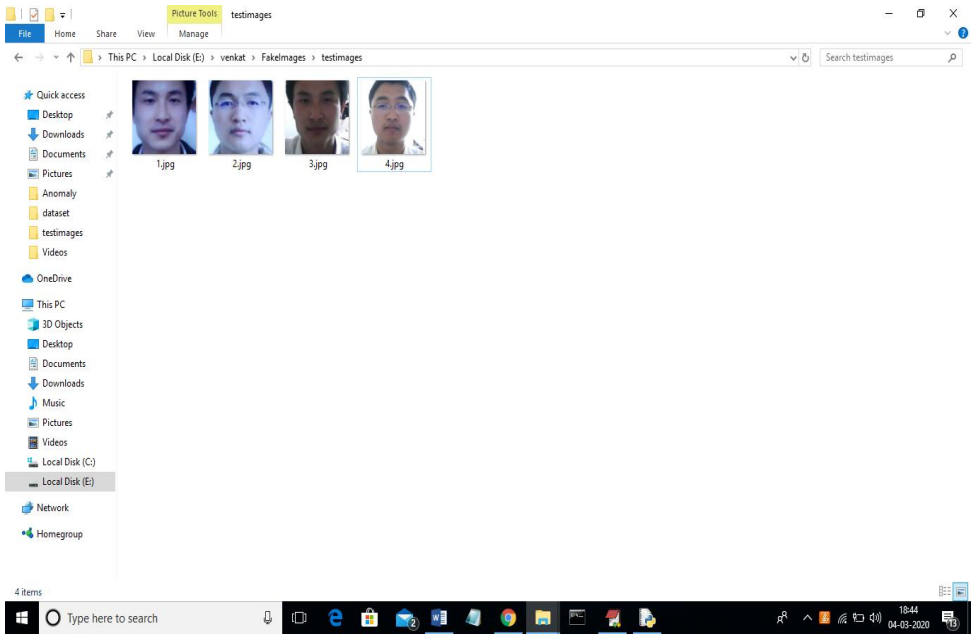
```
            for j in range(0, width):
                img_lbp[i, j] = lbp_calculated_pixel(img_gray, i, j)
        cv2.imwrite('LBP/validation/Fake/'+fdata, img_lbp)
        cv2.imwrite('LBP/train/Fake/'+fdata, img_lbp)

    filename = 'data/Real'
    for root, dirs, files in os.walk(filename):
        for fdata in files:
            image_file = root+"/"+fdata;
            img_bgr = cv2.imread(image_file)
            height, width, channel = img_bgr.shape
            img_gray = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2GRAY)
            img_lbp = np.zeros((height, width,3), np.uint8)
            for i in range(0, height):
                for j in range(0, width):
img_lbp[i, j] = lbp_calculated_pixel(img_gray, i, j)
        cv2.imwrite('LBP/validation/Real/'+fdata, img_lbp)
        cv2.imwrite('LBP/train/Real/'+fdata, img_lbp)
    print("LBP Program is finished")

if __name__ == '__main__':
    main()
```

5.RESULTS

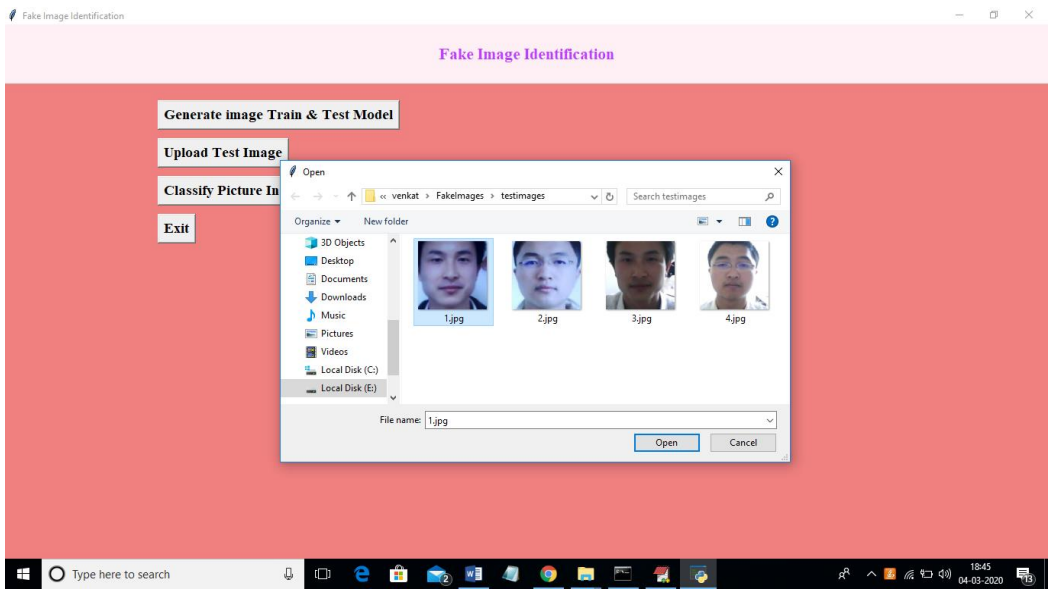
Screenshot 5.1 : Generate Image Train & Test Model



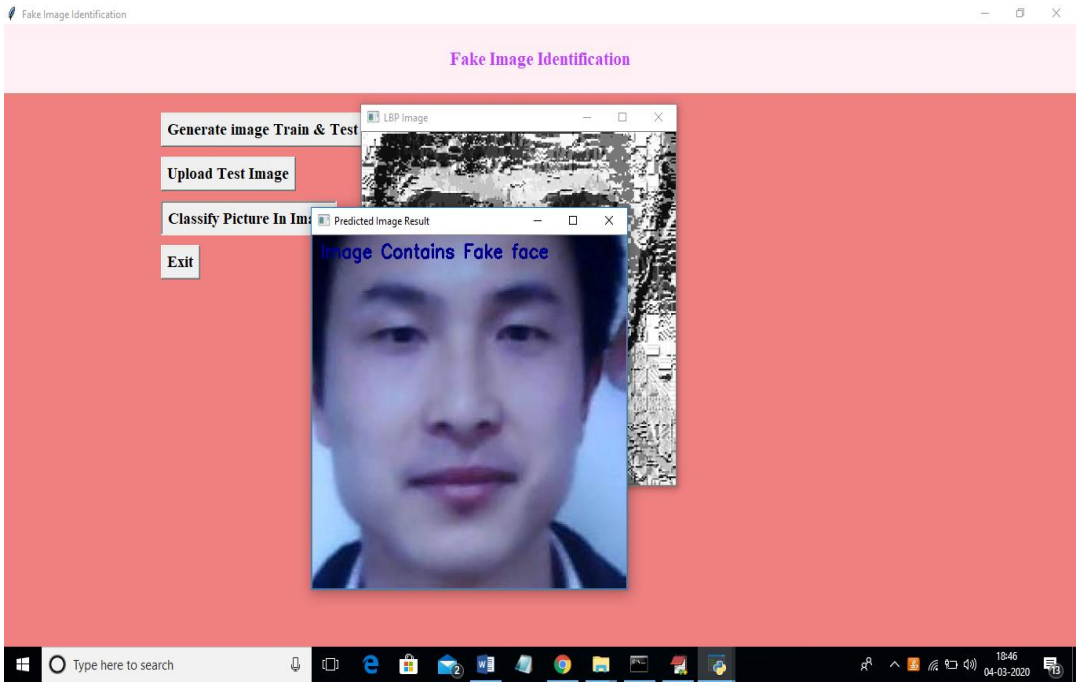
Screenshot 5.2 : Naming the images



Screenshot 5.3 : Detect whether face is real or fake



Screenshot 5.4: uploading jpg image



Screenshot 5.5: In above screen we are getting result as image contains Fake face

6.TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

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

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centred on the following items:

Valid Input	: identified classes of valid input must be accepted.
Invalid Input	: identified classes of invalid input must be rejected.
Functions	: identified functions must be exercised.
Output	: identified classes of application outputs must be exercised.
Systems/Procedures	: interfacing systems or procedures must be invoked

SYSTEM TEST

System testing ensures that the entire integrated software system requirements.It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. You cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format.
- No duplicate entries should be allowed.
- All links should take the user to the correct page.

6.3 Test Results:

All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements

7.CONCLUSION

7.CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

Fake image detection using deep learning has gained significant attention in recent years due to the increasing prevalence of manipulated images in social media and other online platforms. Another important aspect of fake image detection is the availability of large and diverse datasets. The availability of such datasets is essential for training deep learning models and improving their performance. However, creating such datasets is a challenging task, and more research is needed in this area.

In conclusion, deep learning techniques have shown great potential in detecting fake images, but more research is needed to address the challenges in this field, such as dataset availability and the detection of more sophisticated fake images.

7.2 FUTURE SCOPE

The future of fake image detection using deep learning is characterized by ongoing advancements in sophisticated models, multi-modal approaches combining text and image analysis, specialized detection techniques for deepfakes and GAN-generated content, real-time detection solutions, and increased focus on ethical considerations. This field's applications across diverse industries, potential regulatory developments, and interdisciplinary collaborations make it a promising area for research and innovation, as the need to combat fake content and misinformation remains a pressing global challenge.

8.BIBLIOGRAPHY

8.1 REFERENCES

1. Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. arXiv Preprint, arXiv:1710.10196 2017. 256

2. Brock, A.; Donahue, J.; Simonyan, K. Large scale gan training for high fidelity natural image synthesis. arXiv Preprint, arXiv:1809.11096 2018.

3. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent 259 adversarial networks. arXiv Preprint, 2017.

8.2 GITHUB LINK

https://github.com/Akshayasayam/Fake-Images-Detection