# ACKNOWLEDGEMENT

Great achievements are not possible without standing on the shoulders of giants. Without the active involvement of the following experts this project would not have been a reality.

We express our sincere gratitude to **Dr.P.Thamarai M.E, Ph.D.,** Principal, Government College of Technology, Coimbatore for providing us all facilities that we needed for the completion of this project.

We whole-heartedly express our thankfulness and gratitude to **Dr.S.Rathi M.E., Ph.D.,** Associate Professor and Head of the Department of Information Technology, Government College of Technology, for helping us to successfully carry out this project.

Our thankfulness and gratitude to our respectable project guide **Mrs.T.Suguna M.Tech.,** Assistant Professor, who has been of immense help through the various phases of the project. With her potent ideas and excellent guidance, we were able to comprehend the essential aspects involved.

We would like to thank our faculty advisor **Mrs.R.Malavika M.Tech.,** Assistant Professor for her continuous support and encouragement throughout this project.

We extend our sincere thanks to the staff members of Information Technology department, **Mrs.R.Devi M.Tech.,** Assistant Professor, **Thiru.S.Gladson Oliver M.Tech.,** Assistant Professor, **Ms.M.Jeyanthi M.Tech.,** Assistant Professor, **Thiru.M.Gowri Shankar M.E.,** Assistant Professor, **Mrs.R.Malavika M.Tech.,** Assistant Professor, **Dr.M.Blessy Queen Mary M.E.,Ph.D.,** Assistant Professor, for rendering their help for the completion of this project. We also thank all our friends for their cooperation and suggestions towards the successful completion of this project.

iii

# SYNOPSIS

Face Recognition is a technology capable of matching a human face from a digital image or a video frame against a database of a face works by pinpointing and measuring facial features from a given image. Face Recognition techniques authenticate users through ID verification services. These techniques are used for identifying people in photos, video, or in real-time etc. System generates the attendance of the student on the basis of presence in class. In the existing attendance management system we have to mark attendance manually and it takes a lot of time. The model detects the person by analyzing the image and provides the exact result after the extraction of the facial features. This technique ensures feasibility and provides a better solution for real time problems. The proposed method is based on capturing and detecting images using face recognition methodology.

iv

# CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO**  **1** | **TITLE BONAFIDE CERTIFICATE ACKNOWLEDGEMENT SYNOPSIS**  **TABLE OF CONTENTS LIST OF FIGURES**  **LIST OF ABBREVATIONS**  **INTRODUCTION** | **PAGE NO**  i iii iv v xi  xii  **1-3** |
|  | 1.1 DESCRIPTION | 1 |
|  | 1.2 EXISTING SYSTEM | 2 |
|  | 1.3 PROBLEM DEFINITION | 2 |
|  | 1.4 PROPOSED SYSTEM | 2 |
|  | 1.5 ORGANIZATION OF PROJECT | 3 |

### LITERATURE REVIEW 4-17

2.1 AUTOMATIC ATTENDANCE MANAGEMENT SYSTEM 4

USING FACE DETECTION

* + 1. DESCRIPTION

2.1.2 MERIT

2.1.3 DEMERIT

* 1. 2.2 CLASS ATTENDANCE MANAGEMENT SYSTEM 5
  2. USING FACE RECOGNITION

v

* + 1. DESCRIPTION
    2. MERIT
    3. DEMERIT

2.3 AUTOMATED ATTENDANCE SYSTEM USING 6

IMAGE PROCESSING

* + 1. DESCRIPTION
    2. MERIT
    3. DEMERIT

2.4 REAL TIME ATTENDANCE USING FACE 7

RECOGNITION TECHNIQUE

* + 1. DESCRIPTION
    2. MERIT

2.4.3 DEMERIT

2.5 REAL TIME SMART ATTENDANCE SYSTEM USING 8

FACE RECOGNITION TECHNIQUES

* + 1. DESCRIPTION
    2. MERIT
    3. DEMERIT

2.6 STUDENT ATTENDANCE SYSTEM USING 9

FACE RECOGNITION

* + 1. DESCRIPTION
    2. MERIT

2.6.3 DEMERIT

vi

### 3 SYSTEM SPECIFICATION 18-19

* 1. SYSTEM REQUIREMENTS

3.1.1 HARDWARE REQUIREMENT 18

3.1.2 SOFTWARE REQUIREMENT 18

3.2 SOFTWARE DESCRIPTION 18

3.2.1 ABOUT PYTHON 18

3.2.2 CHARACTERISTICS OF PYTHON 18

3.2.3 ABOUT FLASK 18

* + 1. CHARACTERISTICS OF FLASK 18

**4 PROJECT DESIGN 20-26**

* 1. [MODULE DESCRIPTION OF ADMIN PHASE 20](#_heading=h.30j0zll)

4.2 [MODULE DESCRIPTION OF TEACHERS PHASE 21](#_heading=h.1fob9te)

1. **5 IMPLEMENTATION AND RESULT 27-41**
   1. [IMPLEMENTATION](#_heading=h.tyjcwt)
      1. [main.m 27](#_heading=h.3dy6vkm)
      2. [main\_encode.m 29](#_heading=h.1t3h5sf)
      3. [preprocess\_image.m 29](#_heading=h.4d34og8)
      4. [data\_embedding.m 30](#_heading=h.2s8eyo1)
      5. grs2rgb.m 32
      6. [main\_decode.m 33](#_heading=h.17dp8vu)
      7. [re-preprocess\_image.m 33](#_heading=h.3rdcrjn)
      8. [data\_extraction.m 34](#_heading=h.26in1rg)
      9. [structuralsimilarity.m 36](#_heading=h.lnxbz9)
   2. [SAMPLE OUTPUT](#_heading=h.35nkun2)
      1. TEST IMAGES 37
      2. COVER IMAGE 38
      3. HISTOGRAM OF ORIGINAL IMAGE 38
      4. EMBEDDED IMAGE 39
      5. HISTOGRAM OF EMBEDDED IMAGE 39
   3. [PARAMETERS SETTING 39](#_heading=h.1ksv4uv)
   4. [EXPERIMENTAL RESULTS 40](#_heading=h.44sinio)
2. **CONCLUSION 42**

8.1 CONCLUSION 42

**REFERENCES 43**

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **TITLE** | **PAGE NO** |
| 4.1 | EMBEDDING PHASE | 20 |
| 4.2 | EXTRACTING PHASE | 21 |
| 4.4.1 | EMBEDDING PROCESS IN 2D HISTOGRAM | 24 |
| 4.5.1 | EXTRACTION PROCESS IN 2D HISTOGRAM | 26 |
| 5.2.1 | TEST IMAGES | 33 |
| 5.2.2 | COVER IMAGE | 33 |
| 5.2.3 | HISTOGRAM OF COVER IMAGE | 34 |
| 5.2.4 | EMBEDDED IMAGE | 34 |
| 5.2.5 | HISTOGRAM OF EMBEDDED IMAGE | 34 |

# LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| LBP | LOCAL BINARY PATTERN |
| LBPH | LOCAL BINARY PATTERN  HISTOGRAM |
| PCA | PRINCIPAL COMPONENT ANALYSIS |
| CNN | CONVOLUTIONAL NEURAL NETWORK |
| KNN | K-NEAREST NEIGHBOUR |
| SVM | SUPPORT VECTOR MACHINE |

**CHAPTER 1 INTRODUCTION**

* 1. **DESCRIPTION**

Attendance Management System is an application developed for daily evaluation of student’s attendance. It is facilitated to access the information of attendance of a particular Student in a particular subject of study. The information is sorted by the faculties and admin, as provided by the system for a particular student throughout a complete semester. This system will also enable the evaluation of student regular presence in various subjects.///

The availability of the internet everywhere and the risks of illegal accessing of transmitted data is increasing day by day. Therefore, there is a need for protection of secret data from unauthorized users in a public network. Data hiding is one of the most important techniques to protect the security of digital media. Data hiding is used in a wide range of applications for embedding confidential messages. Sometimes hiding the information demolishes the host image even though the distortion is not visible to human eyes. In some applications such as medical imagery, military imagery and law forensics, no degradation of the original cover is allowed. In these cases, we need an important data hiding method known as Reversible Data Hiding or lossless data hiding. Reversible data hiding techniques are devised to solve the problems of lossless embedding of larger messages. The image can be restored completely without any loss. Contrast enhancement is executed to bring out unclear details and is widely used in medical images.

# EXISTING SYSTEM

In the existing methods, some of the disadvantages are listed.

* Staff has to take attendance manually.
* It is a tedious job to maintain records.

1

* It is difficult to generate consolidated reports.
* Sheets may get lost.

# PROBLEM DEFINITION ////

* This project attempts to make use of two dimensional histogram to perform the task of data hiding for the purpose of preserving privacy.
* Though many data hiding methods do the same, our algorithm provides contrast enhancement of images & high data embedding rate.
* Data hiding in a reversible manner will provide double security for the data such as image encryption as well as data hiding in encrypted images.

# PROPOSED SYSTEM

* To overcome the drawbacks of the existing system proposed has been evolved.
* It aims to reduce the paperwork and save time for generating the accurate results from the attendance system.
* The system provides the best user interface and efficient reports are generated.

# ORGANIZATION OF THE PROJECT

* Literature reviews of already existing proposals are discussed in chapter 2.
* Chapter 3 has system specification which tells about the software and hardware requirements.
* Chapter 4 discusses the overall project and design which tells the brief description of each of the modulus in this project.
* Chapter 5 has the implementation and experimental result of the project.
* Chapter 6 deals with the conclusion and future work.
* Finally chapter 7 deals with the references.

# CHAPTER 2

# LITERATURE REVIEW

* 1. **AUTOMATIC ATTENDANCE MANAGEMENT SYSTEM**

**USING FACE DETECTION [2016]**

* + 1. **2.1.1 DESCRIPTION**

This paper is about biometric attendance management. In this method camera is fixed in the classroom. After fixing the camera the image will be captured. During the training phase all students’ images are trained. In the captured image the faces are recognized and faces get cropped from the captured image. During the attendance marking phase the recognized faces are checked against the database image. And if the image is matched with the database image the attendance gets marked. If the image is not found it will ask the admin to add the student image in the database. If the attendance is marked as absent it will send an alert to the concerned student parent. For the face recognition it uses a method called Eigen face. Eigen face is one of the fastest approaches. The Eigen face method decomposes images into small features. Discrete images are mapped using linear values of Eigen face.

Finally all the faces are recognized using the best Eigen face value.

# 2.1.2 MERIT

* Hardware installation is easy.

# 2.1.3 DEMERIT

* Accuracy is low and face recognition is difficult.

# CLASS ATTENDANCE MANAGEMENT SYSTEM USING FACE RECOGNITION [2018]

# 4

**2.2.1 DESCRIPTION**

This paper proposes a comprehensive embedded class attendance system with controlling the door access. The proposed scheme is based on face recognition. By using Local Binary Pattern algorithm (LBP) the face gets recognized. After capturing image it will be passed to the raspberry pi which handles the implementation of face recognition algorithm. If the student image was matched the door will be opened with the help of servo motor. In addition, the attendance result will be stored in the MySQL database. The experimental results show that higher accuracy can be achieved with our proposed one. The numerical results are provided to show the performances of the proposed scheme in different cases of face recognition.

# MERIT

* Easily accessible.

# DEMERIT

* Less sensitivity and not effective in darkness.

**2.3 AUTOMATED ATTENDANCE SYSTEM USING IMAGE PROCESSING [2018]**

**2.3.1 DESCRIPTION**

This paper proposes an idea of using image detection and recognition which can automatically handle the attendance system. In this paper, Viola Jones algorithm used for face detection and Fisher Face Algorithm are used for face recognition. At first video segments are captured from the classroom. Then, the face was detected by preprocessing the video input .Face cropping has been done At last; the cropped image would be compared with the database. If the cropped face matches with the database it marks the attendance and if the image is not found it will ask the user to register.

# 2.3.2 MERIT

* Maintains attendance in high efficient manner

# 2.3.3 DEMERIT

* Accuracy rate is less than 50%

# 2.4 REAL TIME SMART ATTENDANCE SYSTEM USING FACE RECOGNITION TECHNIQUES [2019]

**2.4.1 DESCRIPTION**

Generally attendance system is executed with the help of biometrics. Face recognition is one of the best methods to improve this system. This paper proposes a method to automate the attendance management system by making use of recognition techniques like Eigen face values, Principal Component Analysis(PCA),Convolution Neural Network(CNN).Firstly student has to enroll and details will be stored in the database and the camera will be fixed outside the classroom. If the face is not present in the database it prompts the student to register. In face recognition module the images are recovered from the camera inside the classroom. It will mark present for the students who are present in the class. Marked attendance is stored in the database for the future purpose.

# 2.4.2 MERIT

* Proxy less with high accuracy

# 2.4.3 DEMERIT

* Less efficient

# 2.5 REAL TIME ATTENDANCE USING FACE RECOGINITION TECHNIQUE [2020]

**2.5.1 DESCRIPTION**

This paper is about integrating the face recognition technique with Open Source Computer Vision (OpenCV) for attendance management system. It will facilitate the attendance automation process and enable staff to enquire student details based on the check in and checkout time. Firstly the camera has been installed in the classrooms. After that image will be captured and the captured image will be passed to the face recognition module. The recognized images are checked with images of the students which is stored in a database with individual id .if the image is not found in the database by performing haar cascade method student can add their image into the database. Student information and attendance are stored in Excel Sheet.

# 2.5.2 MERIT

* Accuracy high in smaller area

# 2.5.3 DEMERIT

* Detection rate is low in larger area

# 

# 2.6 ATTENDANCE MANAGEMENT SYSTEM USING FACE RECOGNITION [2020]

**2.6.1 DESCRIPTION**

This paper describes the face recognition technique used for the attendance management system. It uses an algorithm called cascade classifier and Local Binary Pattern Histogram (LBPH) for face recognition. In this application students have to register by entering the ID and Name registered with Face recognition. This data will be stored in database. Local Binary Pattern is a basic algorithm used to detect face from front side. Haar cascade is based on Haar wavelet technique to analyze the pixels from the captured image. The recognized images are checked with images of the students which is stored in a database with individual id .if the image is not found in the database by performing haar cascade method student can add their image into the database. Student information and attendance are stored and it will be available for the respective staffs.

# 2.6.2 MERIT

* Robust Against monotonic grey image

# 2.6.3 DEMERIT

* Take more than 5 secs to recognize.

**2.7 STUDENT ATTENDANCE SYSTEM USING FACE RECOGNITION**

**[2020]**

**2.7.1 DESCRIPTION**

This paper describes the face recognition technique used for the attendance management system.It uses an algorithm called haar cascade classifier,K-Nearest Neighbour(KNN),Convolutional Neural Network(CNN),Support Vector Machine(SVM)and Local Binary Pattern Histogram(LBPH) for face recognition.In this application attendance report will generated and stored in excel format .This system is tested against various conditions.The system developed need less installation .The recognised images are checked with images of the students which is stored in a database .if the image is not found in the database student can add their image into the database.Student information and attendance are stored and it will be available in data base.

# 2.7.2 MERIT

* + - 1. Cost efficient

# 2.7.3 DEMERIT

* Less computational Complexity

# CHAPTER 3

# SYSTEM SPECIFICATION

* 1. **SYSTEM REQUIREMENTS**
     1. **HARDWARE REQUIREMENTS**
* System : Pentium IV 2.4 GHz.
* Hard Disk : 1 TB
* Ram : 4GB.

# SOFTWARE REQUIREMENTS

* Operating system : Windows 10.
* Coding Language : PYTHON
* Tool : PYTHON 3.9,FLASK

# SOFTWARE DESCRIPTION

* + 1. **ABOUT PYTHON**
    2. PYTHON is an interpreted,object-oriented, high level programming language with
    3. dynamic semantics.It supports modules and packages which encourages code reuse.

.Typical uses include:

* + - * User-friendly Data Structures
      * Software development
      * Writing system scripts
      * Data analysis, exploration, and visualization
      * Scientific and engineering graphics

PYTHON is often used as a scripting language for Web applications.It means it can automate specific series of tasks in an efficient way.It works on different platforms like Windows,Mac,Linux,etc.

# CHARACTERISTICS OF PYTHON

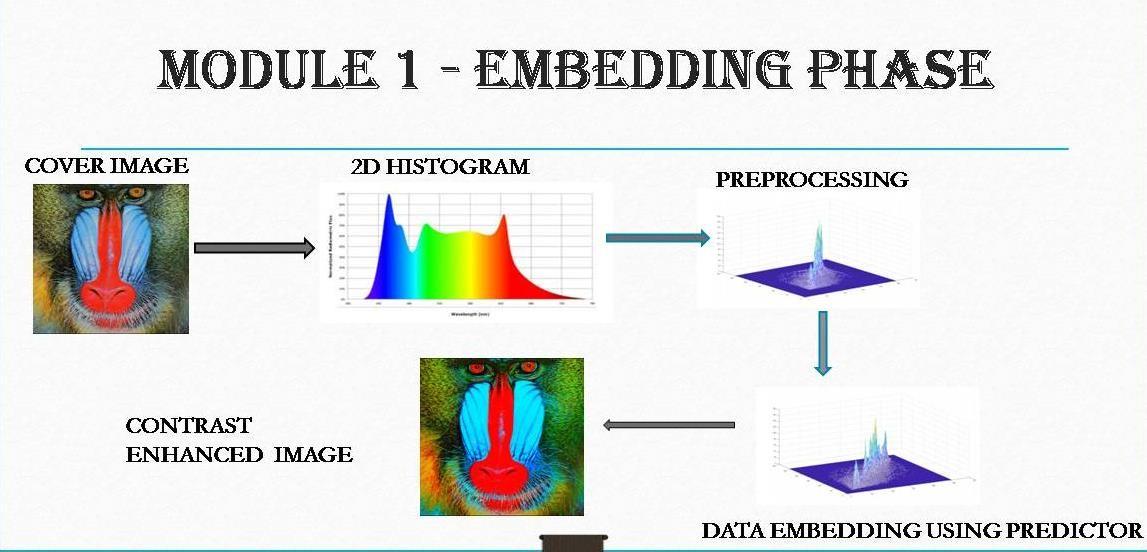
* Python is a very easy to use developer-friendly language.
* Open Source Programming language.
* Support a wide area of GUI.
* Easily Portable language.
* Supports other Programming Languages like C,C++,Java,etc.
* Tools for building applications with custom user interfaces.
  + 1. **ABOUT FLASK**
    2. FLASK is a micro web framework written in python. It does not require specific tools or
    3. libraries. It allows us to build web applications. It will not force any dependency.
    4. It has features like url routing, template engine.

# CHARACTERISTICS OF FLASK

* + - * Development server and debugger.
      * Integrated support for unit testing.
      * RESTful request dispatching
      * Uses Jinja templating.
      * Support for secure cookies (client side sessions)

# CHAPTER 4 PROJECT DESIGN

# 4.1 MODULE DESCRIPTION OF EMBEDDING PHASE



* 1. **FIGURE – EMBEDDING PHASE Inputs**: Cover image and secret text

**Outputs**: Embedded image (image containing a hidden secret text)

**Step 1** Read the cover color image(I).

**Step 2** Generate a 2D histogram of *I* and globally shifting the histogram.

**Step 3** Perform preprocessing by merging adjacent rows and columns on each side for S times of a two dimensional histogram and generating the location map.

**Step 4** Append to the payload certain information such as the location map’s size, location map and Least Significant Bits of t+2 border pixels.

**Step 5** Perform the steps (6 - 9) repeatedly for each predictor.

**Step 6** Replace the t+2 border pixel Least Significant Bits with the payload length and predictor number.

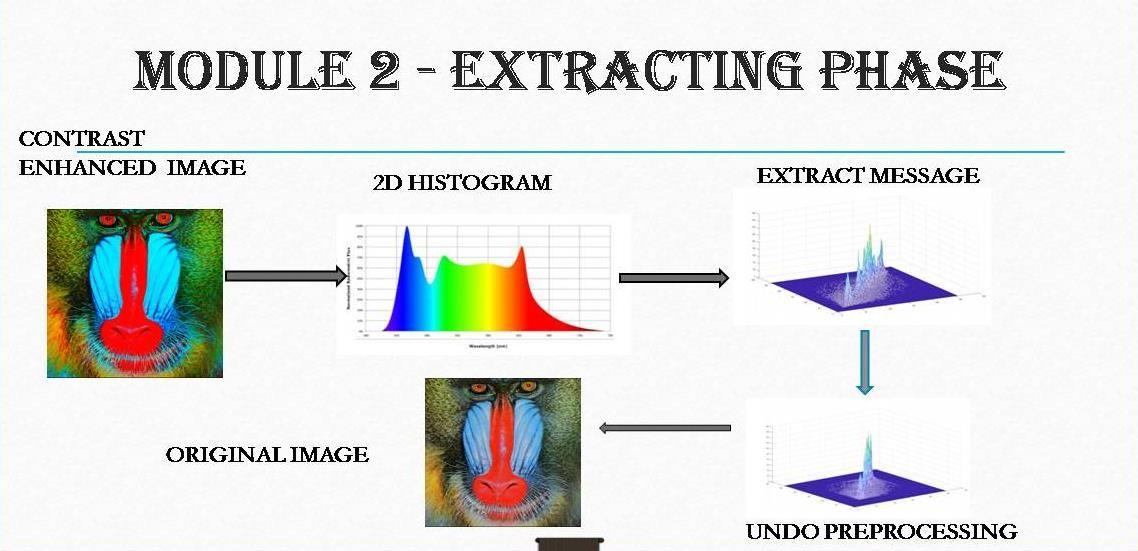
**Step 7** Sort the even pixels. Use Positive Histogram Shifting and then Negative Histogram Shifting to embed the first half of payload.

**Step 8** Sort the odd pixels. Use Positive Histogram Shifting and then Negative Histogram Shifting to embed the first half of payload.

**Step 9** Calculate the PSNR between original and each of the embedded images.

**Step 10** Generate the embedded image with the highest PSNR.

# MODULE DESCRIPTION OF EXTRACTING PHASE



### FIGURE- EXTRACTING PHASE

**Inputs:** Embedded image

**Outputs:** Original image, Secret text

**Step 1** Read the embedded image.

**Step 2** Collect the t+2 border pixel least significant bits and to extract the predictor number and payload length.

**Step 3** Sort the odd pixels. Use Positive Histogram Shifting and then Negative Histogram Shifting to extract the second half of payload.

**Step 4** Sort the even pixels. Use Positive Histogram Shifting and then Negative Histogram Shifting to extract the first half of payload.

**Step 5** Replace the t+2 border pixel Least Significant Bits with the original least significant bits.

**Step 6** Reverse the preprocessing strategy by uncompressing the location map.

**Step 7** Use the extracted global shifting information to shift the whole 2D histogram to recover the original image.

**Step8** The extracted image is now formed.

**Step9** The secret text is now extracted.

# CHAPTER 5 IMPLEMENTATION AND RESULTS

# IMPLEMENTATION

Implementation is the most crucial stage in achieving a successful system and giving the users confidence that the new system is effective and workable. Implementation of this project refers to the installation of the packages in its real environment to the full satisfaction of the users and operations of the system. Testing is done individually at the time of development using the data and verification is done the way specified in the program specification. In short, implementation constitutes all activities that are required to put an already tested and completed package into operation. The success of any information system lies in its successful implementation. System implementation is the stage in the project where the theoretical design is turned into a working system. The most critical stage is achieving a successful system and in giving confidence on the new system for the user that it will work efficiently and effectively. The existing system was a long time process.

# Admin - app.py

* 1. **Packages :**
  2. **from flask import Flask,render\_template,request,redirect,url\_for,session**
  3. **import MySQLdb**
  4. **import os**
  5. **#import for face recognition**
  6. **from math import sqrt**
  7. **from sklearn import neighbors**
  8. **from os import listdir**
  9. **from os.path import isdir, join, isfile, splitext**
  10. **import pickle**
  11. **from PIL import Image, ImageFont, ImageDraw, ImageEnhance**
  12. **import face\_recognition**
  13. **from face\_recognition import face\_locations**
  14. **from face\_recognition.face\_recognition\_cli import image\_files\_in\_folder**
      1. **Authentication System :**

**app = Flask(\_\_name\_\_)**

**APP\_ROOT = os.path.dirname(os.path.abspath(\_\_file\_\_))**

**print(APP\_ROOT)**

**conn = MySQLdb.connect(host="localhost",user="root",password="Arasii@5670",db="login\_info")**

**@app.route('/')**

**def index():**

**return render\_template("index.html",title="Admin Login")**

**@app.after\_request**

**def set\_response\_headers(response):**

**response.headers['Cache-Control'] = 'no-cache, no-store, must-revalidate'**

**response.headers['Pragma'] = 'no-cache'**

**response.headers['Expires'] = '0'**

**return response**

**@app.route('/login',methods=['POST'])**

**def login():**

**user = str(request.form["user"])**

**paswd = str(request.form["password"])**

**cursor = conn.cursor()**

**result = cursor.execute("SELECT \* from admin\_login where binary username=%s and binary password=%s",[user,paswd])**

**if(result==1):**

**return render\_template("task.html")**

**else:**

**return render\_template("index.html",title="Admin Login",msg="The username or password is incorrect")**

**@app.route('/register\_teacher',methods=['POST'])**

**def register\_teacher():**

**return render\_template("signup.html",title="SignUp")**

**@app.after\_request**

**def set\_response\_headers(response):**

**response.headers['Cache-Control'] = 'no-cache, no-store, must-revalidate'**

**response.headers['Pragma'] = 'no-cache'**

**response.headers['Expires'] = '0'**

**return response**

**@app.route('/student',methods=['POST'])**

**def file\_upload():**

**return render\_template("upload.html")**

**@app.route('/signup\_teacher',methods=['POST'])**

**def signup():**

**user = str(request.form["user"])**

**paswd = str(request.form["password"])**

**email = str(request.form["email"])**

**cursor = conn.cursor()**

**result = cursor.execute("SELECT \* from teacher\_login where binary username=%s",[user])**

**print (result)**

**if(result == 1):**

**return render\_template("signup.html",title="SignUp",uname=user,msg="already present")**

**cursor.execute("INSERT INTO teacher\_login (username,password,email) VALUES(%s, %s, %s)",(user,paswd,email))**

**conn.commit()**

**return render\_template("signup.html",title="SignUp",msg="successfully signup",uname=user)**

**@app.route('/signup\_student',methods=['POST'])**

**def signup\_student():**

**user = str(request.form["student\_name"])**

**email = str(request.form["student\_email"])**

**roll\_id = str(request.form["roll\_id"])**

**email1 = str(request.form["parent\_email"])**

**cursor = conn.cursor()**

**result = cursor.execute("SELECT \* from student\_login where binary roll\_id=%s",[roll\_id])**

**print (result)**

**if(result == 1):**

**return render\_template("upload.html",uname=user,msg=" already present")**

**cursor.execute("INSERT INTO student\_login (username,student\_email,parent\_email,roll\_id) VALUES(%s, %s, %s, %s)",(user,email,email1,roll\_id))**

**conn.commit()**

**return render\_template("upload.html",uname=user,msg=" successfully signup")**

**@app.route("/upload", methods=['POST'])**

**def upload():**

**target = os.path.join(APP\_ROOT,"train/")**

**if not os.path.isdir(target):**

**os.mkdir(target)**

**classfolder = str(request.form['class\_folder'])**

**session['classfolder'] = classfolder**

**target1 = os.path.join(target,str(request.form["class\_folder"])+"/")**

**session['target1']=target1**

**print(target1)**

**model = os.path.join(APP\_ROOT,"model/")**

**if not os.path.isdir(model):**

**os.mkdir(model)**

**classname = str(request.form['class\_folder']+"/")**

**model = os.path.join(model,classname)**

**if not os.path.isdir(model):**

**os.mkdir(model)**

**session['model']=model**

**session['classname'] = classname**

**if not os.path.isdir(target1):**

**os.mkdir(target1)**

**id\_folder = str(request.form["id\_folder"])**

**session['id\_folder']= id\_folder**

**target2 = os.path.join(target1,id\_folder+"/")**

**if not os.path.isdir(target2):**

**os.mkdir(target2)**

**target3 = os.path.join(target2,id\_folder+"/")**

**if not os.path.isdir(target3):**

**os.mkdir(target3)**

**for file in request.files.getlist("file"):**

**print(file)**

**filename = file.filename**

**destination = "/".join([target3,filename])**

**print(destination)**

**file.save(destination)**

**return call\_train()**

* 1. **This function is responsible for training student data through face recognition and storing it in a database. The whole training process involves preprocessing the image and the model generation.**

def train(train\_dir, model\_save\_path = "", n\_neighbors = None, knn\_algo = 'ball\_tree', verbose=True):

id\_folder=str(session.get('id\_folder'))

X = []

y = []

z = 0

for class\_dir in listdir(train\_dir):

if not isdir(join(train\_dir, class\_dir)):

continue

for img\_path in image\_files\_in\_folder(join(train\_dir, class\_dir)):

image = face\_recognition.load\_image\_file(img\_path)

faces\_bboxes = face\_locations(image)

if len(faces\_bboxes) != 1:

if verbose:

print("image {} not fit for training: {}".format(img\_path, "didn't find a face" if len(faces\_bboxes) < 1 else "found more than one face"))

os.remove(img\_path)

z = z + 1

continue

X.append(face\_recognition.face\_encodings(image, known\_face\_locations=faces\_bboxes)[0])

y.append(class\_dir)

print(listdir(train\_dir+"/"+id\_folder))

train\_dir\_f = listdir(train\_dir+"/"+id\_folder)

for i in range(len(train\_dir\_f)):

if(train\_dir\_f[i].startswith('.')):

os.remove(train\_dir+"/"+id\_folder+"/"+train\_dir\_f[i])

print(listdir(train\_dir+"/"+id\_folder))

if(listdir(train\_dir+"/"+id\_folder)==[]):

return render\_template("upload.html",msg1="training data empty, upload again")

elif(z >= 1):

return render\_template("upload.html",msg1="Data trained for "+id\_folder+", But one of the image not fit for training")

if n\_neighbors is None:

n\_neighbors = int(round(sqrt(len(X))))

if verbose:

print("Choose n\_neighbors automatically as:", n\_neighbors)

knn\_clf = neighbors.KNeighborsClassifier(n\_neighbors=n\_neighbors, algorithm=knn\_algo, weights='distance')

knn\_clf.fit(X, y)

if model\_save\_path != "":

with open(model\_save\_path, 'wb') as f:

pickle.dump(knn\_clf, f)

return render\_template("upload.html",msg1="Data trained for "+ id\_folder)

**Model Generation :**

def call\_train():

id\_folder = str(session.get('id\_folder'))

model=str(session.get('model'))

if not os.path.isdir(model + id\_folder):

os.mkdir(model + id\_folder)

model = model + id\_folder + "/"

model = model + "model"

target1=str(session.get('target1'))

print(id\_folder)

print (target1)

target1 = target1 +id\_folder

print(target1)

print(model)

return train(train\_dir=target1,model\_save\_path=model)

**Teachers : app.py**

**Packages :**

**from flask import Flask,render\_template,request,redirect,url\_for,session**

**from flask\_bootstrap import Bootstrap**

**import MySQLdb**

**import os**

**from math import sqrt**

**from sklearn import neighbors**

**from os import listdir**

**from os.path import isdir, join, isfile, splitext**

**import shutil**

**import pickle**

**from PIL import Image, ImageFont, ImageDraw, ImageEnhance**

**import face\_recognition**

**from face\_recognition import face\_locations**

**from face\_recognition.face\_recognition\_cli import image\_files\_in\_folder**

**from datetime import datetime,timedelta**

**from pytz import timezone**

**import xlsxwriter**

**import pandas as pd**

**from glob import glob**

**from flask\_mail import Mail, Message**

**from io import BytesIO**

**import base64**

**import lable\_image**

**Image Capturing :**

**def upload():**

**if not os.path.isfile(APP\_ROOT+"/image.jpeg"):**

**return render\_template("upload.html",msg="spoof detected")**

**id\_folder = str(request.form['id\_folder'])**

**session['id\_folder']= id\_folder**

**target = os.path.join(APP\_ROOT,"test/")**

**if not os.path.isdir(target):**

**os.mkdir(target)**

**target1 = os.path.join(target,str(request.form["folder\_name"])+"/")**

**test\_append = str(request.form["folder\_name"])**

**session['test\_append']= test\_append**

**print(target1)**

**if not os.path.isdir(target1):**

**os.mkdir(target1)**

**shutil.copyfile(APP\_ROOT+"/"+"image.jpeg",target1+"image.jpeg")**

**destination = APP\_ROOT + "/" + "test/" + test\_append + "/" + "image.jpeg"**

**session['destination'] = destination**

**teacher\_name = str(session.get('user'))**

**session['teacher\_name'] = teacher\_name**

**#return render\_template("upload.html",msg="uploaded successfully")**

**return match()**

**Image Detection :**

**def match():**

**destination = str(session.get('destination'))**

**print(destination)**

**if os.path.isfile(destination):**

**test\_append = str(session.get('test\_append'))**

**session['test\_append'] = test\_append**

**id\_folder = str(session.get('id\_folder'))**

**train\_dir = APP\_ROOT1[0]+"admin\_site/train/"+ test\_append**

**try:**

**model = APP\_ROOT1[0]+"admin\_site/model/"+test\_append+"/" + id\_folder + "/" +"model"**

**print(model)**

**return predict1(model)**

**except FileNotFoundError:**

**os.remove(APP\_ROOT1[0]+"teachers\_site/image.jpeg")**

**return render\_template("upload.html",msg="trained model not present for " + test\_append + ": "+id\_folder)**

**def predict(X\_img\_path, knn\_clf = None, model\_save\_path ="", DIST\_THRESH = .45):**

**if knn\_clf is None and model\_save\_path == "":**

**raise Exception("must supply knn classifier either thourgh knn\_clf or model\_save\_path")**

**if knn\_clf is None:**

**with open(model\_save\_path, 'rb') as f:**

**knn\_clf = pickle.load(f)**

**X\_img = face\_recognition.load\_image\_file(X\_img\_path)**

**X\_faces\_loc = face\_locations(X\_img)**

**if len(X\_faces\_loc) == 0:**

**return []**

**faces\_encodings = face\_recognition.face\_encodings(X\_img, known\_face\_locations=X\_faces\_loc)**

**closest\_distances = knn\_clf.kneighbors(faces\_encodings, n\_neighbors=1)**

**is\_recognized = [closest\_distances[0][i][0] <= DIST\_THRESH for i in range(len(X\_faces\_loc))]**

**return [(pred) if rec else ("unknown") for pred, rec in zip(knn\_clf.predict(faces\_encodings), is\_recognized)]**

**def predict1(model):**

**test\_append = str(session.get('test\_append'))**

**test\_dir = APP\_ROOT1[0]+"teachers\_site/test/" + test\_append**

**f\_preds = []**

**for img\_path in listdir(test\_dir):**

**preds = predict(join(test\_dir, img\_path) ,model\_save\_path=model)**

**f\_preds.append(preds)**

**print(f\_preds)**

**print(len(preds))**

**print(len(f\_preds))**

**for i in range(len(f\_preds)):**

**if(f\_preds[i]==[]):**

**os.remove(APP\_ROOT1[0]+"teachers\_site/image.jpeg")**

**return render\_template("upload.html",msg="upload again, face not found")**

**else:**

**os.remove(APP\_ROOT1[0]+"teachers\_site/image.jpeg")**

**excel = os.path.join(APP\_ROOT,"excel/")**

**if not os.path.isdir(excel):**

**os.mkdir(excel)**

**excel1 = os.path.join(excel,test\_append)**

**if not os.path.isdir(excel1):**

**os.mkdir(excel1)**

**teacher\_name = str(session.get('teacher\_name'))**

**excel2 = os.path.join(excel1,teacher\_name)**

**if not os.path.isdir(excel2):**

**os.mkdir(excel2)**

**session['excel2'] = excel2**

**excel3 = excel2+"/"+date+'.xlsx'**

**if not os.path.isfile(excel3):**

**workbook = xlsxwriter.Workbook(excel2+"/"+date+'.xlsx')**

**worksheet = workbook.add\_worksheet()**

**worksheet.set\_column(0,0,20)**

**worksheet.write('A1','Roll Id')**

**f\_preds.sort()**

**row = 1**

**col = 0**

**if f\_preds[0][0] == 'unknown':**

**return render\_template("upload.html",msg= "Student Not Matched")**

**for i in range(len(f\_preds)):**

**for j in range(len(f\_preds[i])):**

**worksheet.write\_string(row,col,f\_preds[i][j])**

**row += 1**

**workbook.close()**

**return render\_template("upload.html",msg= f\_preds[0][0] + " present")**

**else:**

**df = pd.read\_excel(excel2+"/"+date+'.xlsx')**

**writer = pd.ExcelWriter(excel2 + "/" + date+'.xlsx')**

**df.to\_excel(writer,sheet\_name="Sheet1",index=False)**

**workbook = writer.book**

**worksheet = writer.sheets['Sheet1']**

**rows=df.shape[0]**

**worksheet.write\_string(rows+1,0,f\_preds[0][0])**

**writer.save()**

**df = pd.read\_excel(excel2+"/"+date+'.xlsx')**

**df.drop\_duplicates(['Roll Id'],keep='first',inplace=True)**

**# result = df.sort\_values("Roll Id")**

**writer = pd.ExcelWriter(excel2 + "/" + date+'.xlsx')**

**df.to\_excel(writer,'Sheet1',index=False)**

**workbook = writer.book**

**worksheet = writer.sheets['Sheet1']**

**worksheet.set\_column(0,0,20)**

**writer.save()**

**return render\_template("upload.html",msg= f\_preds[0][0] + " present")**

**Attendance View :**

**def view():**

**test\_append = str(request.form['folder\_name'])**

**session['test\_append']=test\_append**

**teacher\_name = str(session.get('user'))**

**excel\_dir = APP\_ROOT+"/excel/"+test\_append+"/"+teacher\_name+"/"**

**excel\_date = request.form['fname']**

**time = request.form['ftime']**

**time = time[:2]**

**print(time)**

**final\_excel=glob(excel\_dir + "/" + excel\_date+ "@" + time +"\*.xlsx")[0]**

**print(final\_excel)**

**df = pd.read\_excel(final\_excel)**

**df.index += 1**

**return render\_template("files.html",msg=final\_excel,df=df,date=excel\_date+"@"+time+"hrs")**

**Mail Configuration :**

**app.config.update(**

**DEBUG = True,**

**#Email settings**

**MAIL\_SERVER = 'smtp.gmail.com',**

**MAIL\_PORT = 465,**

**MAIL\_USE\_SSL = True,**

**MAIL\_USERNAME = 'college1118@gmail.com',**

**MAIL\_PASSWORD = 'yog12345',**

**MAIL\_DEFAULT\_SENDER = 'college1118@gmail.com'**

**)**

**mail = Mail(app)**

**# declaring timezone then creating custom date format**

**india = timezone('Asia/Kolkata')**

**date = str(datetime.now(india))[:10] + "@" + str(datetime.now())[11:13] + "hrs"**

**Mail alert :**

**def send\_mail():**

**test\_append = str(request.form['folder\_name'])**

**teacher\_name = str(session.get('user'))**

**excel\_dir = APP\_ROOT+"/excel/"+test\_append+"/"+teacher\_name+"/"**

**excel\_date = request.form['fname']**

**time = request.form['ftime']**

**time = time[:2]**

**final\_send = glob(excel\_dir + "/" + excel\_date+ "@" + time +"\*.xlsx")[0]**

**print(final\_send)**

**df = pd.read\_excel(final\_send)**

**roll\_id = list(df['Roll Id'])**

**print(type(roll\_id))**

**print(roll\_id)**

**cursor = conn.cursor()**

**for i in range(len(roll\_id)):**

**cursor.execute("SELECT student\_email,parent\_email from student\_login where binary roll\_id=%s",[roll\_id[i]])**

**email = list(cursor.fetchone())**

**print(type(email[1]))**

**print(email[0])**

**print(email[1])**

**msg = Message('Auto Generated',recipients= [email[0],email[1]])**

**msg.body = "Hi.. " + roll\_id[i] + " is present for the lecture of " + "Prof. " +str(teacher\_name.split('.',1)[0]) + ", which is held on " + excel\_date + "@" + time + "hrs"**

**msg.html = "Hi.. " + roll\_id[i] + " is present for the lecture of " + "Prof. " +str(teacher\_name .split('.',1)[0])+ ", which is held on " + excel\_date + "@" + time + "hrs"**

**mail.send(msg)**

**return "<h1>mail sent<h1>"**

**Attendance Updation :**

**def update():**

**test\_append = str(request.form['excel\_folder'])**

**print(test\_append)**

**teacher\_name = str(session.get('user'))**

**print(teacher\_name)**

**excel\_dir = APP\_ROOT + "/excel/" + test\_append + "/" + teacher\_name + "/"**

**print(excel\_dir)**

**for file in request.files.getlist("updated\_excel"):**

**print(file)**

**filename = file.filename**

**print(filename)**

**destination = "/".join([excel\_dir,filename])**

**print(destination)**

**file.save(destination)**

**return render\_template("excel.html",msg="updated successfully")**

**Consolidated report generation :**

**def calculate():**

**test\_append = str(request.form['final\_class'])**

**print(test\_append)**

**teacher\_name = str(session.get('user'))**

**print(teacher\_name)**

**excel\_root = APP\_ROOT + "/excel/" + test\_append + "/" + teacher\_name + "/"**

**print(excel\_root)**

**excel\_names = os.listdir(excel\_root)**

**print(excel\_names)**

**for i in range(len(excel\_names)):**

**if excel\_names[i].startswith("."):**

**os.remove(excel\_root+excel\_names[i])**

**else:**

**if os.path.isdir(excel\_root+excel\_names[i]):**

**shutil.rmtree(excel\_root+excel\_names[i], ignore\_errors=False, onerror=None)**

**excel\_names = os.listdir(excel\_root)**

**if(excel\_names==[]):**

**return render\_template("excel.html",msg1="No excel files found")**

**for i in range(len(excel\_names)):**

**excel\_names[i] = excel\_root + excel\_names[i]**

**print(type(excel\_names))**

**# read them in**

**excels = [pd.ExcelFile(name) for name in excel\_names]**

**# turn them into dataframes**

**frames = [x.parse(x.sheet\_names[0], header=None,index\_col=None) for x in excels]**

**# delete the first row for all frames except the first**

**# i.e. remove the header row -- assumes it's the first**

**frames[1:] = [df[1:] for df in frames[1:]]**

**# concatenate them..**

**combined = pd.concat(frames)**

**if not os.path.isdir(excel\_root+"final/"):**

**os.mkdir(excel\_root + "final/")**

**final = excel\_root + "final/"**

**print(final)**

**# write it out**

**combined.to\_excel(final+"final.xlsx", header=False, index=False)**

**# below code is to find actual repetative blocks**

**workbook = pd.ExcelFile(final+"final.xlsx")**

**df = workbook.parse('Sheet1')**

**sample\_data = df['Roll Id'].tolist()**

**print (sample\_data)**

**#a dict that will store the poll results**

**results = {}**

**for response in sample\_data:**

**results[response] = results.setdefault(response, 0) + 1**

**finaldf = (pd.DataFrame(list(results.items()), columns=['Roll Id', 'Total presenty']))**

**finaldf = finaldf.sort\_values("Roll Id")**

**print (finaldf)**

**writer = pd.ExcelWriter(final+"final.xlsx")**

**finaldf.to\_excel(writer,'Sheet1',index=False)**

**workbook = writer.book**

**worksheet = writer.sheets['Sheet1']**

**worksheet.set\_column(0,1,20)**

**writer.save()**

**final = final + "final.xlsx"**

**session['final']=final**

**final = final[91:]**

**return viewfinal(final)**



# SAMPLE OUTPUT

# 

# CHAPTER 6 CONCLUSION

In this project, we have proposed a new reversible data hiding method with image contrast enhancement. A preprocessing strategy has been developed by merging the adjacent rows or columns of bins in two-dimensional histogram while the changes that have been made can be recorded. Since the constraints on choosing the bins have been relaxed in two-dimensional plane, the lowest bins are chosen in preprocessing while the highest bins can be expanded for data embedding. Thus comparable contrast enhancement effects can be obtained with satisfactory image quality. Different from the normal methods, the process of image contrast enhancement can be performed in a lossless manner with the proposed method. The experimental results have shown that an original image can be exactly recovered from a series of its contrast-enhanced versions. Moreover, the degree of contrast enhancement can be adjusted with respect to only one parameter. Compared with the methods using one-dimensional histogram, generally better performances in contrast enhancement effects and image quality have been achieved based on two-dimensional histogram modification.

# CHAPTER 7

# REFERENCES

* **[1]** Yohei Kawaguchi, Tetsuo Shoji, **Face Recognition-Based Lecture**

**Attendance System**, 3rd AERU, 2005.

* **[2]**  B. Kavinmathi, S.Hemalatha, **Attendance System for Face Recognition**

**using GSM module**, 4th International Conference on Signal Processing and

Integrated Networks, 2018.

* **[3]** Ketan N. Mahajan, Nagaraj V. Dharwadkar, **Classroom Attendance System Using Surveillance Camera**, International Conference on Computer Systems, Electronics and Control ,2017.
* **[4]**  Shubhobrata Bhattacharya, Gowtham Sandeep Nainala, Prosenjit Das,

Aurobinda Routray ,**Smart Attendance Monitoring System (SAMS): A Face**

**Recognition based Attendance System for Classroom Environment**, IEEE 18th International Conference on Advanced Learning Technologies, 2018.

* **[5]**  E.Varadharajan, R.Dharani, S.Jeevitha, **Automatic Attendance**

**Management System using Face Detection,** 2017.

* **[6]** Guo, Jing-Ming, **Complexity Reduced Face Detection using Probability-** **based Face Mask Prefiltering and pixel-based Hierarchical-feature Adaboosting, Signal Processing Letters**, IEEE 2011.
* **[7]** K.Senthamil Selvi1, P.Chitrakala, A.Antony, Jenitha S, **Face Recognition**

**based Attendance Marking System**, International Journal of Computer Science and Mobile Computing, 2014.

* **[8]** Chen, Joy Iong Zong. **Smart Security System for Suspicious Activity**

**Detection in Volatile Areas**. Journal of Information Technology 2, 2020.

* **[9]** Jacob, I. Jeena. **Capsule Network based Biometric Recognition System.** Journal of Artificial Intelligence 1,2019.
* **[10]** Kirtiraj Kadam, Manasi Jadhav, Shivam Mulay, Tushar Indalkar,

**Attendance Monitoring System Using Image Processing and Machine**

**Learning**, International Journal of Advance Engineering and Research

Development, 2017.

* **[11]**  Rajat Kumar Chauhan, Vivekanand Pandey, Lokanath M, **Smart**

**Attendance System Using CNN**, International Journal of Pure and Applied

Mathematics,2018.

* **[12]**  Mayank Yadav, Anmol Aggarwal, **Motion based Attendance System in** **Real Time Environment for Multimedia Application**, 2018.
* **[13]** Wei Wu, Chuanchang Liu, Zhiyuan Su, **Novel Real-time Face**

**Recognition from Video Streams**, International Conference on Computer

Systems, Electronics and Control, 2017.

* **[14]** Changxing Ding, Dacheng Tao, **Trunk-Branch Ensemble Convolutional Neural Networks for Video-Based Face Recognition**, IEEE transactions on pattern analysis and machine intelligence, 2018.
* **[15]** Aziza Ahmedi, Dr Suvarna Nandyal, **An Automatic Attendance System**

**Using Image processing**, The International Journal of Engineering and

Science, 2015.