VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF ELECTRONICS ENGINEERING

Department of Embedded Technology

Face recognition based attendance system using Raspberry pi.

18BEC0815 - Varanasi Ravi Kiran
18BEC0767 - Byreddy Aneesh Reddy
18BEC0540 - Lepuru Chandan
18BEC0591 - Rohith Kothuri

Technical Answers for Real World Problems(TARP)

Winter Sem 2020-21

Submitted to
Faculty: Prof. Dr. S.HARIHARAN **Signature:**
Date:
Slot : TG1

Abstract: Student participation in the classroom is very much important for better learning and increased success rates. A major issue in the classroom is the time consuming attendance process which leads to a lack of interest in the lecture for both teacher and the students. Many teacher prefer to take attendance by calling out the names of the students which is both tedious and prone to proxy. An effective solution to this manual system is very much required.

Providing accurate attendance management systems for real-time application is challenging. It is very challenging and time consuming process for a teacher to mark the attendance of a whole class. In the recent years many facial recognition based attendance systems have been proposed but they all have their own abilities and drawbacks. Many alternatives to manual attendance have been already proposed which include Biometric , Iris and face recognition based attendance systems. Many researchers have used algorithms like Principle component analysis and eigen faces but none of them commented on developing an application layer for the end user.

This project aims to design a robust face recognition based attendance system using deep learning. The major steps in this project are face detection , face recognition and design of android application for the end user. Face detection is done by employing facial feature extraction and mapping. This project aims to provide an attendance solution for schools , educational institutes and hospitals.

1.Introduction:

Face recognition is an effective system for attendance management . This system can be easily be deployed in schools , colleges and educational institutes. The existing manual attendance is a tedious task and requires human effort all these difficulties can be replaced by this facial recognition attendance system.

The whole project can be divided into three major steps which include face detection , face recognition and android application. Face detection can be achieved using a neural network. For facial recognition each face is mapped to a 128 byte array of facial features . A local database of the students and their facial features is required to detect and identify their faces. When a student

comes in front of the system the camera takes a photograph and identifies the face in it followed by the generation of 128 byte facial features array, this array data is the matched with the existing students data in the local data and the name is retrieved .The attendance status of the person is then send to the cloud and a mail is sent to the student regarding his attendance for that day.

The student can also monitor his attendance record in the android application. The enrolment of new students is also possible. Whenever a new student is enrolled his name and email address are stored in the cloud along with his photo in the local database. An email regarding his credentials for the application is also sent.

The proposed system is developed using Raspberry pi , pi camera. For the cloud storage Google sheets are used and an android application for the end user is developed using MIT app inventor.

The next section focuses on the existing models for face recognition and their Literature survey.

2 . Literature survey:

S.No	Authors	Journal	Proposed method
1.	Arun katara, Sudesh, V Kolhe.	Attendance System Using Face Recognition and Class Monitoring System, Arun Katara1, Mr. Sudesh2, V.Kolhe3http://www.ijritcc.org/download/browse/Volume_ 5_Issues/February_17_Volume_5_Issue_2/ 1489565866_ 15- 03-2017.pd	Using PCA with MATLAB to implement face recognition
2.	E.Rekha P.Ramaprasad	"An efficient automated attendance managementsystembasedonEigenFacerecognition,"2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence, vol. 5. IEEE, jan 2017, pp. 605–608	Using PCA and eigenface. The pre-processing cropping the region of interest that make the accuracy more higher. The computation takes time and it is very costly to implement.

3.	Wagh, Priyanka; Thakare, Roshani; Chaudhari, Jagruti; Patil, Shweta	"Attendance system based on face recognition using eigen face and PCA algorithms", International Conference on Green Computing and Internet of Things (ICGCIoT). IEEE, oct 2015, pp. 303–308.	Using PCA and Histogram based noise reduction techniques. Providing a better accuracy by implementation Ada-Boost algorithm on Face Detection. The database is update to date with the new images. Skin classification use for improving the accuracy of face detection
4.	Visar Shehu, A Dika	"Using real time computer vision algorithms in automatic attendance management systems,"Information Technology Interfaces (ITI), 2010 32nd International Conference on, pp. 397–402, 2010.	Using HAAR Classifier and computer vision algorithm to implement face recognition. The system integrated with the existing system that can use the existing feature from Learning Management System
5.	AalamGumber, NavneetKaur	"Face Recognition Based Automated Attendance Management System using Principal Component Analysis" , International Journal of Science and Research (IJSR) , Vol 4, Issue 6, June 2015.	Using PCA to train and reduce dimensionality and ANN to classify input data and find the pattern. The accuracy is high due to the combination PCA with ANN. ANN use for classificaton is more accuracy than PCA with Eigenface
6.	N Kar, MK Debbarma, A Saha	"Study of Implementing Automated Attendance System Using Face Recognition Technique," International Journal of Computer and Communication Engineering, vol. 1, no. 2, pp. 100–103, 2012	Using Eigenvector and Eigenvalue for face recognition The system have prevented the fake attendance due to the implementation clock time in and time, which used for checking the

			student presence inside the class for the period or not

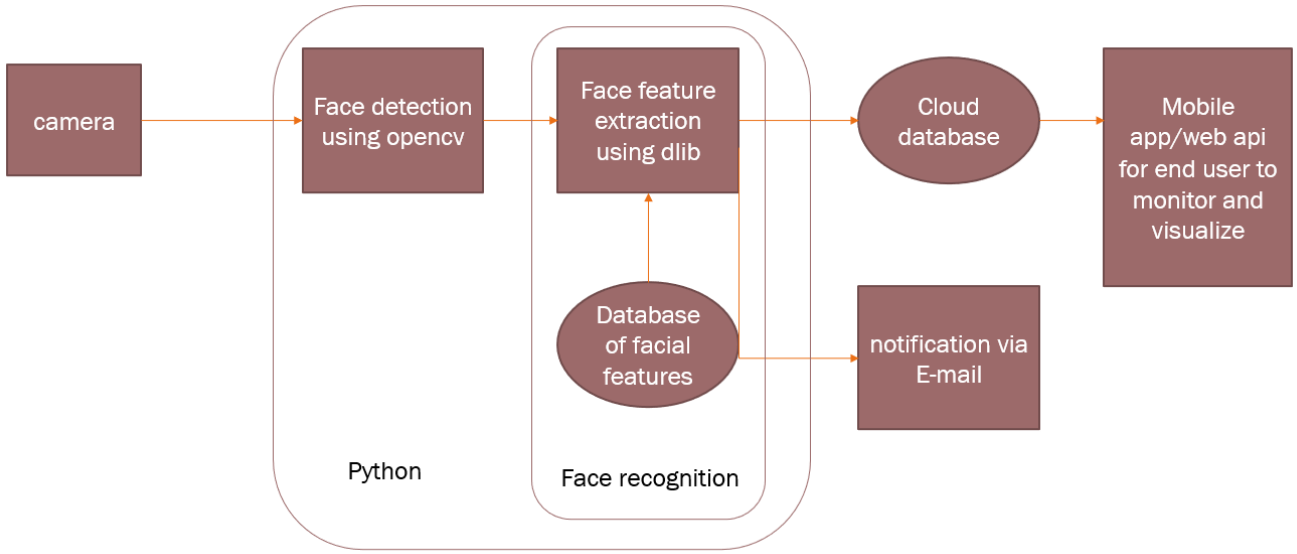
3 . Drawback in existing model:

The existing models does not provide an application layer for the end user and the system is prone to loss of data. The student does not have any real-time information regarding his attendance status and his attendance record is also not known.

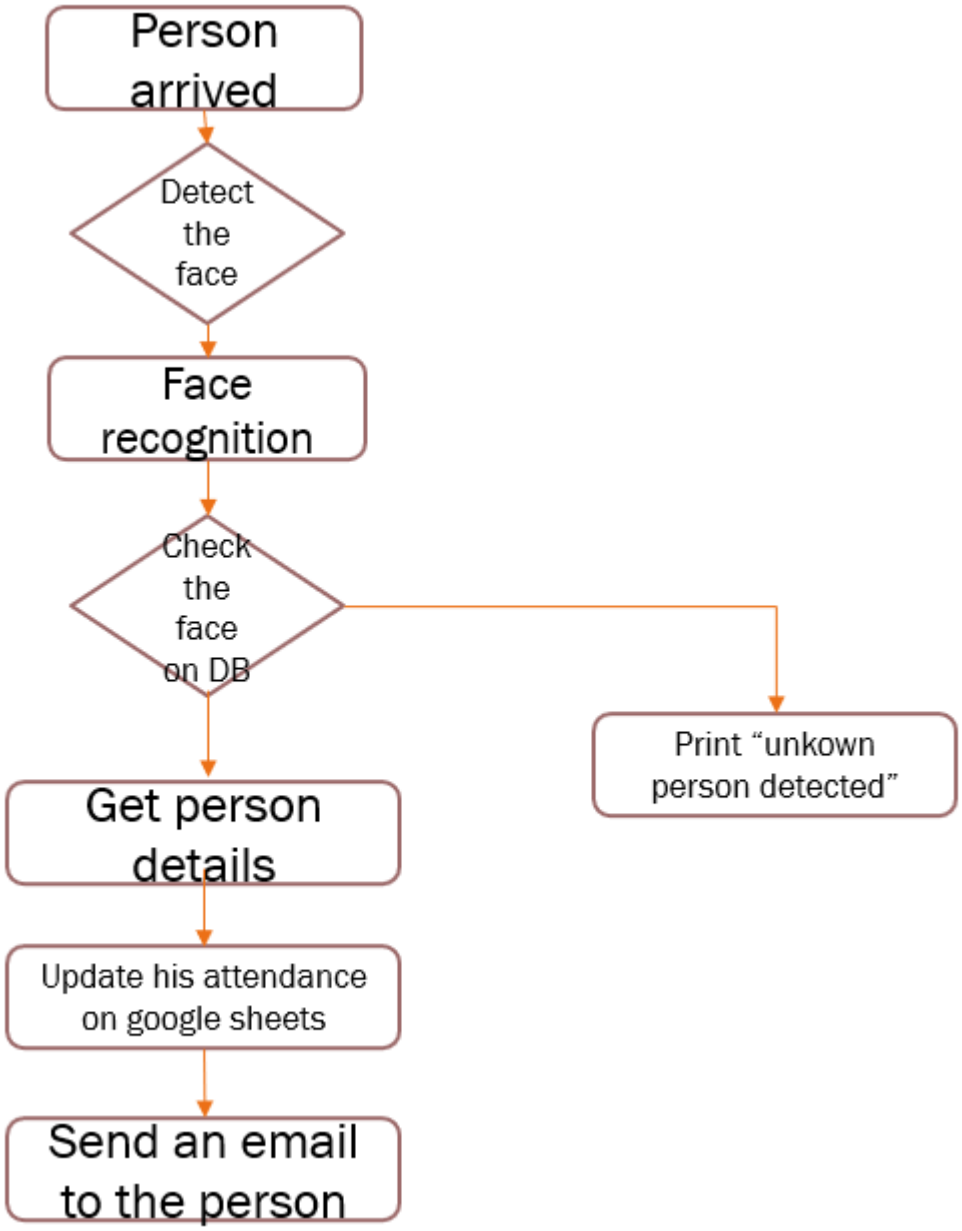
4 . Proposed work:

The proposed model provides real-time acknowledgement of the attendance status to the student by an email. A cloud database also gets the attendance status so that the admin can manage the data. An android application is developed for the end user whereby he can check his attendance data.

5. a. Block diagram:



5.b.Flow chart:



6.Implementation:

6.a.Face detection and recognition:

The face recognition is done by facial feature extraction and matching. Photographs and the facial features of the faces that are to detect are prestored in the local database. Whenever new faces are detected a 128 byte array of their facial features is created , this data is the matched with the existing database and the name of the person is displayed. The matching process is done by calculating the Euclidian distance between the required data and the existing data and the minimum of the Euclidian distance is taken. The minimum distance is compared with the predefined threshold if it is less that the threshold the name of the person is returned.

6.b.Real-time acknowledgement with email:

The attendance status is updated in the google sheets. The email address corresponding to the student is fetched from the server and an email regarding the attendance status along with the date is sent to the user.

6.c.Application layer:

An android application is developed that can show the attendance record of a particular student. The app uses a login page for the authentication of the user, once the user logs in it fetches the data from the google sheet and displays it in a list view along with the total number of lectures attended and attendance percentage.

7.a.Software results:

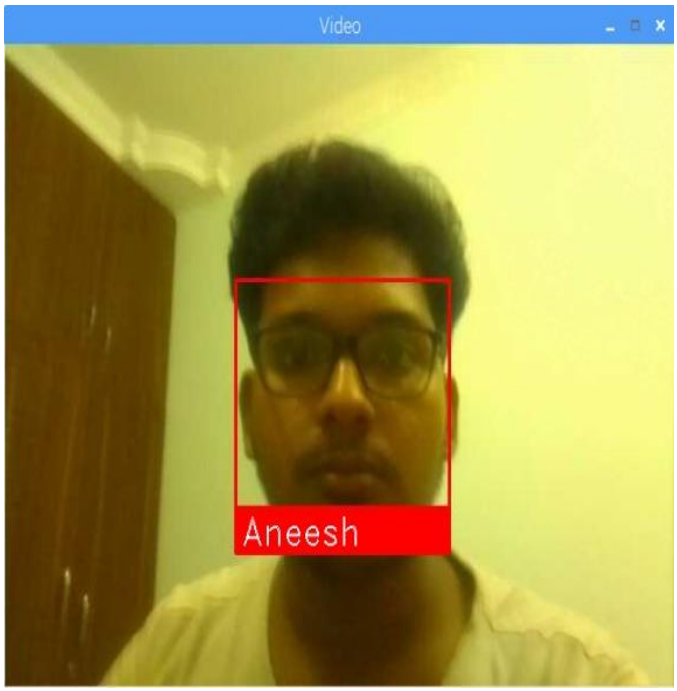


Fig3: Detected image with a bounding box around the face.

attendance

File Edit View Insert Format Data Tools Add-ons Help Last edit was seconds ago

H3 absent

	A	B	C	D	E	F	G	H	I
1	name	email	app OTP	6/2/2021	6/3/2021	6/4/2021	6/5/2021	6/6/2021	
2	Aneesh	aneeshbyreddy44@gmail.com	9900	present	late	absent	present	late	
3	Ravi kiran	abcd@gmail.com	8235	absent	present	late	present	absent	
4	Chandan	efgh@gmail.com	9242	late	present	absent	present	present	
5	rohit	wxyz@gmail.com	8226	present	absent	late	present	present	
6									
7									
8									

Fig4: Google sheet as cloud database.

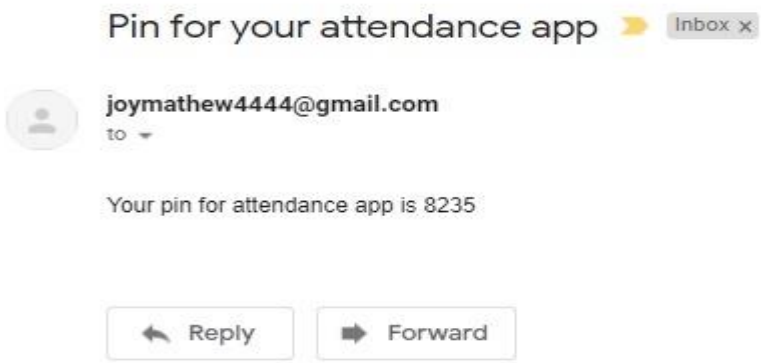


Fig5: Mail regarding the pin for newly enrolled student.

Fig6: Mail regarding the attendance status.

8.b.Results from android app:

Fig7: Login page.

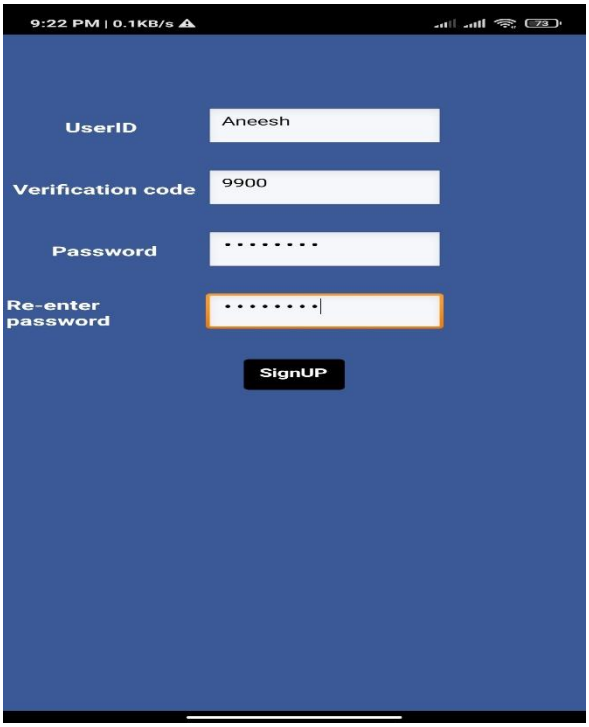


Fig8: Signup page.

Fig9: Attendance page.

9 . Conclusion: A robust facial recognition based attendance system is made for deployment in schools , colleges and educational institutes. The system includes complete hardware development in the physical layer , cloud analytics in the google sheets and an android app in the application layer. A complete system is proposed along with its merits and demerits.

10. References:

[1] Attendance System Using Face Recognition and Class Monitoring System, Arun Katara1, Mr. Sudesh2, V.Kolhe3http://www.ijritcc.org/download/browse/Volume_ 5_Issues/February_17_Volume_5_Issue_2/1489565866_ 15- 03-2017.pd.

[2] "An efficient automated attendance managementsystembasedonEigenFacerecognition,"2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence, vol. 5. IEEE, jan 2017, pp. 605–608.

[3] "Attendance system based on face recognition using eigen face and PCA algorithms", International Conference on Green Computing and Internet of Things (ICGCIoT). IEEE, oct 2015, pp. 303–308.

[4] "Face Recognition Based Automated Attendance Management System using Principal Component Analysis" , International Journal of Science and Research (IJSR) , Vol 4, Issue 6, June 2015.

[5] "Study of Implementing Automated Attendance System Using Face Recognition Technique," International Journal of Computer and Communication Engineering, vol. 1, no. 2, pp. 100–103, 2012.

[6] "Using real time computer vision algorithms in automatic attendance management systems,"Information Technology Interfaces (ITI), 2010 32nd International Conference on, pp. 397–402, 2010.

[7] "OpenCV tutorial" available at "https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_tutorials.html"

[8] "Write data to a google sheet using python" available at "https://gspread.readthedocs.io/en/latest/user-guide.html#updating-cells"

[9] "Understanding pickle library" available at "https://www.geeksforgeeks.org/understanding-python-pickling-example"

[10] “Data analytics using python and google sheet” available at “https://medium.com/analytics-vidhya/how-to-read-and-write-data-to-google-spreadsheet-using-python-ebf54d51a72c”

11. Appendix:

Enroll:

```
import face_recognition,cv2,pickle
import spreadsheet

photo_folder = '/home/pi/Desktop/Face-recognition-based-attendance-system-master/known face photos/'
facial_encodings_folder='/home/pi/Desktop/Face-recognition-based-attendance-system-master/known face encodings/'

cap = cv2.VideoCapture(0)

def encoding_of_enrolled_person(name,image):
    enroll_encoding=[]

    enroll_encoding.append(face_recognition.face_encodings(face_recognition.load_image_file(image))[0])
    f=open(facial_encodings_folder+name+'.txt','w+')

    with open(facial_encodings_folder+name+'.txt','wb') as fp:
        pickle.dump(enroll_encoding,fp)
    f.close



def enroll_via_camera():
    name=input('Enter your name: ')
    email=input('Enter email address: ')
    while True:
        ret,frame=cap.read()
        cv2.imshow('Enrolling new attendee',frame)
        k=cv2.waitKey(1)
        if k & 0xFF==ord('y'):
            cv2.imwrite(photo_folder+name+'.jpg',frame)
            encoding_of_enrolled_person(name,photo_folder+name+'.jpg')
            cv2.destroyAllWindows()
            break
        if k& 0xFF==ord('q'):
            print('quitting')
            cv2.destroyAllWindows()
            break
    cap.release()
    #email=input("Enter email address: ")
    spreadsheet.enroll_person_to_sheet(name,email)
```

Spreadsheet:

```
import datetime,gspread,random
```

```
from oauth2client.service_account import ServiceAccountCredentials
import emailing as em

scope =
["https://spreadsheets.google.com/feeds",'https://www.googleapis.com/auth/spreadsheets',"https://www.googleap
is.com/auth/drive.file","https://www.googleapis.com/auth/drive"]

creds = ServiceAccountCredentials.from_json_keyfile_name("credentials.json", scope)

client = gspread.authorize(creds)

sheet = client.open("attendance").sheet1

max_intime='22:00:00'

def enroll_person_to_sheet(name,email):
    nrows = len(sheet.col_values(1))
    pin=random.randint(999,9999)
    sheet.update_cell(nrows+1,1,name)
    sheet.update_cell(nrows+1,2,email)
    sheet.update_cell(nrows+1,3,pin)
    em.email_pin(email,pin)


def mark_all_absent():
    now=datetime.datetime.now()
    date=now.strftime('%m/%d/%Y').replace('/0','/')
    if(date[0]=='0'):
        date=date[1:]
    datecell=sheet.find(date)
    nrows = len(sheet.col_values(1))
    for row in range(2,nrows+1):
        sheet.update_cell(row,datecell.col,'absent')
```

Emailing:

```
import smtplib, ssl,datetime

sender= "joymathew4444@gmail.com"     #senders email id
password="joy@4444"                #password



def email_pin(email,pin):
    port = 465
    now=datetime.datetime.now()
    date=now.strftime('%m/%d/%Y').replace('/0','/')
    if(date[0]=='0'):
        date=date[1:]
    subject="Pin for your attendance app "
    text="\nYour pin for attendance app is "+str(pin)
```

```
    message ='Subject: {}\n\n{}'.format(subject, text)


    context = ssl.create_default_context()

    print("Starting to send")
    with smtplib.SMTP_SSL("smtp.gmail.com", port, context=context) as server:
        server.login(sender, password)
        server.sendmail(sender, email, message)

    print("sent email!")



def send_email(receiver_mail,attendance):
    port = 465
    now=datetime.datetime.now()
    date=now.strftime('%m/%d/%Y').replace('/0','/')
    subject="Attendance on "+str(date)
    text="\nYour attendance is marked "+attendance

    message ='Subject: {}\n\n{}'.format(subject, text)


    context = ssl.create_default_context()

    print("Starting to send")
    with smtplib.SMTP_SSL("smtp.gmail.com", port, context=context) as server:
        server.login(sender, password)
        server.sendmail(sender, receiver_mail , message)

    print("sent email!")
```

Recognition:

```
import face_recognition,cv2,os,pickle
import numpy as np
import spreadsheet

known_face_encodings=[]
known_face_names=[]
cap = cv2.VideoCapture(0,cv2.CAP_DSHOW)

photo_folder = '/home/pi/Desktop/Face-recognition-based-attendance-system-master/known face photos/'
facial_encodings_folder='/home/pi/Desktop/Face-recognition-based-attendance-system-master/known face
encodings/'

def load_facial_encodings_and_names_from_memory():
    for filename in os.listdir(facial_encodings_folder):
        known_face_names.append(filename[:-4])
        with open (facial_encodings_folder+filename, 'rb') as fp:
```

```
            known_face_encodings.append(pickle.load(fp)[0])


def run_recognition():


    video_capture = cv2.VideoCapture(0)


    face_locations = []
    face_encodings = []
    face_names = []
    process_this_frame = True

    while True:
    # Grab a single frame of video
        ret, frame = video_capture.read()

    # Resize frame of video to 1/4 size for faster face recognition processing
        small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)

    # Convert the image from BGR color (which OpenCV uses) to RGB color (which face_recognition uses)
        rgb_small_frame = small_frame[:, :, ::-1]

    # Only process every other frame of video to save time
        if process_this_frame:
       # Find all the faces and face encodings in the current frame of video
            face_locations = face_recognition.face_locations(rgb_small_frame)
            face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)

            face_names = []
            for face_encoding in face_encodings:
           # See if the face is a match for the known face(s)
                matches = face_recognition.compare_faces(known_face_encodings, face_encoding)
                name = "Unknown"

           # # If a match was found in known_face_encodings, just use the first one.
           # if True in matches:
           #     first_match_index = matches.index(True)
           #     name = known_face_names[first_match_index]

           # Or instead, use the known face with the smallest distance to the new face
                face_distances = face_recognition.face_distance(known_face_encodings, face_encoding)
                best_match_index = np.argmin(face_distances)
                if matches[best_match_index]:
                    name = known_face_names[best_match_index]

                face_names.append(name)

        process_this_frame = not process_this_frame
```

```
    # Display the results
        for (top, right, bottom, left), name in zip(face_locations, face_names):
        # Scale back up face locations since the frame we detected in was scaled to 1/4 size
            top *= 4
            right *= 4
            bottom *= 4
            left *= 4

        # Draw a box around the face
            cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)

        # Draw a label with a name below the face
            cv2.rectangle(frame, (left, bottom - 35), (right, bottom), (0, 0, 255), cv2.FILLED)
            font = cv2.FONT_HERSHEY_DUPLEX
            cv2.putText(frame, name, (left + 6, bottom - 6), font, 1.0, (255, 255, 255), 1)

    # Display the resulting image
        cv2.imshow('Video', frame)
        flag=-1
        if(len(face_names)!=0):
          count=0
          for person in face_names:
              if(person=='Unknown'):
                  count+=1
          if(count==len(face_names)):
              flag=1
          else:
              flag=0


    # Hit 'q' on the keyboard to quit!


        if cv2.waitKey(1) & 0xFF==ord('q') or flag==0:
          spreadsheet.write_to_sheet(face_names[0])
          break



# Release handle to the webcam
    video_capture.release()
    cv2.destroyAllWindows()

spreadsheet.mark_all_absent()

load_facial_encodings_and_names_from_memory()

while True:
    run_recognition();
```

.Plagiarism report:

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	1039	Date	October 26,2020
Characters	6526	Exclude URL	

2%	98%	1	52
Plagiarism	Unique	Plagiarized Sentences	Unique Sentences

Content Checked For Plagiarism

Abstract: Student participation in the classroom is very much important for better learning and increased success rates. A major issue in the classroom is the time-consuming attendance process which leads to a lack of interest in the lecture for both teacher and the students. Many teachers prefer to take attendance by calling out the names of the students which is both tedious and prone to proxy. An effective solution to this manual system is very much required.

Providing accurate attendance management systems for real-time application is challenging. It is very challenging and time consuming process for a teacher to mark the attendance of a whole class. In the recent years many facial recognition based attendance systems have been proposed but they all have their own abilities and drawbacks. Many alternatives to manual attendance have been already proposed which include Biometric , Iris and face recognition based attendance systems. Many researchers have used algorithms like Principle component analysis and eigen faces but none of them commented on developing an application layer for the end user.
This project aims to design a robust face recognition based attendance system using deep learning. The major steps in this project are face detection , face recognition and design of android application for the end user. Face detection is done by employing facial feature extraction and mapping. This project aims to provide an attendance solution for schools , educational institutes and hospitals.

Introduction: Face recognition is an effective system for attendance management . This system can be easily be deployed in schools , colleges and educational institutes. The existing manual attendance is a tedious task and requires human effort all these difficulties can be replaced by this facial recognition attendance system.
The whole project can be divided into three major steps which include face detection , face recognition and android application. Face detection can be achieved using a neural network.
For facial recognition each face is mapped to a 128 byte array of facial features . A local database of the students and their facial features is required to detect and identify their faces. When a student comes in front of the system the camera takes a photograph and identifies the face in it followed by the generation of 128 byte facial features array, this array data is the matched with the existing students data in the local data and the name is retrieved . The attendance status of the person is then send to the cloud and a mail is sent to the student regarding his attendance for that day.
The student can also monitor his attendance record in the android application. The enrolment of new students is also possible. Whenever a new student is enrolled his name and email address are stored in the cloud along with his photo in the local database. An email regarding his credentials for the application is also sent.
The proposed system is developed using Raspberry pi, pi camera and a 16x2 LCD. The LCD shows the attendance status in real time. For the cloud storage Google sheets are used and an android application for the end user developed using MIT app inventor.