Camera Application
With Recognition and Attendance Generator

A. Vishnu Vardhan, K. J. Sai Jitendra Reddy, G.Bhargavi, K MohanaChandrika, K Venkata Prasad

Abstract

Problem Statement:Using camera module to the fullest extent by providing additional features for traditional Camera Application

In the current scenario, it is possible to use a camera module in very different way by providing addition features to the application design. Considering the frequent use of web camera applications, there is a need for an enhanced and automated version of the application. Face Recognition also plays a crucial role in daily lives. An integrated solution of face detection with the camera application is a great duo and handy to the user as well. In this paper, we present a method for accessing the camera to take an image and perform image processing using pre-trained detection commands. Additionally, the model also generates the attendance user and produces excel sheet output of the recognized faces.

Keywords: Face Recognition, Attendance Report Generator, Traditional Camera features, Image Viewer

1 Introduction

Attendance maintenance is a significant function in all the institutions to monitor the performance of the students. Every institute does this in its own way. Some of these institutes use the old paper or file-based systems and some have adopted strategies of automatic attendance using some biometric techniques. A facial recognition system is a computerized software which is suited for determining or validating a person by performing comparison on patterns based on their facial appearances. Face recognition systems have upgraded appreciably in their management over the recent years and this technology is now vastly used for various objectives like security and in commercial operations. Face recognition is a powerful field of research which is a computer based digital technology. Face recognition for the intent of marking attendance is a resourceful application of attendance system. It is widely used in security systems and it can be compared with other biometrics such as fingerprint or eye iris recognition systems. As the number of students in an educational institute or employees at an organization increases, the needs for lecturers or to the organization also increase the complication of attendance control. This project may be helpful for the explanation of these types of problems. The number of students present in a lecture hall is observed, each person is identified and then the information about the number of students who are present is maintained.

The goal of the Camera system is to leverage the user's knowledge of the target documents to create custom applications rather than attempting to meet the needs of diverse users with a monolithic application. The system allows a knowledgeable user to combine image processing and detection tools in an intuitive, interactive, graphical scripting environment based on Python. Additionally, the resulting applications are suitable for a large-scale digitization project because they can be run in a batch-processing mode and easily integrated into a digitization framework.

The main objective is Computerizing the process of student record management and generating the attendance of users and producing excel sheet output of the recognized faces who are present in the classroom. In today's time every school around the world needs one or other kind of School Management System or which is called as School Management Software. Most of the schools have experienced one or other kind of system, however due to limitations of features, product experience or customer support problems they end up looking for better options.

2 Methodology

Face recognition being a biometric technique implies determination if the image of the face of any particular person matches any of the face images that are stored in a database. This difficulty is tough to resolve automatically because of the changes that several factors, like facial expression, aging and even lighting can affect the image. Facial recognition among the various biometric techniques may not be the most authentic but it has various advantages over the others. Face recognition is natural, feasible and does not require assistance. The expected system engages the face recognition approach for the automating the attendance procedure of students or employees without their involvement. A web cam is used for capturing the images of students or employees. The faces in the captured images are detected and compared with the images in database and the attendance is marked.

This project was carried out to show how a Local Binary Pattern Histogram (LBPH) face recognizer could be used for taking attendance of students. LBPH facial recognizer is a pre-trained facial recognition classifier. If enough data set are available on the face that is needed to be identified, LBPH can perform facial recognition with high accuracy. Face Recognition Student Attendance System is a desktop application that identifies and verifies student's identities with the help of a digital image. Once the recognized face matches with the stored image, the attendance is completed and marked in the database for the student. This system will provide an alternative and easier way of taking attendance.

The facial recognition system has three main phases, which are described below:

1. Face Detection

Face detection is the ability to identify the person's faces within the digital images. This system identifies the human face present in an image or video. We need to define a general structure of a face to determine certain picture or video contains a face (or several). Human faces have the same features such as eyes, nose, forehead, mouth, and chin. Therefore, the objective of face detection is to find the location and size of the face in an image. The located face is then used by the facial recognition algorithm.

2. Feature Extraction

In this phase, we are extracting the features from the detected face. In LBPH, the first local binary pattern images are computed, and a histogram is created for facial recognition. This generates a template. A template is a set of data that represents the unique and distinctive features of the detected face.

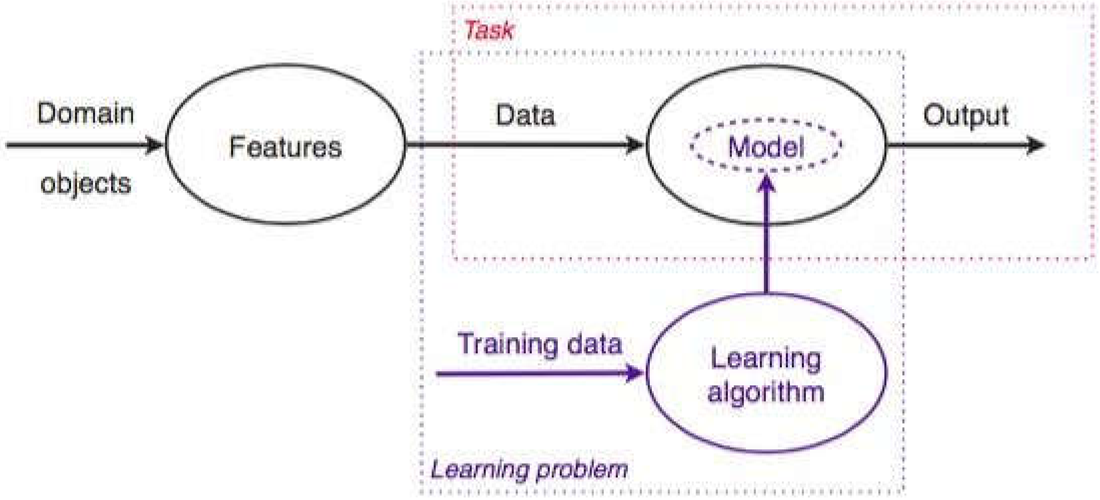
3. Face Recognition

Face Recognition is being able to uniquely identify and verify a person's face by comparing and analyzing a biometrics person's face. A face recognition system is an application that is used for identifying or verifying a person from a digital image.

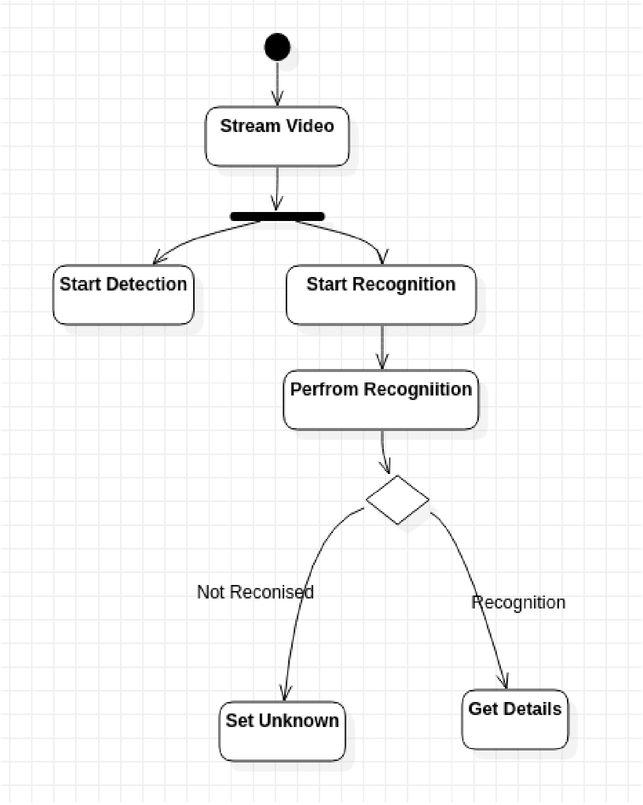
3 Concepts & Methods

The main objective of facial recognition is to identify individuals, whether individually or collectively. The number of false positives can vary, depending on the technology used for facial recognition. A facial recognition system is a technology capable of matching a human face from a digital image or a video frame against a database of faces, typically employed to authenticate users through ID verification services, works by pinpointing and measuring facial features from a given image. While humans can recognize faces without much effort, facial recognition is a challenging pattern recognition problem in computing. Facial recognition systems attempt to identify a human face, which is three-dimensional and changes in appearance with lighting and facial expression, based on its two-dimensional image.

Using camera module to the fullest extent by providing additional features for traditional Camera Applicationand providing a better abstraction for users to visualize the attendance recorded for students and also providing files to students using a web portal which is synchronous and having active listening to the database A complete face recognition system includes face detection, face preprocessing and face recognition processes. Therefore, it is necessary to extract the face region from the face detection process and separate the face from the background pattern, which provides the basis for the subsequent extraction of the face difference features. The recent rise of the face based on the depth of learning detection methods, compared to the traditional method not only shorten the time, and the accuracy is effectively improved. Face recognition of the separated faces is a process of feature extraction and contrast identification of the normalized face images in order to obtain the identity of human faces in the images.



Overview of task performance with machine learning



3.2 Activity Diagram

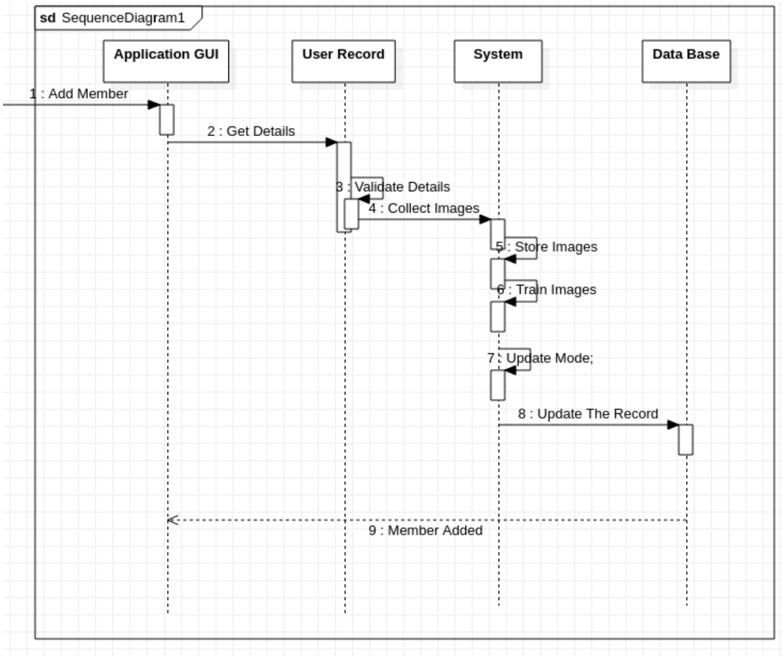


Fig 3.4 Sequence Diagram Adding Member

4 Implementation

Pseudo Code/Algorithms

Collecting Faces

```
import os
import cv2
from training import Trainer
class FaceData:
def __init__(self, name, id):
self.vid_cam = cv2.VideoCapture(0)
self.name = name
self.id = id
    face_detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
    count = 0
self.path = f"dataset/{name}/"
if not os.path.exists(path=self.path):
      os.makedirs(self.path)
while count <50:
      _, image_frame = self.vid_cam.read()
      gray = cv2.cvtColor(image_frame, cv2.COLOR_BGR2GRAY)
      faces = face_detector.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5,
minSize=(60, 60),
flags=cv2.CASCADE_SCALE_IMAGE)
for (x, y, w, h) in faces:
        cv2.rectangle(image_frame, (x, y), (x + w, y + h), (0, 255, 255), 2)
        count += 1
cv2.imwrite(self.path + self.name + "." +str(self.id) + "." + str(count) + ".jpg",
              gray[y:y + h, x:x + w])
      cv2.imshow('frame', image_frame)
if cv2.waitKey(100) &0xFF == ord('q'):
break
self.vid_cam.release()
    cv2.destroyAllWindows()
    Trainer()
```

Training

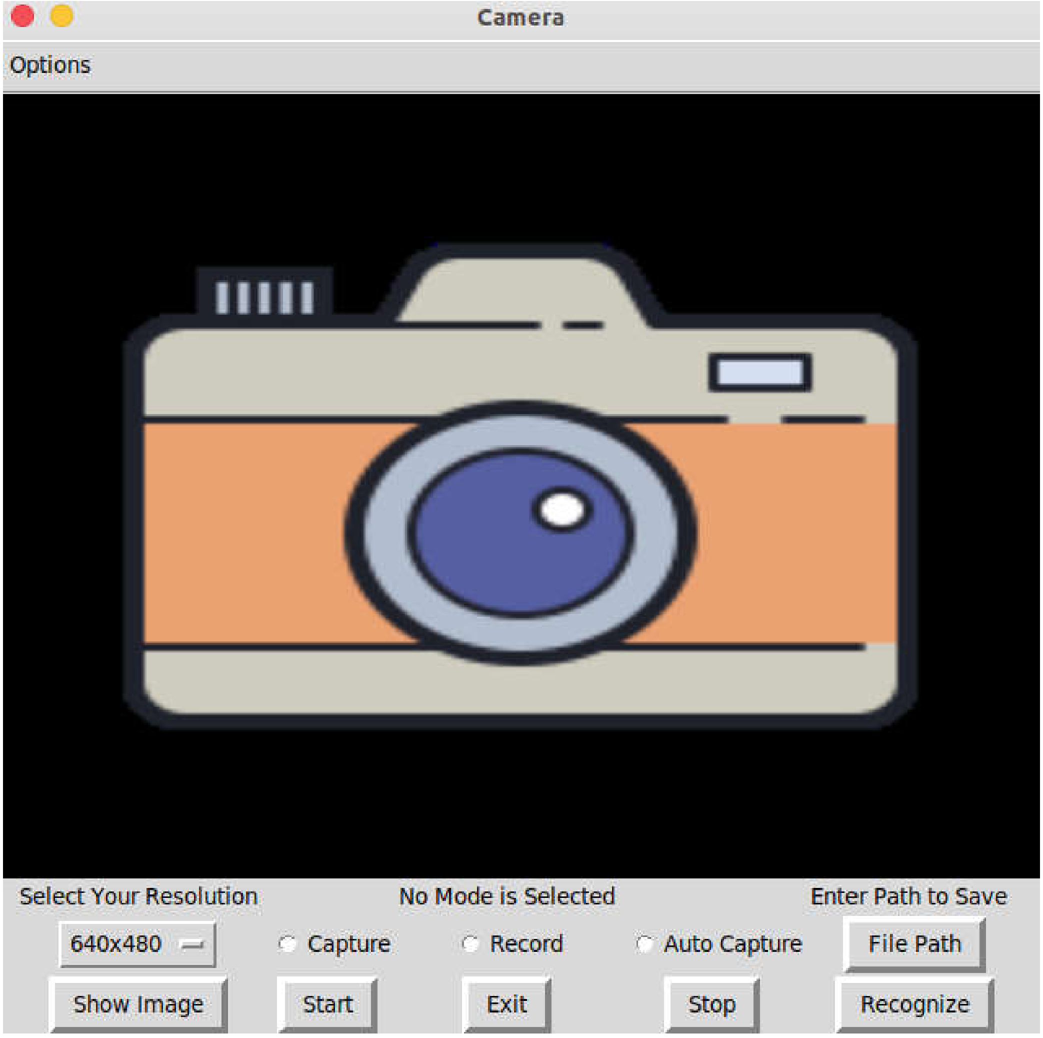
```
import os
import cv2
import numpy as np
from PIL import Image
from data_base import FaceRecognitionDataBase
class Trainer:
def __init__(self):
self.root_path = "./dataset"
self.recognizer = cv2.face.LBPHFaceRecognizer_create()
self.recognizer.read("trainer.yml")
self.detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
self.getImagesAndLabels()

def getImagesAndLabels(self):
self.sub_dir = [it.path for it in os.scandir(self.root_path) if it.is_dir()]
    face_samples = []
    ids = []
    names = []
for path in self.sub_dir:
      image_paths = [os.path.join(path, f) for f in os.listdir(path)]
print(image_paths)
for imagePath in image_paths:
        pil_img = Image.open(imagePath).convert('L')
        img_numpy = np.array(pil_img, 'uint8')
        id = int(os.path.split(imagePath)[-1].split(".")[1])
        name = os.path.split(imagePath)[-1].split(".")[0]
        names.append(name)
        faces = self.detector.detectMultiScale(img_numpy)
for (x, y, w, h) in faces:
          face_samples.append(img_numpy[y:y + h, x:x + w])
          ids.append(id)
self.recognizer.train(src=face_samples, labels=np.array(ids))
self.recognizer.write('trainer.yml')
```

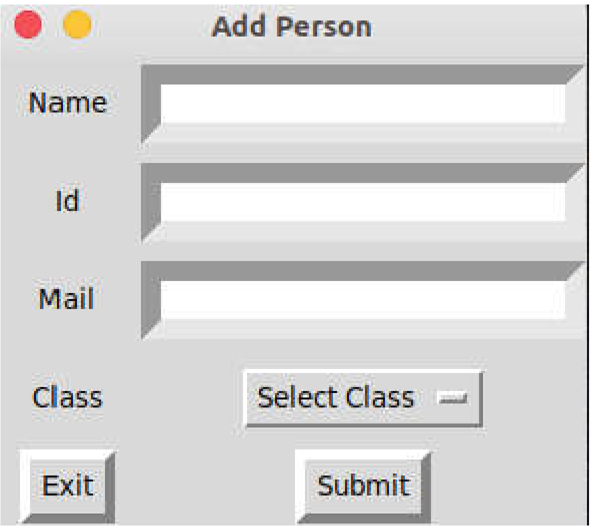
Face Recognition

```
import cv2
from data_base import FaceRecognitionDataBase
class FaceRecogniser:
def __init__(self):
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    recognizer.read('trainer.yml')
    cascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath)
    font = cv2.FONT_HERSHEY_SIMPLEX
    cam = cv2.VideoCapture(0)
    data_base = FaceRecognitionDataBase()
while True:
      ret, im = cam.read()
      gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
      faces = faceCascade.detectMultiScale( gray,scaleFactor=1.2, minNeighbors=5 )
for (x, y, w, h) in faces:
        cv2.rectangle(im, (x - 20, y - 20), (x + w + 20, y + h + 20), (0, 255, 0), 4)
        record_Id, predict = recognizer.predict(gray[y:y + h, x:x + w])
        cv2.imshow("Gray", gray[y:y + h, x:x + w])
print(record_Id, predict)
        record = data_base.get_record(record_Id)
if record is not None and predict <100:
          cv2.rectangle(im, (x - 22, y - 90), (x + w + 22, y - 22), (0, 255, 0), -1)
          cv2.putText(im, str(record.get_name()), (x, y -40), font, 2, (255, 255, 255), 3)
else:
          cv2.rectangle(im, (x - 22, y - 90), (x + w + 22, y - 22), (0, 255, 0), -1)
          cv2.putText(im, "Unknown", (x, y - 40), font, 2, (255, 255, 255), 3)
      cv2.imshow('im', im)
if cv2.waitKey(10) &0xFF == ord('q'):
break
cam.release()
    cv2.destroyAllWindows()
```

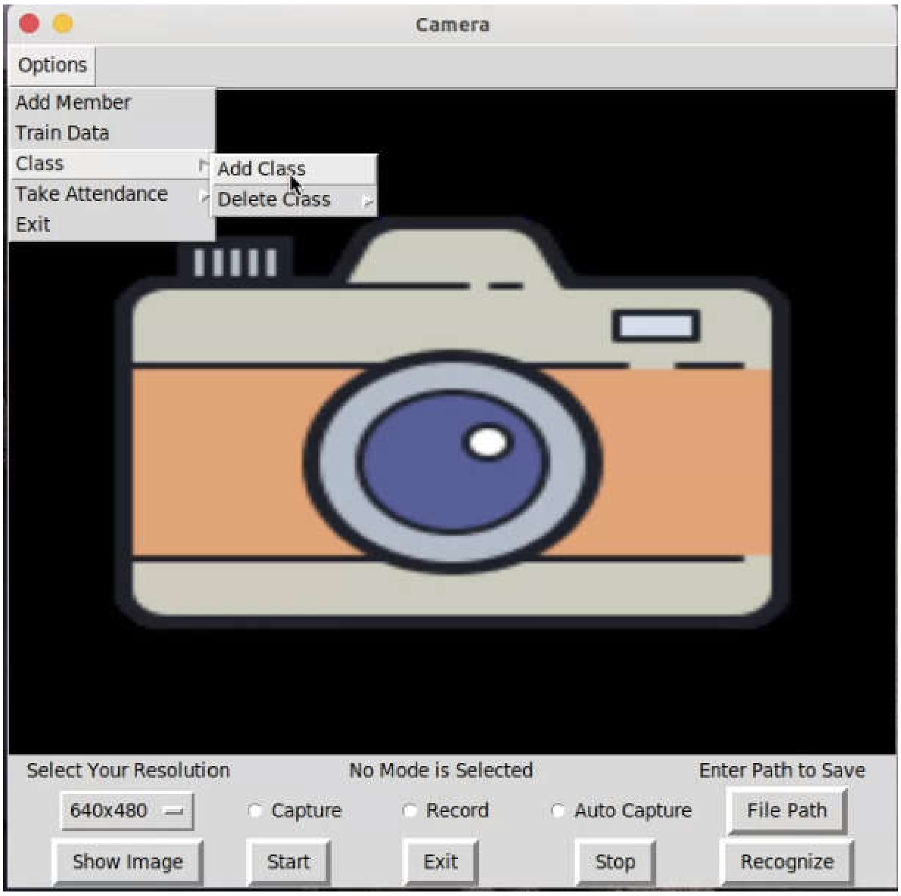
Screenshots:



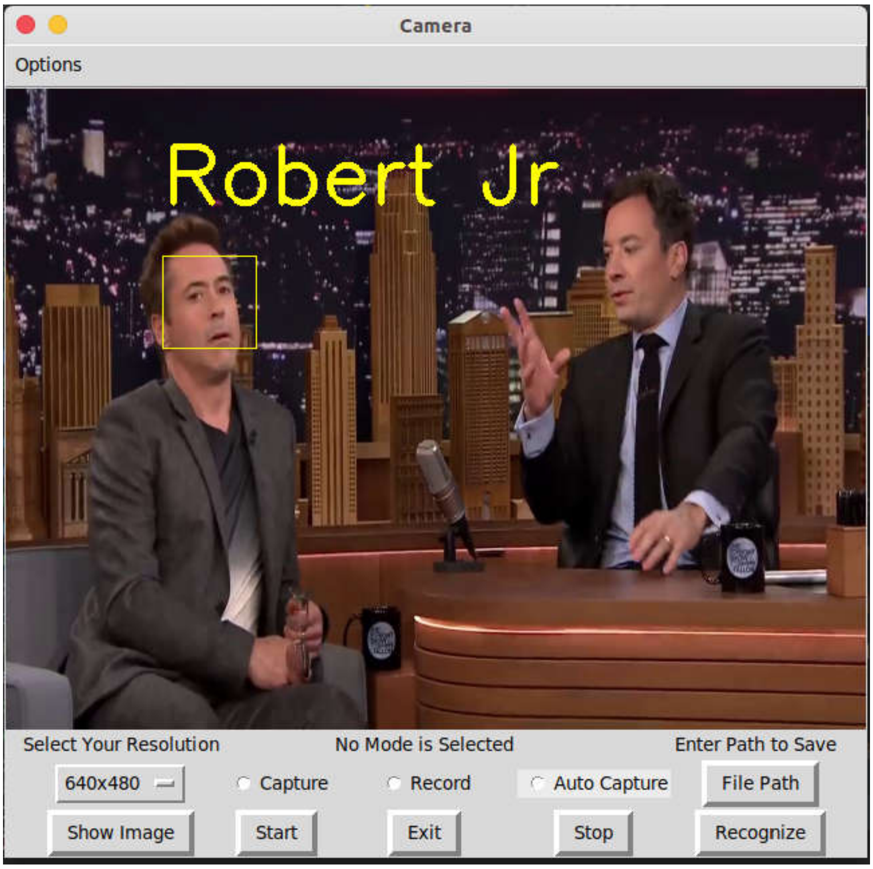
Main Window



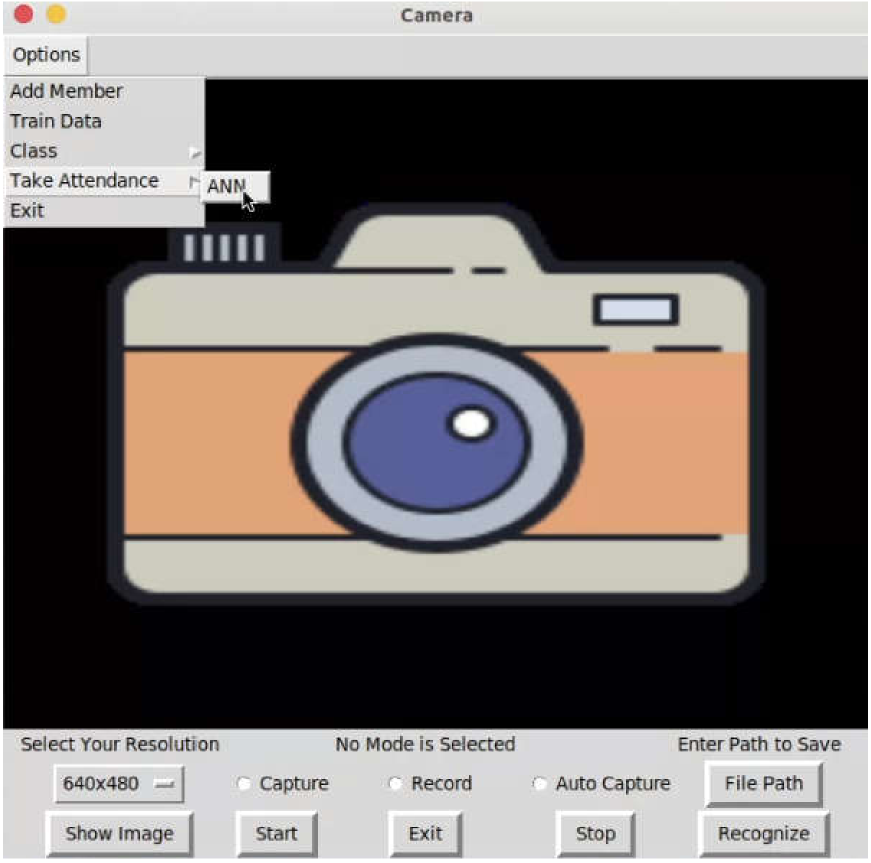
Add Person Screen



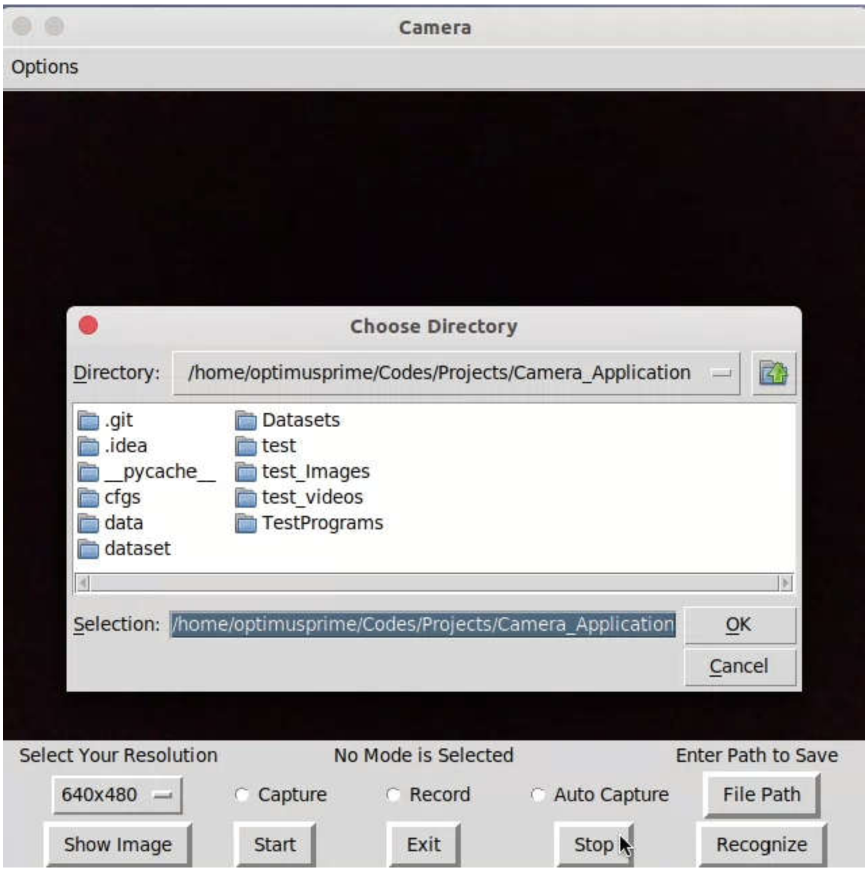
Add Class



Recognition



Take Attendance



Directory: /home/optimusprime/Codes/Projects/Camera_Application

Selection: /home/optimusprime/Codes/Projects/Camera_Application

Save Attendance

References

- Smart Attendance System using Computer Vision and Machine Learning Dipti Kumbhar, Prof. Dr. Y. S. Angal Department of Electronics and Telecommunication, BSIOTR, Wagholi, Pune, India
- ATTENDANCE SYSTEM USING MULTI-FACE RECOGNITION 1P. Visalakshi, 2Sushant Ashish 1Assistant Professor 1,2Department of Computer Science and Engineering SRM Institute of Science and Technology, Chennai, Tamil Nadu, INDIA
- Face Recognition Based Student Attendance System with OpenCV CH. VINOD KUMAR1 , DR. K. RAJA KUMAR2 1 PG Scholar, Dept of CS& SE, Andhra University, Vishakhapatnam, AP, India. 2Assistant Professor, Dept of CS& SE, Andhra University, Vishakhapatnam, AP, India.
- Automatic Attendance System Using Face Recognition. Ashish Choudhary1, Abhishek Tripathi2, Abhishek Bajaj3,Mudit Rathi4 and B.M Nandini5 1,2,3,4,5 Information Science and Engineering, The National Institute of Engineering,
- Face Recognition based Attendance Management System using Machine Learning Anushka Waingankar1, Akash Upadhyay2, Ruchi Shah3, Nevil Pooniwala4, Prashant Kasambe5
- https://www.superdatascience.com/blogs/opencv-face-recognition
- https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b
- https://www.pyimagesearch.com/2018/09/24/opencv-face-recognition/
- http://nxglabs.in/cloud/impact-biometric-attendance-system-educational-institutes.html
- https://iopscience.iop.org/article/10.1088/1757-899X/263/4/042095/pdf
- http://www.ijsrp.org/research-paper-0218/ijsrp-p7433.pdf
- https://www.theseus.fi/bitstream/handle/10024/132808/Delbiaggio_Nicolas.pdf?sequence=1&isAllowed=y