ENCRYPTED PHOTOGUARD

PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY

(COMPUTER SCIENCE & ENGINEERING)

To

Punjab Technical University, Jalandhar

SUBMITTED BY

PRIYAM SAHIL

1902186

VITH SEMESTER (2022)

UNDER THE GUIDANCE OF

DR. GAGANDEEP JINDAL SIR

PROFESSOR

Department of Computer Science & Engineering

Chandigarh Engineering College

Kharar-Banur Highway,Sector-112 Greater Mohali,

Punjab-140307(INDIA)

(Approved by AICTE, New Delhi and Affiliated to IKGPTU,Jalandhar)

CHANDIGARH ENGINEEERING COLLEGE, CGC-LANDRAN,MOHALI

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the project report entitled "ENCRYPTED PHOTOGUARD " by "PRIYAM SAHIL " in partial fulfillment of requirements for the award of degree of B.tech (CSE) submitted in the Department of CSE at Chandigarh Engineering College under the PUNJAB TECHNICAL UNIVERITY, JALANDHAR is an authentic record of my own work carried out during a period from January to April under the supervision of DR. GAGANDEEP JINDAL SIR.

PRIYAM SAHIL

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

The B.Tech Viva-Voice Examination of PRIYAM SAHIL has been on _____and accepted.

Name &Sign of Project Guide

Professor

ABSTRACT

Face detection is a computer technology that determines the location and size of human face in arbitrary image. The facial features are detected and any other objects like trees, buildings and bodies etc. are ignored from the digital image.

Majorly three different face detection algorithm are available based on RGB,YCbCr and HIS color space models. Three main steps viz.

- Classify the skin region in the color space.
- Apply threshold to mask the skin region
- Draw bounding box to extract the face image.

Python Imaging Library (abbreviated as PIL) (in newer versions known as Pillow) is a free and open-source additional library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats. It is available for Windows, Mac OS X and Linux

In the past decade, image encryption is given much attention in research of information security and a lot of image encryption algorithms have been introduced. Due to some intrinsic features of images like bulk data capacity and high data redundancy, the encryption of image is different from that of text; therefore it is difficult to handle them by traditional encryption methods. In the proposed work, a new image encryption algorithm based on Magic Rectangle (MR) is being applied. To begin with, the plain-image is converted into blocks of single bytes and then the block is replaced as the value of MR. Further, the control parameters of Magic Rectangle (MR) are selected randomly by the user. Subsequently the image is being encrypted with public key cryptography algorithms such as RSA, ElGamal etc. The experimental result shows that the proposed algorithm can successfully encrypt/decrypt the images with separate secret keys, and the algorithm has good encryption effect

ACKNOWLEDEMENT

I would like to please on record my deep sense of gratitude to DR. GAGANDEEP JINDAL SIR Dept. of Computer Science & Engineering, CEC-CGC, Landran for his generous guidance, help and useful suggestions.

I express my sincere gratitude to Dr. Sukhpreet Kaur, HOD in Department of CSE,CECCGC, Landran for her stimulating guidance, continuous encouragement.

I also wish to extends my thanks to friends, seniors and all the people who helped my for making the wonderful project and for their insightful comments and constructive suggestions to improve the quality of this research work.

PRIYAM SAHIL

1902186

VITH SEMESTER

CONTENTS **PAGE NO.**

1.INTRODUCTION

In recent years, along with the rapid promotion and popularization of network technology and digital communication technology in the world, digital images, and digital video-based digital images have become an important medium for information storage and transmission in the computer network in the civil and military fields. However, network security issues have long been an important factor that plagued and restricted the development of network technology. Especially in the context of the information resources of the public and government departments, how to realize the data security protection in the computer network is the important content and direction of the research in the field of network security and information security. Among them, digital image and digital video have become the important content of data transmission in the network by virtue of its intuitiveness and convenience. Therefore, the security protection of digital images has received great attention from all parties. Especially in the background of the increasingly severe network security situation in recent years, information transmission and sharing based on digital images often face the problems of data theft, tampering, deletion, and attack, which have caused great losses to the owners or publishers of digital images. So, the confidentiality and the integrity of data are needed to prevent eavesdroppers from accessing and using transmitted data. These photos not only contain our personal image, it also contains another detail like location, date of photo, technology details and much more. To preventing these data from hackers' photo-guards, get used to encrypt these critical data.

In the past decade, image encryption is given much attention in research of information security and a lot of image encryption algorithms have been introduced. Due to some intrinsic features of images like bulk data capacity and high data redundancy, the encryption of image is different from that of text; therefore it is difficult to handle them by traditional encryption methods. In the proposed work, a new image encryption algorithm based on Magic Rectangle (MR) is being applied. To begin with, the plain-image is converted into blocks of single bytes and then the block is replaced as the value of MR. Further, the control parameters of Magic Rectangle (MR) are selected randomly by the user. Subsequently the image is being encrypted with public key cryptography algorithms such as RSA, ElGamal etc. The experimental result shows that the proposed algorithm can successfully encrypt/decrypt the images with separate secret keys, and the algorithm has good encryption effect

In the field of secure encryption of digital images, two kinds of technical means are generally used:

1) Digital watermarking technology, namely Digital Watermarking Technology. The technology adopts the signature processing of digital images and adds custom watermark information to the original digital images to protect the copyright of digital images. It is one of the important technical means for image security protection in the Internet. However, the disadvantage of digital watermarking technology is that the visibility of digital images cannot

be avoided. Usually, only the copyright of the image is not infringed, and when the content of the digital image needs to be protected, there is nothing that can be done.

2) The digital image protection method is image encryption technology (encrypted photo-guard), and its basic principle is to encrypt the digital information contained in the digital image, and get the completely different encrypted images of the appearance and the original digital image, so that the content of the digital image cannot be viewed directly. When the digital image is needed for viewing or using, the corresponding decryption algorithm is used to calculate and decrypt the encrypted image to restore the original content of the digital image, which is an important means for digital image content protection in a distributed environment with high security requirements.

1.1 Objective OF Encrypted Photoguard.

Image encryption is necessary for future multimedia Internet applications. Password codes to identify individual users will likely be replaced with biometric images of fingerprints. However, such information will likely be sent over a network. When such images are sent over a network, an eavesdropper may duplicate or reroute the information. By encrypting these images, the content still has some degree of added security. Furthermore, by encrypting non-critical images as well, an eavesdropper is less likely to be able to distinguish between important and non-important information. Image encryption can also be used to protect privacy. An example for image encryption to protect privacy is in medical imaging applications. Recently, in order to reduce the cost and to improve the service, electronic forms of medical records have been sent over networks from the laboratories to medical centers or to doctors' offices. According to the law, medical records, which include many images, should not be disclosed to any unauthorized persons. Medical images, therefore, should be encrypted before they are sent over networks. Moreover, image encryption can be used to protect intellectual properties. One of concerns of the entertainment industry is that movies and videos in digital format are vulnerable to unauthorized access, theft, and replication. Entertainment industry has lost billion dollars due to the illegal copies. Recently, new technologies have been developed which allows multimedia can be delivered to millions of households very quickly. Entertainment industry will utilize Internet and satellites for multimedia distributions. The threat of unauthorized access during transmission over networks and the threat of illegal copy increase significantly. Image encryption, therefore, can be used to minimize these problems.

Although encryption is sufficient to protect digital images and videos in some civil applications, these issues have taken into consideration when advanced encryption algorithms are specially designed for sensitive digital images and videos, for their special features are very different from texts.

1.2 Problem Definition

The name of the project is ENCRYPTED PHOTOGUARD . In this project we emphasize on the face recognition and the security of the image.

The idea is to develop an application whose work is to secure the image. This application allows the user to click the image by the help of webcam, edit the image, encrypt it and decrypt also. By this the image is secured.

This project is based on python. There is general use of face recognition, encryption, decryption and some concepts of python.

ENCRYPTED PHOTOGUARD basically, provides to the user:

1. Face Recognition
2. Edit Image
3. Encryption
4. Decryption

1.3 Project Overview

The goal of this project is to secure the image for users to maintain their privacy. The user can choose among the following options to avail the services present in the respective option.

1. Face recognition: - In this step, firstly the program clicks a picture by the help of web camera and save it in our pc and image is added through face recognition and the user need not to go anywhere to have a photograph.

2 Edit image: - In this step, we make some editing in the image like contrast, brightness, sharpness etc. Then resave it as an edited image.

3 Encryption: - In this step, the editited picture is converted in to codes, by this no one will be able to open that image.

4 Decryption: - In this step, we decode the image by which the image is again converted from codes to image. So, then only we are able to look at that particular image

1.4 Hardware Specification

Strongly recommend a computer fewer than 5 years old.

- Processor – Core i3 or above
- Hard disk – minimum 512GB
- Memory – 1 GB RAM minimum
- Monitor
- Webcam

1.5 Software Specification

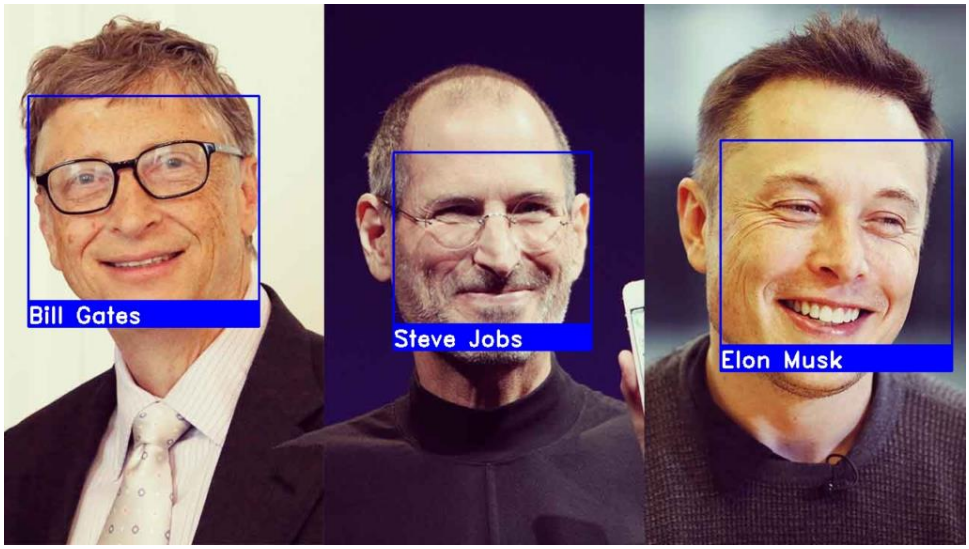
Development Tools and Information Programming Languages

- Python 3 Editor
- Python libraries
  - Numpy
  - Matplotlib
  - OpenCV
    0. cv
    1. cv2

2.LITERATURE SURVEY

2.1 Existing System

Face recognition, as one of the most successful applications of image analysis, has recently gained significant attention. It is due to availability of feasible technologies, including mobile solutions. Research in automatic face recognition has been conducted since the 1960s, but the problem is still largely unsolved. Last decade has provided significant progress in this area owing to advances in face modelling and analysis techniques. Although systems have been developed for face detection and tracking, reliable face recognition still offers a great challenge to computer vision and pattern recognition researchers. There are several reasons for recent increased interest in face recognition, including rising public concern for security, the need for identity verification in the digital world, face analysis and modelling techniques in multimedia data management and computer entertainment.



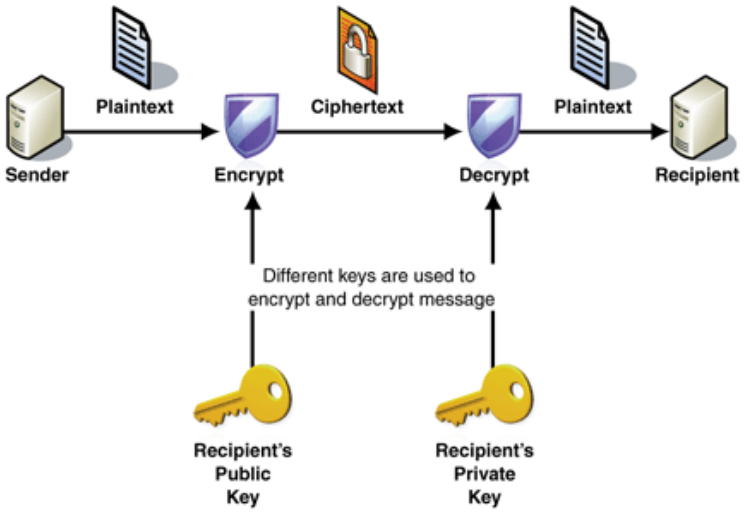
Recent advances in automated face analysis, pattern recognition and machine learning have made it possible to develop automatic face recognition systems to address these applications. On the one hand, recognizing face is natural process, because people usually do it effortlessly without much conscious. On the other hand, application of this process in area of computer vision remains a difficult problem. Being part of a biometric technology, automated face recognition has a plenty of desirable properties. They are based on the important advantage—non-invasiveness. The various biometric methods can be distinguished into physiological (fingerprint, DNA, face) and behavioral (keystroke, voice print) categories. The physiological approaches are more stable and non-alterable, except by severe injury. Behavioral patterns are more sensitive to human overall condition, such as stress, illness or fatigue.

Many applications have shown good results of the linear projection appearance-based methods such as principal component analysis (PCA) , independent component analysis (ICA) , linear discriminate analysis (LDA) , 2DPCA and linear regression classifier (LRC).

However, due to large variations in illumination conditions, facial expression and other factors, these methods may fail to adequately represent the faces. The main reason is that the face patterns lie on a complex nonlinear and non-convex manifold in the high-dimensional space.

ENCRYPTION

One of the earliest forms of encryption is symbol replacement, which was first found in the tomb of Khnumhotep II, who lived in 1900 B.C. Egypt. Symbol replacement encryption is "non-standard," which means that the symbols require a cipher or key to understand. This type of early encryption was used throughout Ancient Greece and Rome for military purposes. One of the most famous military encryption developments was the Caesar Cipher, which was a system in which a letter in normal text is shifted down a fixed number of positions down the alphabet to get the encoded letter. A message encoded with this type of encryption could be decoded with the fixed number on the Caesar Cipher.



Around 1790, Thomas Jefferson theorized a cipher to encode and decode messages in order to provide a more secure way of military correspondence. The cipher, known today as the Wheel Cipher or the Jefferson Disk, although never actually built, was theorized as a spool that could jumble an English message up to 36 characters. The message could be decrypted by plugging in the jumbled message to a receiver with an identical cipher.

A similar device to the Jefferson Disk, the M-94, was developed in 1917 independently by US Army Major Joseph Mauborne. This device was used in U.S. military communications until 1942.

4 of the most common encryption methods

Different encryption methods are based on the type of keys used, key length, and size of data blocks encrypted. Here are some of the common encryption methods that you might see used in various encryption tools:

1. Advanced Encryption Standard (AES)

Advanced Encryption Standard is a symmetric encryption algorithm that encrypts fixed blocks of data (of 128 bits) at a time. The keys used to decipher the text can be 128-, 192-, or 256-bit long. The 256-bit key encrypts the data in 14 rounds, the 192-bit key in 12 rounds, and the 128-bit key in 10 rounds. Each round consists of several steps of substitution, transposition, mixing of plaintext, and more. AES encryption standards are the most commonly used encryption methods today, both for data at rest and data in transit.

2. Rivest-Shamir-Adleman (RSA)

Rivest-Shamir-Adleman is an asymmetric encryption algorithm that is based on the factorization of the product of two large prime numbers. Only someone with the knowledge of these numbers will be able to decode the message successfully. RSA is often used when transmitting data between two separate endpoints (e.g., web connections), but works slowly when large volumes of data need to be encrypted.

3. Triple DES (Data Encryption Standard)

Triple DES is a symmetric encryption and an advanced form of the DES method that encrypts blocks of data using a 56-bit key. Triple DES applies the DES cipher algorithm three times to each data block. Triple DES is commonly used to encrypt ATM PINs and UNIX passwords.

4. Twofish

Twofish is a license-free encryption method that ciphers data blocks of 128 bits. It's considered the successor to the 64-bit Blowfish encryption method and more versatile than its specialized successor, Threefish. Twofish always encrypts data in 16 rounds regardless of the key size. Though it works slower than AES, the Twofish encryption method continues to be used by some file and folder encryption software solutions.

DECRYPTION

In one of the earliest and simplest ciphers, Julius Caesar sent messages in which each letter was substituted by the letter three places after it in the alphabet. In place of A, then, one would use a D. The key for such a cipher would be simply, "Shift right by three," or something similar.

A key is an algorithm, or a method for solving a mathematical problem by using a finite number of computations, usually involving repetition of certain operations or steps. An excellent example of an algorithm is f(x) = y, a formula by which a relationship between two elements is shown on a Cartesian coordinate system. It is said that “y ” is a function of x, meaning that for every value of x, there is a corresponding value of y. Suppose it is established that 2 x = y; then the key for the function has been established, and all possible values of x and y can be mapped.

In a simplified form, this is what occurs in decryption

2.2 Proposed System

With the further developments and deepening into concepts the project could be modified and be capable of detection of moving faces i.e detection of human faces in animated images or detection of non human objects using own developed built cascades and using pillow module we can edit the image like we can change the theme ,use the border in the picture ,change the contrast etc. and encrypt it for the protection of the image and decrypt it who has the excess for that image by using cryptography. The whole project is based on python.

2.3 Feasibility Study

Face recognition are geared towards simplifying difficult face recognition problems in uncontrolled environments. Such systems are able to control illumination, offer neutral pose and improve the poor performance of many face recognition algorithms. Door access control systems control illumination and pose in order to overcome face recognition problems. While there have been significant improvements in the algorithms with increasing recognition accuracy, very little research has been conducted on implementing these in hardware devices. Most of the previous studies focused on implementing a simple principal component analysis in hardware with low recognition accuracy. In contrast, the use of a Gabor filter for feature extraction and the nearest neighbour method for classification were found to be better alternatives. Dramatic developments in field programmable gate arrays (FPGAs) have allowed designers to select various resources and functions to implement many complex designs. The aim of this paper is to present the feasibility of implementing Gabor filter and nearest neighbour face recognition algorithms in an FPGA device for face recognition.Pillow module helps to edit image in this project. In the proposed work, a new image encryption algorithm based on Magic Rectangle (MR) is being applied. To begin with, the plain-image is converted into blocks of single bytes and then the block is replaced as the value of MR. Further, the control parameters of Magic Rectangle (MR) are selected randomly by the user. Subsequently the image is being

encrypted with public key cryptography algorithms such as RSA, ElGamal etc. The experimental result shows that the proposed algorithm can successfully encrypt/decrypt the images with separate secret keys, and the algorithm has good encryption effect

3.SYSTEM ANALYSIS AND DESIGN

3.1 Requirement Specification

Face detection

Face detection is a computer vision technology that helps to locate/visualize human faces in digital images. This technique is a specific use case of object detection technology that deals with detecting instances of semantic objects of a certain class (such as humans, buildings or cars) in digital images and videos. With the advent of technology, face detection has gained a lot of importance especially in fields like photography, security, and marketing.

Numpy

Numpy is an open-source library for working efficiently with arrays. Developed in 2005 by Travis Oliphant, the name stands for Numerical Python. As a critical data science library in Python, many other libraries depend on it.

Mathematical operations on NumPy's ndarray objects are up to 50x faster than iterating over native Python lists using loops. The efficiency gains are primarily due to NumPy storing array elements in an ordered single location within memory, eliminating redundancies by having all elements be the same type and making full use of modern CPUs. The efficiency advantages become particularly apparent when operating on arrays with thousands or millions of elements, which are pretty standard within data science.

It offers an Indexing syntax for easily accessing portions of data within an array.

It contains built-in functions that improve quality of life when working with arrays and math, such as functions for linear algebra, array transformations, and matrix math.

It requires fewer lines of code for most mathematical operations than native Python lists.

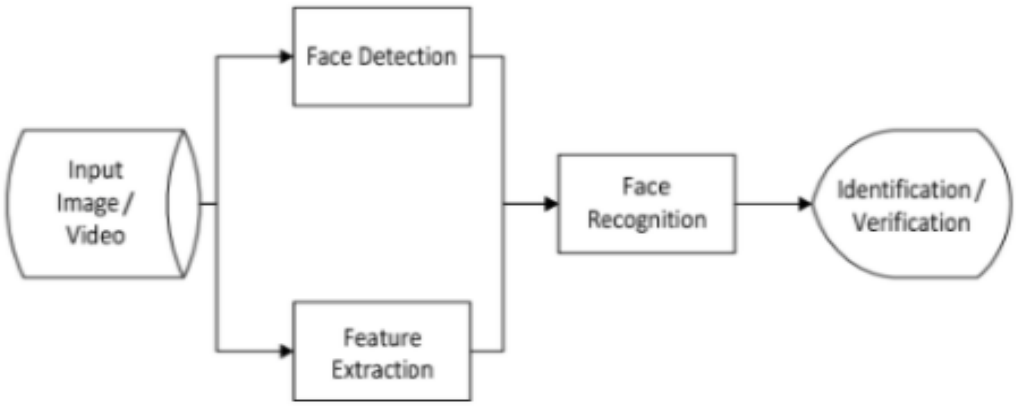
Matplotlib

Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open-source alternative to MATLAB. Developers can also use matplotlib's APIs (Application Programming Interfaces) to embed plots in GUI applications.

OpenCV

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as **NumPy**, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features.

Face Recognition Process:



Pillow Module

This is the home of Pillow, the friendly PIL fork. PIL is the Python Imaging Library.

The Python Imaging Library supports a wide variety of image file formats. To read files from disk, use the open() function in the Image module. You don't have to know the file format to open a file. The library automatically determines the format based on the contents of the file.

To save a file, use the save() method of the Image class. When saving files, the name becomes important. Unless you specify the format, the library uses the filename extension to discover which file storage format to use.

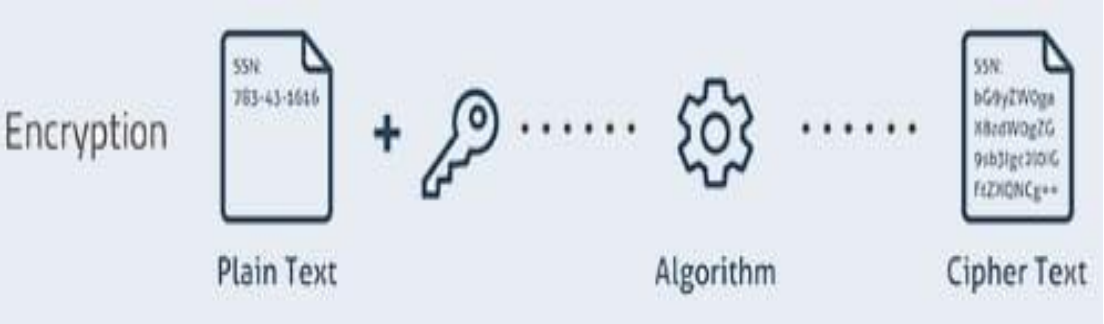
This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities.

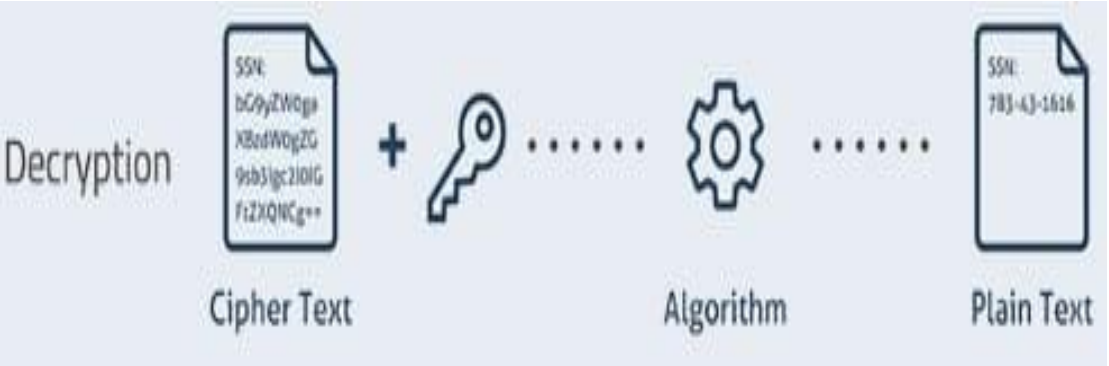
The core image library is designed for fast access to data stored in a few basic pixel formats. It should provide a solid foundation for a general image processing tool.

1.Image Archives 2.Image Display 3.Image Processing

Encryption: In cryptography, encryption is the process of encoding information. This process converts the original representation of the information, known as plaintext, into an alternative form known as ciphertext.

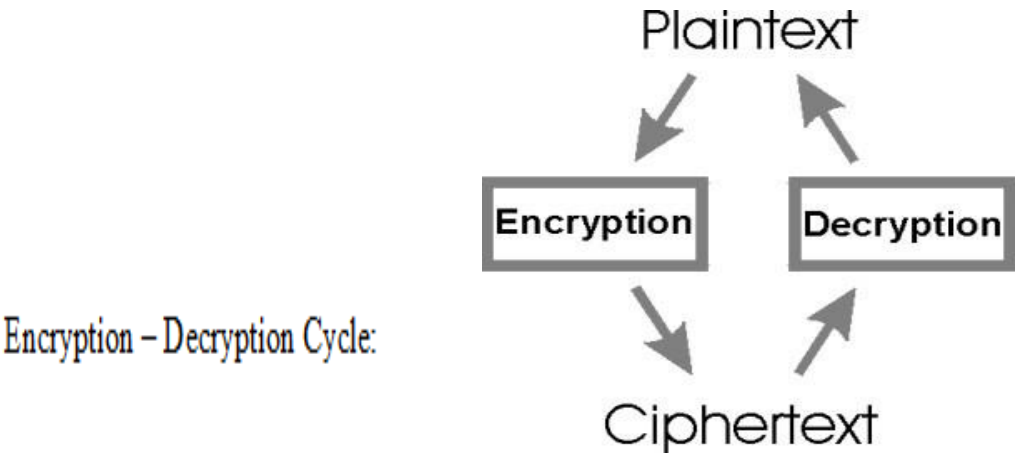
Encryption does not itself prevent interference but denies the intelligible content to a would-be interceptor. For technical reasons, an encryption scheme usually uses a pseudo-random encryption key generated by an algorithm. It is possible to decrypt the message without possessing the key, but, for a well-designed encryption scheme, considerable computational resources and skills are required.





Decryption: It is the process of transforming data that has been rendered unreadable through encryption back to its unencrypted form. In decryption, the system extracts and converts the garbled data and transforms it to texts and images that are easily understandable not only by the reader but also by the system. Decryption may be accomplished manually or automatically. It may also be performed with a set of keys or passwords.

It is the process of decoding encrypted information so that is can be accessed again by authorized users.



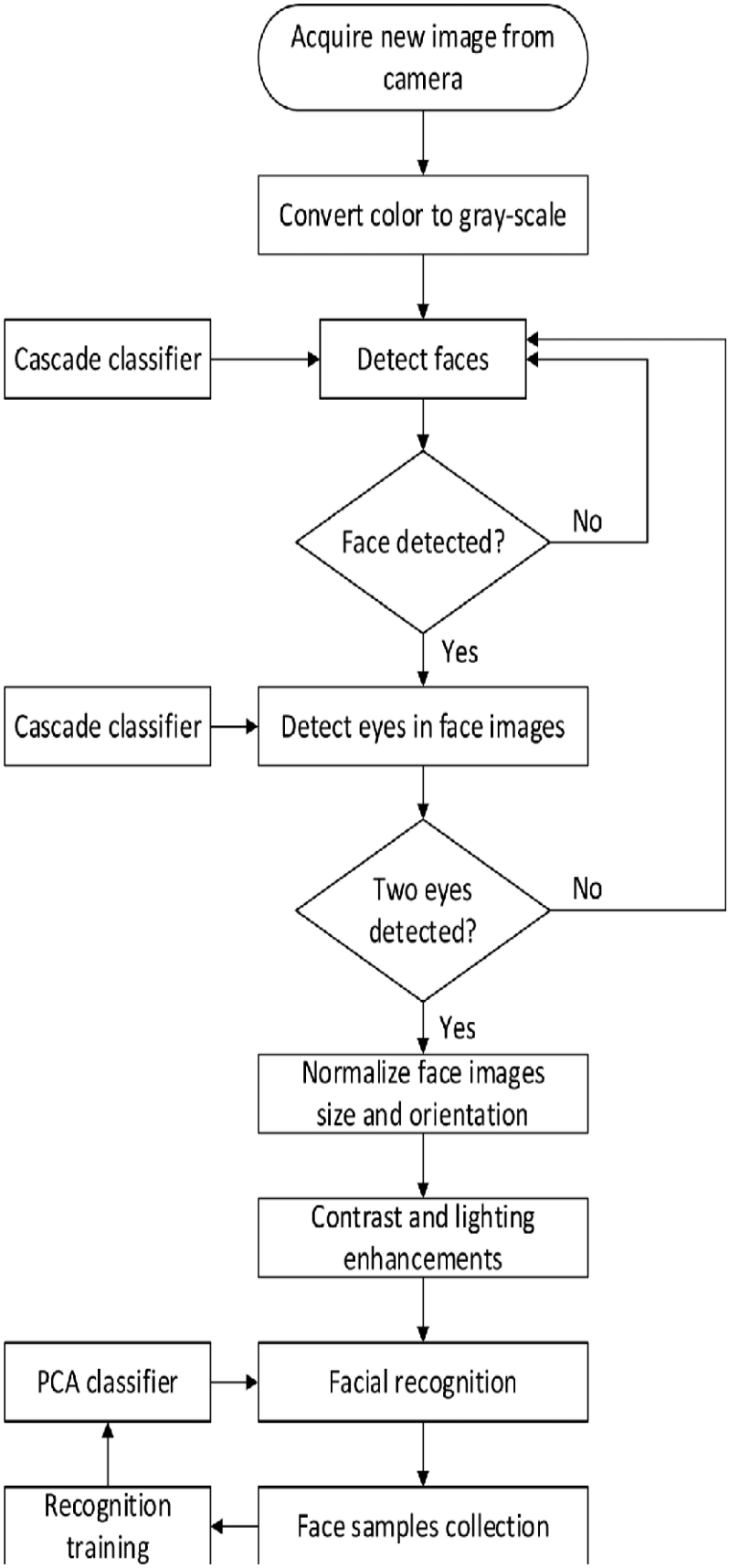
Encryption – Decryption Cycle:

To make the data confidential, data (plain text) is encrypted using a particular algorithm and a secret key. After encryption process, plain text gets converted into cipher text. To decrypt the cipher text, similar algorithm is used and at the end the original data is obtained again.

3.2 Flowcharts/DFDs/ERDs

Flowchart depicting the algorithm of the process of face detection is presented on the next page. It depicts how the compiler works to recognize the face and it is convenient for layman to understand the working behind this procedure.

FIGURE 1 ON NEXT PAGE:


Acquire new image from camera
Convert color to gray-scale
Cascade classifier
Detect faces
Face detected?
No
Yes
Cascade classifier
Detect eyes in face images
Two eyes detected?
No
Yes
Normalize face images size and orientation
Contrast and lighting enhancements
PCA classifier
Facial recognition
Recognition training
Face samples collection

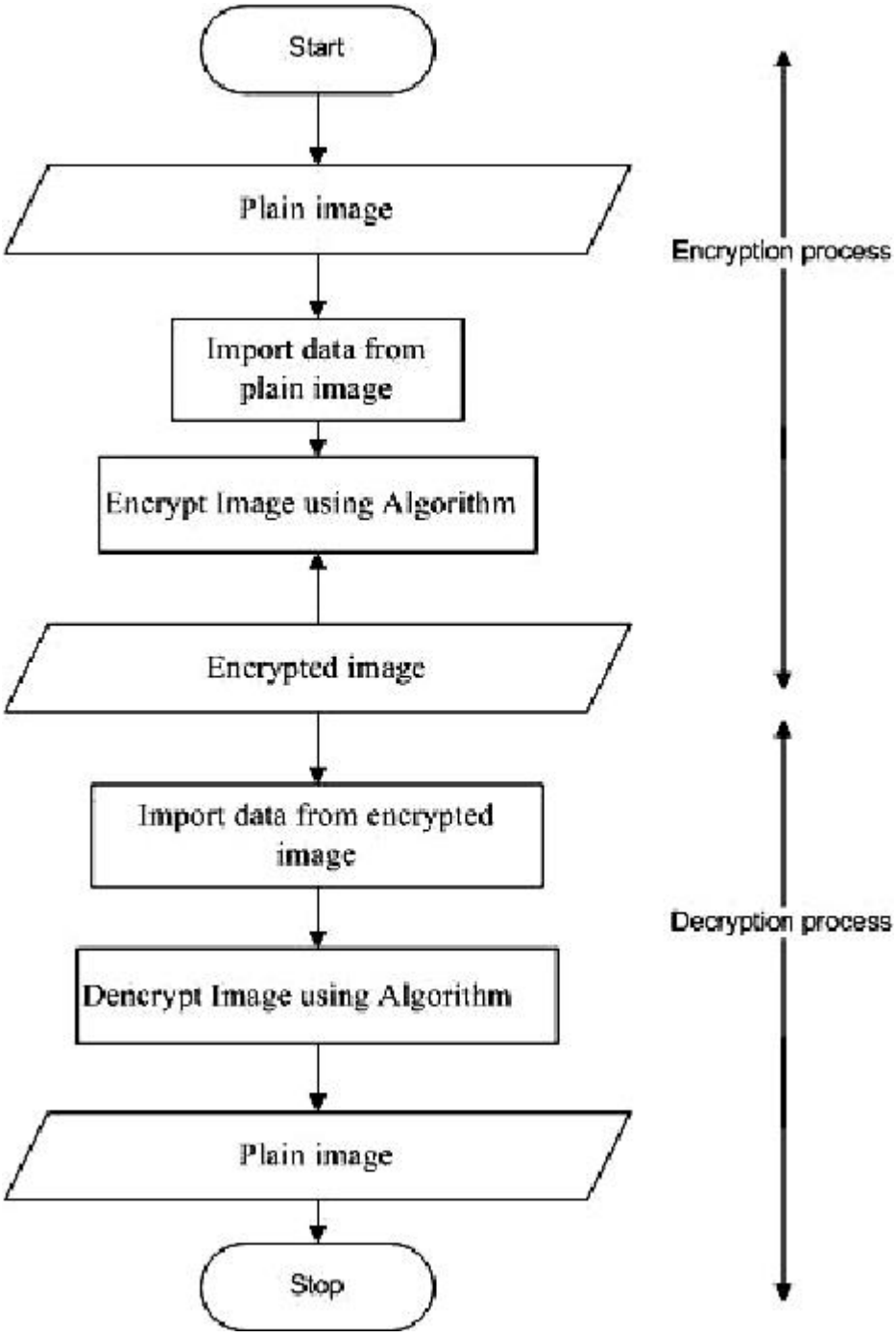

Start
Plain image
Import data from plain image
Encrypt Image using Algorithm
Encrypted image
Import data from encrypted image
Dencrypt Image using Algorithm
Plain image
Stop
Encryption process
Decryption process

FIGURE 2

3.3 Design and Test Steps

The facial recognition process normally has four interrelated phases or steps. The first step is face detection, the second is normalization, the third is feature extraction, and the final step is face recognition. These steps are separate components of a facial recognition system and depend on each other

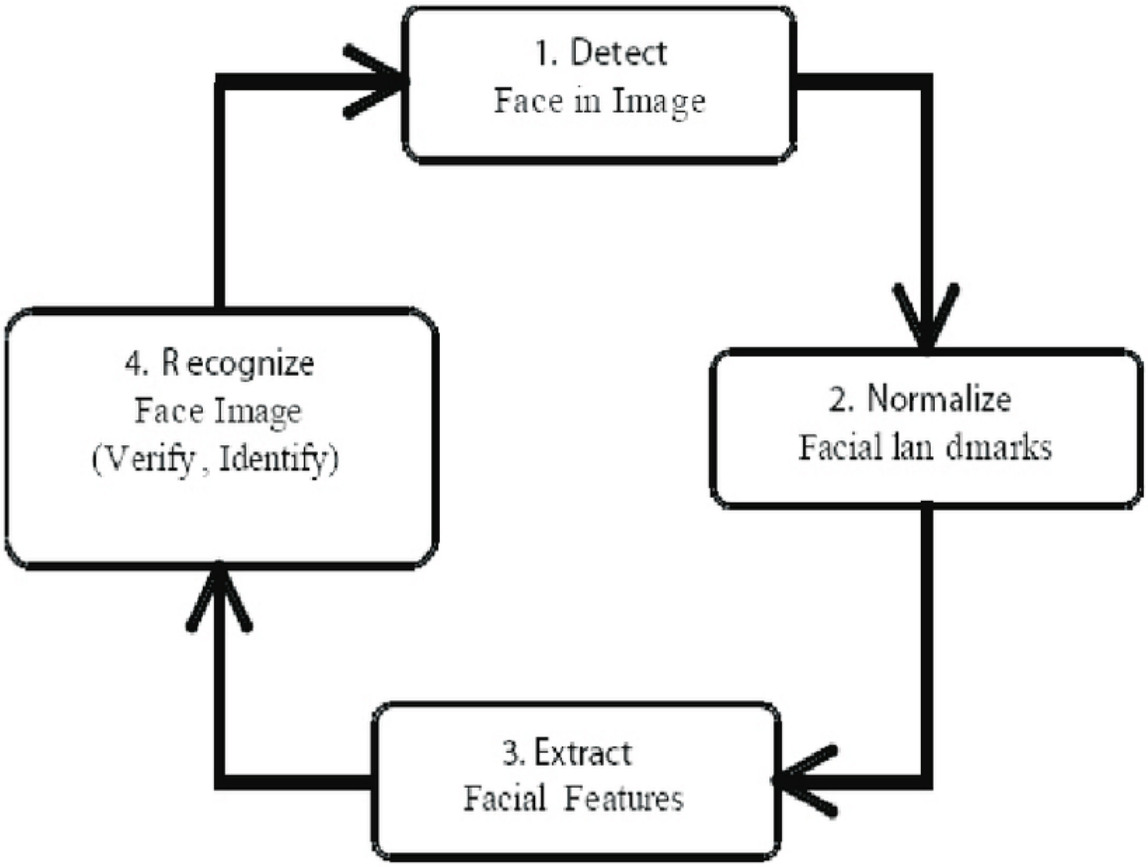
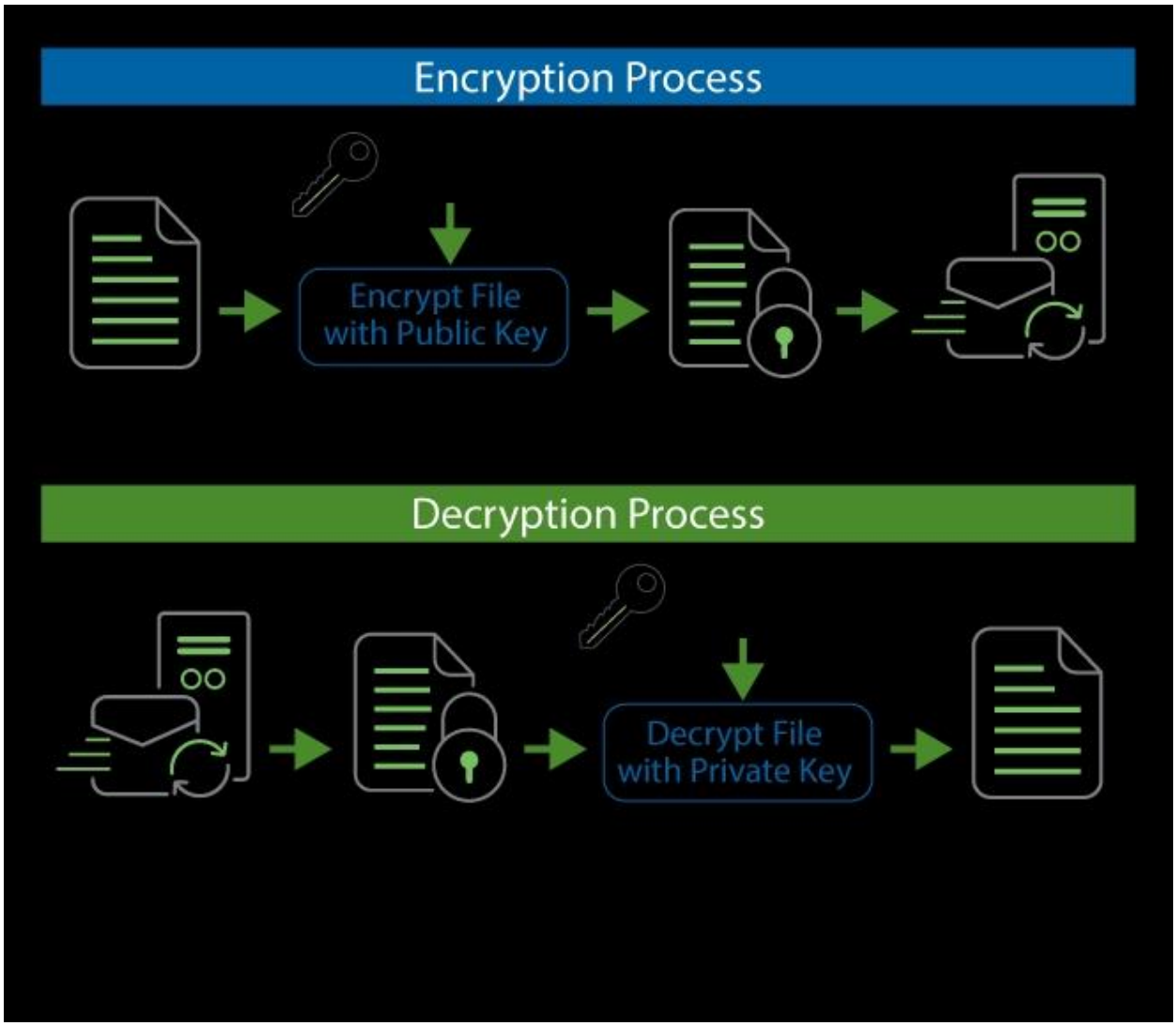


FIGURE 3 present the relationship diagram between the phases

Then using pillow module we can edit and save the image in pc.

Encryption and Decryption :-

Encryption is the process of converting normal message (plaintext) into meaningless message (Ciphertext). Whereas Decryption is the process of converting meaningless message (Ciphertext) into its original form (Plaintext)


Encryption Process
Encrypt File
with Public Key
Decryption Process
Decrypt File
with Private Key

3.4 Algorithm and Pseudo Code

ALGORITHM AND PSEUDOCODE OF OPEN CV:

1.) READING, WRITING AND DISPLAYING IMAGES :

ALGORITHM:

1.#import the libraries

2.import numpy as np

3.import matplotlib.pyplot as plt

4.import cv2

5.%matplotlib inline

6.#reading the image

7.image = cv2.imread('index.png')

8.image = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)

9.#plotting the image

10.plt.imshow(image)

11.#saving image

12.cv2.imwrite('test_write.jpg',image)

By default, the imread function reads images in the BGR (Blue-Green-Red) format. We can read images in different formats using extra flags in the imread function:

cv2.IMREAD_COLOR: Default flag for loading a color image

cv2.IMREAD_GRAYSCALE: Loads images in grayscale format

cv2.IMREAD_UNCHANGED: Loads images in their given format, including the alpha channel. Alpha channel stores the transparency information – the higher the value of alpha channel, the more opaque is the pixel

2.) CHANGING COLOR SPACES :

Algorithm :

1.#import the required libraries

2.import numpy as np

3.import matplotlib.pyplot as plt

4.import cv2

5.%matplotlib inline

6.image = cv2.imread('index.jpg')

7.#converting image to Gray scale

8.gray_image = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)

9.#plotting the grayscale image

10.plt.imshow(gray_image)

11.#converting image to HSV format

12.hsv_image = .cv2.cvtColor(image,cv2.COLOR_BGR2HSV)

13.#plotting the HSV image

14.plt.imshow(hsv_image)

3.) RESIZING IMAGES :

Algorithm:

1.import cv2

2.import numpy as np

3.import matplotlib.pyplot as plt

4.%matplotlib inline

5.#reading the image

6.image = cv2.imread('index.jpg')

7.#converting image to size (100,100,3)

8.smaller_image = cv2.resize(image,(100,100),inerpolation='linear')

9.#plot the resized image

10.plt.imshow(smaller_image)

4.) IMAGE ROTATION

Algorithm:

1.importing the required libraries

2.import numpy as np

3.import cv2

4.import matplotlib.pyplot as plt

5.%matplotlib inline

6.image = cv2.imread('index.png')

7.rows,cols = image.shape[:2]

8.#(col/2,rows/2) is the center of rotation for the image

9.# M is the cordinates of the center

10.M = cv2.getRotationMatrix2D((cols/2,rows/2),90,1)

11.dst = cv2.warpAffine(image,M,(cols,rows))

12.plt.imshow(dst)

13.for m,n in matches:

    if m.distance < 0.75*n.distance:

        good.append([m])

14.image3 = cv2.drawMatchesKnn(image1,kp1,image2,kp2,good,flags = 2)

5.) FACE DETECTION:

OpenCV supports haar cascade based object detection. Haar cascades are machine learning based classifiers that calculate different features like edges, lines, etc in the image. Then, these classifiers train using multiple positive and negative samples.

Algorithm :

```
1.import required libraries
2.import numpy as np
3.import cv2 as cv
4.import matplotlib.pyplot as plt
5.%matplotlib inline
6.#load the classifiers downloaded
7.face_cascade = cv.CascadeClassifier('haarcascade_frontalface_default.xml')
8.eye_cascade = cv.CascadeClassifier('haarcascade_eye.xml')
9.#read the image and convert to grayscale format
10.img = cv.imread('rotated_face.jpg')
11.gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
12.#calculate coordinates
faces = face_cascade.detectMultiScale(gray, 1.1, 4)
13.for (x,y,w,h) in faces:
14.cv.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
15.  roi_gray = gray[y:y+h, x:x+w]
16.  roi_color = img[y:y+h, x:x+w]
17.   eyes = eye_cascade.detectMultiScale(roi_gray)
18.  #draw bounding boxes around detected features
19.   for (ex,ey,ew,eh) in eyes:
      cv.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)
20.#plot the image
21.plt.imshow(img)
22.#write image
23.cv2.imwrite('face_detection.jpg',img)
24.dist_transform = cv2.distanceTransform(opening,cv2.DIST_L2,5)
ret,sure_fg = cv2.threshold(dist_transform,0.7*dist_transform.max(),255,0)
25.sure_fg = np.uint8(sure_fg)
unknown = cv2.subtract(sure_bg,sure_bg)
```

26.ret, markers = cv2.connectedComponents(sure_fg)

markers = markers+1

markers[unknown==255] = 0

27.markers = cv2.watershed(image,markers)

image[markers==-1] = [255,0,0]

28.plt.imshow(sure_fg)

6.) EDGE DETECTION :

Algorithm :

1.import the required libraries

2.import numpy as np

3.import cv2

4.import matplotlib.pyplot as plt

5.%matplotlib inline

6.#read the image

7.image = cv2.imread('coins.jpg')

8.#calculate the edges using Canny edge 9.algorithm

edges = cv2.Canny(image,100,200)

10.#plot the edges

11.plt.imshow(edges)

INTER_NEAREST: Nearest neighbor interpolation

INTER_LINEAR: Bilinear interpolation

INTER_AREA: Resampling using pixel area relation

INTER_CUBIC: Bicubic interpolation over 4×4 pixel neighborhood

INTER_LANCZOS4: Lanczos interpolation over 8×8 neighborhood

OpenCV’s resize function uses bilinear interpolation by default.

)

)

7.) IMAGE TRANSLATION :

Algorithm :

1.importing the required libraries

2.import numpy as np

3.import cv2

4.import matplotlib.pyplot as plt

5.%matplotlib inline

6.#reading the image

7.image = cv2.imread('index.png')

8.#shifting the image 100 pixels in both dimensions

9.M = np.float32([[1,0,-100],[0,1,-100]])

10.dst = cv2.warpAffine(image,M,(cols,rows))

11.plt.imshow(dst)

ALGORITHM AND PSEUDOCODE OF PILLOW MODULE :

Beginning with Pillow

1.) Opening an image using open():

The PIL.Image.Image class represnts the image object. This class provides the open() method that is used to open the image.

```
# test.png => location_of_image
```

```
img = Image.open(r"test.png")
```

Note: Location of image should be relative only if the image is in the same directory as the Python program, otherwise absolute (full) path of the image should be provided.

2.) Displaying the image using show(): This method is used to display the image..

```
 img = Image.open(r"test.png")
img.show()
```

3.) Obtaining information about the opened image:

A) Getting the mode (color mode) of the image

```
img = Image.open(r"test.png")

print(img.mode)
```

B) Getting the size of the image: This attribute provides the size of the image. It returns a tuple that contains width and height.

```
img = Image.open(r"test.png")
print(img.size)
```

C) Getting the format of the image: This method returns the format of the image file.

```
img = Image.open(r"test.png")
print(img.format)
```

4.) Rotating an image using rotate(): After rotating the image, the sections of the image having no pixel values are filled with black (for non-alpha images) and with completely transparent pixels (for images supporting transparency)

```
 angle = 40
```

```
img = Image.open(r"test.png")

r_img = img.rotate(angle)
```

5.) Resizing an image using resize(): Interpolation happens during the resize process, due to which the quality of image changes whether it is being upscaled. Therefore resize() should be used cautiously and while providing suitable value for resampling argument.

```
 size = (40, 40)

img = Image.open(r"test.png")

r_img = img.resize(size)

r_img.show()
```

6.) Saving an image using save(): While using the save() method Destination_path must have the image filename and extension as well.

```
size = (40, 40)

img = Image.open(r"test.png")

r_img = img.resize(size, resample = Image.BILINEAR)

# resized_test.png => Destination_path

r_img.save("resized_test.png")

# Opening the new image

img = Image.open(r"resized_test.png")

print(img.size)
```

ALGORITHM AND PSEUDOCODE OF ENCRYPTION AND DECRYPTION :

1. To encrypt an image, first place that image in the input/ folder
2. Then run

python encrypt.py <image_name>
The encrypted image can be found at the encrypted_images/ folder.
The keys generated during encryption is stored in the keys.txt file.
(Note: The number of iterations of encryption to be performed can be adjusted by changing the ITER_MAX value in the encrypt.py file. Larger values will make encryption more secure but it is more time consuming)

3. To decrypt the image, run

python decrypt.py <image_name>

And Then enter the value of the Keys (Kr, Kc and ITER_MAX)
The decrypted image can be found at the decrypted_images/ folder

3.5 Testing Process

The testing process of this project is that whether the webcam is open automatically and capture the image or not. i.e test the accuracy of the computer working with the instructions given through the code. Hopefully, the program was successful in capturing the image using webcam.

Then, we edit the image and test whether the image is edited or not by the program using pillow module. If the image is edited then it automatically saved .

Then we use the cryptography for encryption and decryption of the image. Give the password as input and test the program ,if the output comes as a blurry image then the program was successful.

Then, we test whether the encrypted image open using the decryption program or not. Hopefully, the program was successful in detecting image.

4.REAL WORLD APPLICATIONS

Image security is of utmost concern as web attacks have become more and more serious. Image encryption decryption has applications in internet communication, multimedia systems, medical imaging, telemedicine, military communication. Since, these images may carry highly confidential information, so these images entail extreme protection when users amass somewhere over an unreliable repository. Furthermore, when people wish to transfer images over an insecure network, then it becomes crucial to provide an absolute protection. In brief, an image requires protection against various security attacks. To make the data secure from various attacks and for the integrity of data we must encrypt the data before it is transmitted or stored. Government, military, financial institution, hospitals and private business deals with confidential images about their patient (in Hospitals), geographical areas (in research), enemy positions (in defense), product, financial status. Most of this information is now collected and stored on electronic computers and transmitted across network to another computer.

The primary intention of keeping images protected is to maintain confidentiality, integrity and authenticity. Different techniques are available for making images secure and one technique is encryption. Adopting and applying various encryption algorithms can ensure security of data from spoofing and eavesdropping from the unauthorized attacks and crypto-analyst. However, the current forms of broadcasting and delivery of multimedia data through wireless channels, are highly insecure and vulnerable due to the inherent nature open access from massive users and receivers, if not properly encrypted. Generally, Encryption is a procedure that transforms an image into a cryptic image by using a key. Furthermore, a user can retrieve the initial image by applying a decryption method on the cipher image which is usually a reverse execution of the encryption process. The diverse algorithms are accessible to encrypt information, specifically; RSA, DES, AES, etc

Now days, the security of digital images has become immensely important in many applications- medical image, confidential video conferencing, defense database, mobile computing, personal communication etc.

5.RESULTS/OUTPUTS

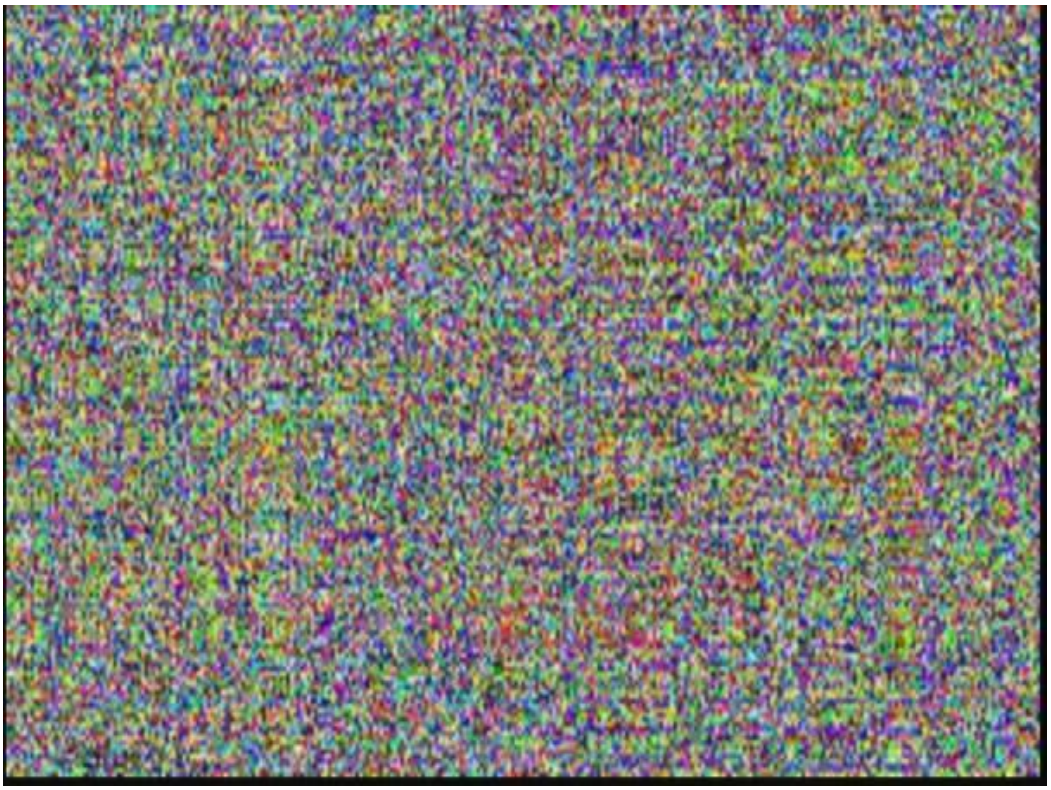
The result of the this project is the webcam capture the image then edit the image and save it in the pc then the image is encrypted so that the image is secured i.e No one can see the image without using decryption program or the password given during the encryption of the image. At last, using the decryption program the actual image popped up. In our project we are doing it in 4 different steps.

So, the outputs of the program or the project are shown below:-

1. Output of capturing the image using webcam. Using OpenCV tool for image processing and performing computer vision tasks.

2. Output of editing the image.

3. Output of encrypted Image



4. Output of decrypted image.

6.CONTRIBUTION

The world of work has changed. It used to be that most of us worked as a part of a process, whether on an assembly line, managing interactions with customers, or any one of a thousand other processes. Processes are ongoing, repeatable and never have an ending.

As we know the technologies like face recognition, encryption and decryption are already present but their use in our modern era is limited. Since from Day 1 I have clear idea about my project that what I am going to contribute to our Society as a Computer Science student.

My main aim is to provide "Security" to the existing system. As in nowadays use of internet is growing rapidly and at the same time the issue of security is also increasing rapidly. I have developed my project in the time of Covid, so I am sure that in today's world we need a compact and most reliable technology.

So, I also use the concept of Face Technology as in current scenario "Face Mask" is one of the most important parts of our life. So, using this we can detect the person who is not wearing the mask. Now, for Security concern I used the concept of Encryption and Decryption and I have given the feature of Image editing at the same platform, so user can do all the editing at the same platform decreasing time complexity and increasing easiness.

So, I have packed all the four things at a single place namely Face Recognition, Image Editing, Encryption and Decryption which can secure the photo, image and videos on the social networking site and in databases of different platform giving Authentication and Authorization of our work.

7.CONCLUSION

Hence, working throughout with the project it is concluded that MACHINE LEARNING is a vast filed that is playing a major role in today's era and will keep expanding it's limits in the upcoming years. Human minds could be invested to train their own made machines to work for their convenience. The study of matrices(i.e eigen values) and programming language are embedded to help the machines detect objects**.** Pillow is a fork of the Python Imaging Library (PIL). PIL is a library that offers several standard procedures for manipulating images. It's a powerful library, but hasn't been updated since 2011 and doesn't support Python 3. Pillow builds on this, adding more features and support for Python 3. It supports a range of image file formats such as PNG, JPEG, PPM, GIF, TIFF and BMP. We'll see how to perform various operations on images such as cropping, resizing, adding text to images, rotating, greyscaling, etc. using this library. In the past decade, image encryption is given much attention in research of information security and a lot of image encryption algorithms have been introduced. Open cv, pillow module ,encryption and decryption techniques are used in this project. In this project ,our motive is to secure the picture to get in to bad hands.

REFERNCES

The completion of this work was a work of total dedication and determination towards learning something new and acquiring a new skill.

The references regarding this project were taken from different sources like:

[1] You tube, course offering sites like Coursera, Udemy, Alison Courses and Shaw Academy.

[2] Unisys, Dr Glen E. Newton (2013-05-07). "The Evolution of Encryption". Wired. ISSN 1059-1028. Retrieved 2020-04-02.

[3] “Symmetric-key encryption software”

[4] Kessler, Gary (November 17, 2006). "An Overview of Cryptography". Princeton University.

[5] "History of Cryptography". Binance Academy. Retrieved 2020-04-02.

[6] "Caesar Cipher in Cryptography". GeeksforGeeks. 2016-06-02. Retrieved 2020-04-02.

[7] "Wheel Cipher". www.monticello.org. Retrieved 2020-04-02.

[8] "M-94". www.cryptomuseum.com. Retrieved 2020-04-02.

[9] Hern, Alex (2014-11-14). "How did the Enigma machine work?". The Guardian. ISSN 0261-3077. Retrieved 2020-04-02

[10] Naik Riddhi , Nikunj Gamit,” An Efficient Algorithm for Dynamic Key Generation for Image Encryption,” IEEE International Conference on Computer, Communication and Control (IC4-2015).

[11] Guodong Ye, Xiaoling Huang,” An image encryption algorithm based on auto-blocking and ECG signal,” IEEE MultiMedia,Volume: 23, Issue: 2, Apr.-June 2016.

[12] Mohit Kumar, Akshat Aggarwal, Ankit Garg,” A Review on Various Digital Image Encryption Techniques and Security Criteria,” International Journal of Computer Applications (0975 – 8887)Volume 96– No.13, June 2014

[13] Vishakha Kelkar, Hitesh Nemade," Reversible Watermarking in Medical Images Using Histogram Shifting Method with Improved Security and Embedding Capacity,"IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), May 2016.

[14] Philip P. Dang and Paul M. Chau,"Image Encryption For Secure Internet Multimedia Applications," IEEE Transactions on Consumer Electronics, Vol. 46, No. 3, AUGUST 2000

[15] Reza Moradi Rad, Abdolrahman Attar, and Reza Ebrahimi Atani," A New Fast and Simple Image Encryption Algorithm Using Scan Patterns and XOR," International Journal of Signal Processing, Image Processing and Pattern Recognition Vol.6, No.5 (2013), pp.275-290.