FACIAL RECOGNITION BASED ATTENDANCE (AI OR ML)

ABSTRACT

[[](https://www.google.com/search?q=What+is+facial+recognition+attendance+system?&sxsrf=AOaemvKxBelFBEPnZ3YpD_kWwZLOFdE5Mw:1640694848008&tbm=isch&source=iu&ictx=1&fir=mCjm4hiCNiZRbM%2CFcxjinIKHN8ZkM%2C_&vet=1&usg=AI4_-kT5o7R8z7gXcuXw-y4M9tHJQhNvvg&sa=X&ved=2ahUKEwjzl-bMwIb1AhUfxzgGHUTSAaMQ9QF6BAgUEAE#imgrc=mCjm4hiCNiZRbM)](https://www.google.com/search?q=What+is+facial+recognition+attendance+system?&sxsrf=AOaemvKxBelFBEPnZ3YpD_kWwZLOFdE5Mw:1640694848008&tbm=isch&source=iu&ictx=1&fir=mCjm4hiCNiZRbM%252CFcxjinIKHN8ZkM%252C_&vet=1&usg=AI4_-kT5o7R8z7gXcuXw-y4M9tHJQhNvvg&sa=X&ved=2ahUKEwjzl-bMwIb1AhUfxzgGHUTSAaMQ9QF6BAgUEAE" \l "imgrc=mCjm4hiCNiZRbM)

The face is one of the easiest ways to distinguish the individual identity of each other. Face recognition is a personal identification system that uses personal characteristics of a person to identify the person's identity. Human face recognition procedure basically consists of two phases, namely face detection, where this process takes place very rapidly in humans, except under conditions where the object is located at a short distance away, the next is the introduction, which recognize a face as individuals. Stage is then replicated and developed as a model for facial image recognition (face recognition) is one of the much-studied biometrics technology and developed by experts.

A facial recognition attendance system **uses facial recognition technology to identify and verify a person using the person's facial features and automatically mark attendance**. The software can be used for different groups of people such as employees, students, etc. The system records and stores the data in real-time.

With a face recognition attendance system, **the entire environment is automated**. You won't just take the attendance but also automatically record the entry-exit time of the student. It also adds to the security of the workplace as the system can recognize who left the designated area and when accurately. **Face detection** is a broader term than face recognition. Face detection just means that a system is able to identify that there is a human face present in an image or video. ... Face recognition can confirm identity. It is therefore used to control access to sensitive areas.

Features

* Watchlist-as-a-Service. All facial recognition systems begin with assembling a database of students
* An Airtight Matching Algorithm. ...
* Scalability. ...
* Built-in Privacy Protection. ...
* Predictive Analytics.

**Existing System**

In the existing system every teacher need call out the names of every student and mark the attendance,then take out all the statistics ,which is a tedious task for every teacher

So, it is very difficult for a recognition system to identify them. These problems can be due system faults used in face recognition, such as **camera distortion, background noise, inefficient storage, improper techniques** etc.

More than that there can be network problems due to environmental conditions.

Proposed System

The **Student Attendance Management System Project In Python** was developed using **Python OpenCV** in **Real-Time**, This **Attendance Management System Project In Python** provide a valuable attendance service for both teachers and students. Reduce manual process errors by provide automated and a reliable attendance system uses face recognition technology.

A **Attendance Management System Using Face Recognition Python** is a simple python script that recognizes faces and mark attendance for the recognized faces in an excel sheet. We seek to provide a valuable attendance service for both teachers and students. Reduce manual process errors by provide automated and a reliable attendance system uses face recognition technology.

**Face Recognition Algorithms**  used

* **Haar Cascade**

Haar Cascade is a face detection algorithm that is used to make out faces in images and real-time videos

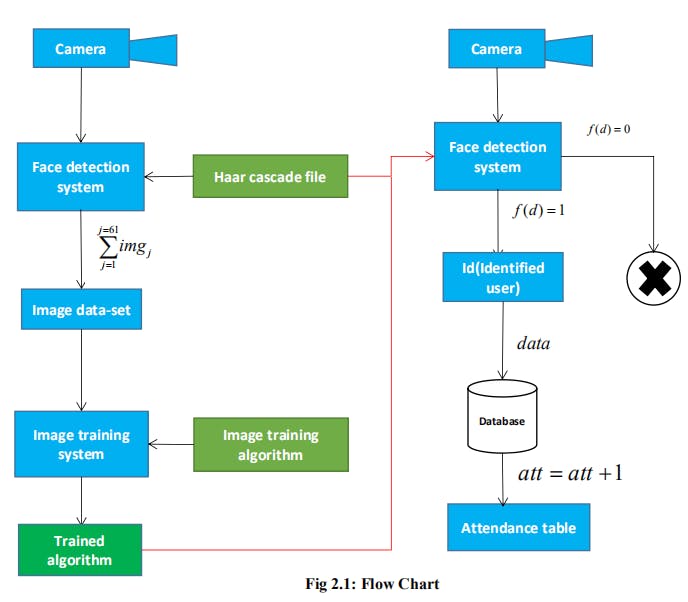
* **LBPH (Local Binary Pattern Histogram)**

Local Binary Pattern Histogram (LBPH) is used to recognize a person's face

* **Hardware Requirement: -**
* i3 Processor Based Computer
* 4GB-Ram
* 250 GB Hard Disk
* Internet Connection
* **Software Requirement:**
* Windows 8 or higher
* PYTHON INTERPRETER 3.6 ONWARDS
* PYCHARM OR PYTHON IDLE
* EXCEL
* **OpenCV**, **Pillow**, **Pandas**, **Shutil**, **CSV** and **Numpy** are the Python packages that are necessary for this project

MODULES

* CHECK CAMERA This module is used for checking the camera and its functionality
* CAPTURE FACES This module used for capturing and storing faces of all the students
* TRAIN IMAGES This module is used to train images and recognize images
* RECOGNIZE AND ATTENDANCE This module is used to recognize and take attendance



Source code

Mian.py

import os # accessing the os functions

import check\_camera

import Capture\_Image

import Train\_Image

import Recognize

# creating the title bar function

def title\_bar():

os.system('cls') # for windows

# title of the program

print("\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("\t\*\*\*\*\* Face Recognition Attendance System \*\*\*\*\*")

print("\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

# creating the user main menu function

def mainMenu():

title\_bar()

print()

print(10 \* "\*", "WELCOME MENU", 10 \* "\*")

print("[1] Check Camera")

print("[2] Capture Faces")

print("[3] Train Images")

print("[4] Recognize & Attendance")

# print("[5] Auto Mail")

print("[6] Quit")

while True:

try:

choice = int(input("Enter Choice: "))

if choice == 1:

checkCamera()

break

elif choice == 2:

CaptureFaces()

break

elif choice == 3:

Trainimages()

break

elif choice == 4:

RecognizeFaces()

break

elif choice == 5:

os.system("py automail.py")

break

mainMenu()

elif choice == 6:

print("Thank You")

break

else:

print("Invalid Choice. Enter 1-4")

mainMenu()

except ValueError:

print("Invalid Choice. Enter 1-4\n Try Again")

exit

# ---------------------------------------------------------

# calling the camera test function from check camera.py file

def checkCamera():

check\_camera.camer()

key = input("Enter any key to return main menu")

mainMenu()

# --------------------------------------------------------------

# calling the take image function form capture image.py file

def CaptureFaces():

Capture\_Image.takeImages()

key = input("Enter any key to return main menu")

mainMenu()

# -----------------------------------------------------------------

# calling the train images from train\_images.py file

def Trainimages():

Train\_Image.TrainImages()

key = input("Enter any key to return main menu")

mainMenu()

# --------------------------------------------------------------------

# calling the recognize\_attendance from recognize.py file

def RecognizeFaces():

Recognize.recognize\_attendence()

key = input("Enter any key to return main menu")

mainMenu()

# ---------------main driver ------------------

mainMenu()

captureimages.py

import csv

import cv2

import os

# counting the numbers

def is\_number(s):

try:

float(s)

return True

except ValueError:

pass

try:

import unicodedata

unicodedata.numeric(s)

return True

except (TypeError, ValueError):

pass

return False

# Take image function

def takeImages():

Id = input("Enter Your Id: ")

name = input("Enter Your Name: ")

if(is\_number(Id) and name.isalpha()):

cam = cv2.VideoCapture(0)

harcascadePath = "haarcascade\_frontalface\_default.xml"

detector = cv2.CascadeClassifier(harcascadePath)

sampleNum = 0

while(True):

ret, img = cam.read()

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

faces = detector.detectMultiScale(gray, 1.3, 5, minSize=(30,30),flags = cv2.CASCADE\_SCALE\_IMAGE)

for(x,y,w,h) in faces:

cv2.rectangle(img, (x, y), (x+w, y+h), (10, 159, 255), 2)

#incrementing sample number

sampleNum = sampleNum+1

#saving the captured face in the dataset folder TrainingImage

cv2.imwrite("TrainingImage" + os.sep +name + "."+Id + '.' +

str(sampleNum) + ".jpg", gray[y:y+h, x:x+w])

#display the frame

cv2.imshow('frame', img)

#wait for 100 miliseconds

if cv2.waitKey(100) & 0xFF == ord('q'):

break

# break if the sample number is more than 100

elif sampleNum > 100:

break

cam.release()

cv2.destroyAllWindows()

res = "Images Saved for ID : " + Id + " Name : " + name

row = [Id, name]

with open("StudentDetails"+os.sep+"StudentDetails.csv", 'a+') as csvFile:

writer = csv.writer(csvFile)

writer.writerow(row)

csvFile.close()

else:

if(is\_number(Id)):

print("Enter Alphabetical Name")

if(name.isalpha()):

print("Enter Numeric ID")

checkcamera.py

def camer():

import cv2

# Load the cascade

face\_cascade = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')

# To capture video from webcam.

cap = cv2.VideoCapture(0)

while True:

# Read the frame

\_, img = cap.read()

# Convert to grayscale

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

# Detect the faces

faces = face\_cascade.detectMultiScale(gray, 1.3, 5, minSize=(30, 30),flags = cv2.CASCADE\_SCALE\_IMAGE)

# Draw the rectangle around each face

for (x, y, w, h) in faces:

cv2.rectangle(img, (x, y), (x + w, y + h), (10,159,255), 2)

# Display

cv2.imshow('Webcam Check', img)

# Stop if escape key is pressed

if cv2.waitKey(1) & 0xFF == ord('q'):

break

# Release the VideoCapture object

cap.release()

cv2.destroyAllWindows()

recognize.py

import face\_recognition

import cv2

from openpyxl import Workbook

import datetime

# Get a reference to webcam #0 (the default one)

video\_capture = cv2.VideoCapture(0)

# Create a woorksheet

book=Workbook()

sheet=book.active

# Load images.

image\_1 = face\_recognition.load\_image\_file("1.jpg")

image\_1\_face\_encoding = face\_recognition.face\_encodings(image\_1)[0]

image\_5 = face\_recognition.load\_image\_file("5.jpg")

image\_5\_face\_encoding = face\_recognition.face\_encodings(image\_5)[0]

image\_7 = face\_recognition.load\_image\_file("7.jpg")

image\_7\_face\_encoding = face\_recognition.face\_encodings(image\_7)[0]

image\_3 = face\_recognition.load\_image\_file("3.jpg")

image\_3\_face\_encoding = face\_recognition.face\_encodings(image\_3)[0]

image\_4 = face\_recognition.load\_image\_file("4.jpg")

image\_4\_face\_encoding = face\_recognition.face\_encodings(image\_4)[0]

# Create arrays of known face encodings and their names

known\_face\_encodings = [

image\_1\_face\_encoding,

image\_5\_face\_encoding,

image\_7\_face\_encoding,

image\_3\_face\_encoding,

image\_4\_face\_encoding

]

known\_face\_names = [

"1",

"5",

"7",

"3",

"4"

]

# Initialize some variables

face\_locations = []

face\_encodings = []

face\_names = []

process\_this\_frame = True

# Load present date and time

now= datetime.datetime.now()

today=now.day

month=now.month

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]

# Assign attendance

if int(name) in range(1,61):

sheet.cell(row=int(name), column=int(today)).value = "Present"

else:

pass

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)

# Save Woorksheet as present month

book.save(str(month)+'.xlsx')

# Hit 'q' on the keyboard to quit!

if cv2.waitKey(1) & 0xFF == ord('q'):

break

# Release handle to the webcam

video\_capture.release()

cv2.destroyAllWindows()

trainimage.py

import os

import time

import cv2

import numpy as np

from PIL import Image

from threading import Thread

# -------------- image labesl ------------------------

def getImagesAndLabels(path):

# get the path of all the files in the folder

imagePaths = [os.path.join(path, f) for f in os.listdir(path)]

# print(imagePaths)

# create empth face list

faces = []

# create empty ID list

Ids = []

# now looping through all the image paths and loading the Ids and the images

for imagePath in imagePaths:

# loading the image and converting it to gray scale

pilImage = Image.open(imagePath).convert('L')

# Now we are converting the PIL image into numpy array

imageNp = np.array(pilImage, 'uint8')

# getting the Id from the image

Id = int(os.path.split(imagePath)[-1].split(".")[1])

# extract the face from the training image sample

faces.append(imageNp)

Ids.append(Id)

return faces, Ids

# ----------- train images function ---------------

def TrainImages():

recognizer = cv2.face\_LBPHFaceRecognizer.create()

harcascadePath = "haarcascade\_frontalface\_default.xml"

detector = cv2.CascadeClassifier(harcascadePath)

faces, Id = getImagesAndLabels("TrainingImage")

Thread(target = recognizer.train(faces, np.array(Id))).start()

# Below line is optional for a visual counter effect

Thread(target = counter\_img("TrainingImage")).start()

recognizer.save("TrainingImageLabel"+os.sep+"Trainner.yml")

print("All Images")

# Optional, adds a counter for images trained (You can remove it)

def counter\_img(path):

imgcounter = 1

imagePaths = [os.path.join(path, f) for f in os.listdir(path)]

for imagePath in imagePaths:

print(str(imgcounter) + " Images Trained", end="\r")

time.sleep(0.008)

imgcounter += 1

Screen shots

**Conclusion**

Therefore, for this **Project In Python** **Using OpenCV** We seek to provide a valuable attendance service for both teachers and students. Reduce manual process errors by provide automated and a reliable attendance system uses face recognition technology.