IOT Based Automated Attendance System Based on Face Recognition

INTRODUCTION

Overview :

Every organization requires an attendance system to maintain record of presence of student and employees. They have their own method to do the same. Some do manually and some use automated attendance system. Manual method includes pen and papers which consumes lot of time and wastage of resources. Also, it has risk of proxies and human error. Automated attendance system includes many methods like Radio Frequency Identification, Biometric, Face Recognition.

Purpose:

Face Recognition is one of the most efficient systems of all existing ones for identification of people. It can be used in school, colleges or any organization. To avoid the difficulty of taking attendance of large number, there is a need of automated attendance system that is fast and reduces the chance of fake attendance. In this technology system is developed for deploying an easy and a secure way of taking down attendance. This attendance is recorded, by continuously detecting faces of employees or students via camera as they enter the classroom. The software first detects the faces and simultaneously compares them with the predefined database.

LITERATURE SURVEY

Existing Problem:

In general, the attendance system of the student can be taken manually

Manual Student Attendance Management system is a process where a teacher

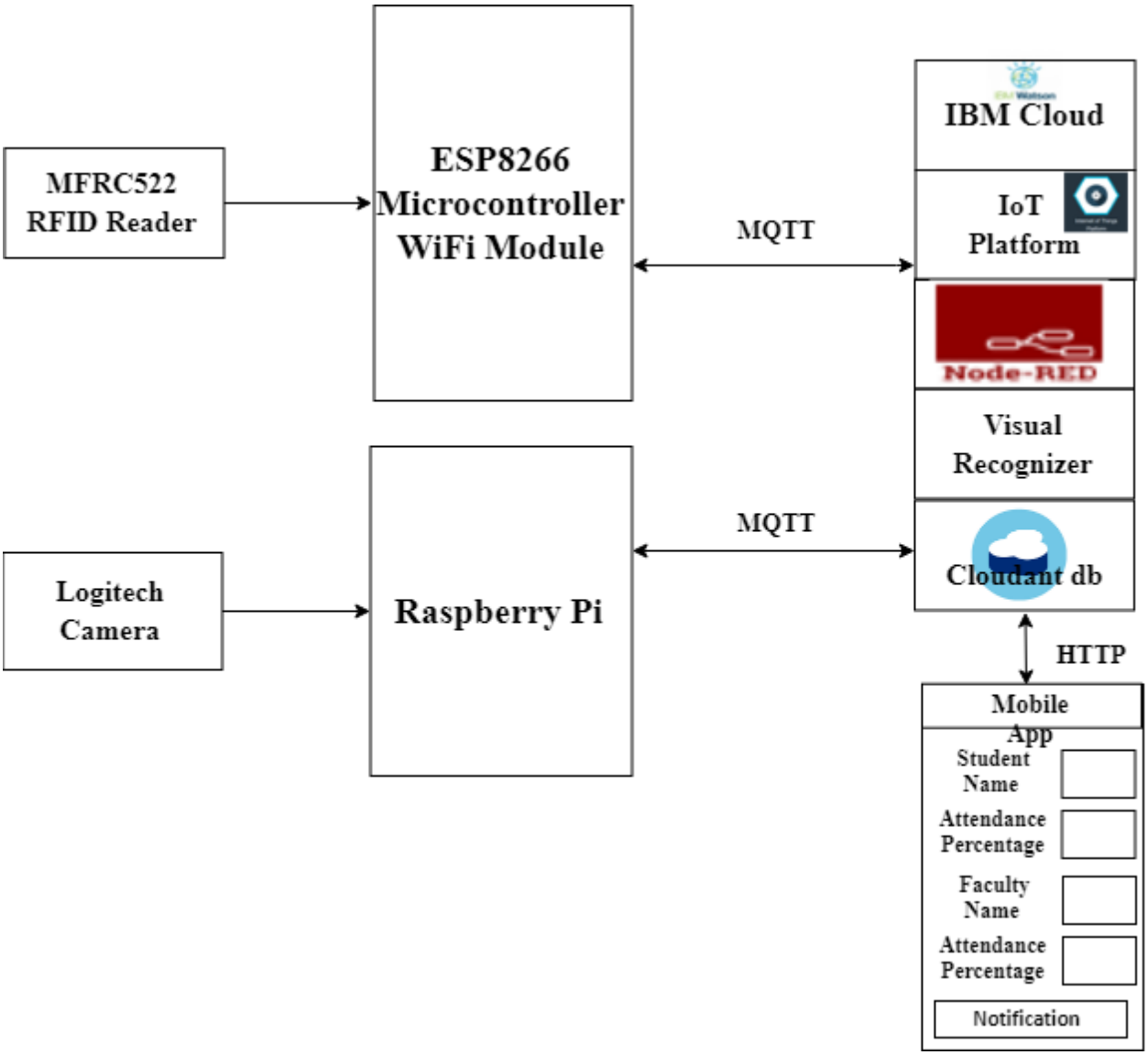
concerned with the particular subject need to call the students name and mark the attendance manually. Manual attendance may be considered as a time-consuming process or sometimes it happens for the teacher to miss someone or students may answer multiple times on the absence of their friends. So, the problem arises when we think about the traditional process of taking attendance in the classroom. To solve all these issues, we go with Automatic Attendance System (AAS).

Proposed solution :

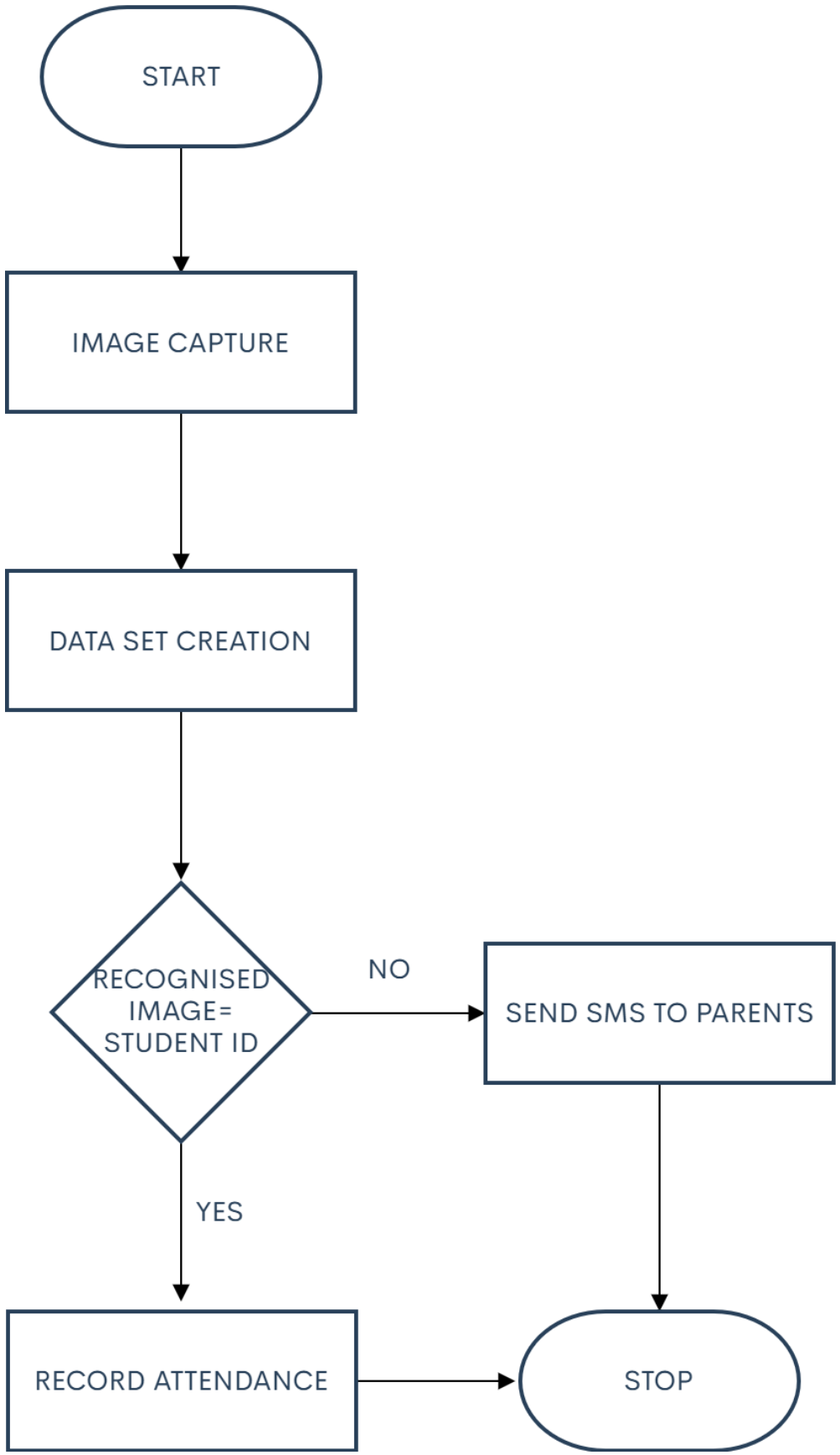
Automated Attendance System (AAS) is a process to automatically estimate the presence or the absence of the student in the classroom by using face recognition technology. It is also possible to recognize whether the student is sleeping or awake during the lecture and it can also be implemented in the exam sessions to ensure the presence of the student. The presence of the students can be determined by capturing their faces on to a high-definition monitor video streaming service, so it becomes highly reliable for the machine to understand the presence of all the students in the classroom.

THEORITICAL ANALYSIS

Block Diagram :


MFRC522
RFID Reader
ESP8266
Microcontroller
WiFi Module
MQTT
IBM Cloud
IoT
Platform
Node-RED
Visual
Recognizer
Logitech
Camera
Raspberry Pi
MQTT
Cloudant db
HTTP
Mobile
App
Student
Name
Attendance
Percentage
Faculty
Name
Attendance
Percentage
Notification

FLOW CHART


START
IMAGE CAPTURE
DATA SET CREATION
RECOGNISED IMAGE= STUDENT ID
NO
SEND SMS TO PARENTS
YES
RECORD ATTENDANCE
STOP

RESULT :

The project “IOT based automated attendance system based on face recognition” was designed such that the Attendance is taken by the smart device using face recognition in video streaming.

ADVANTAGES :

- Security levels will be significantly improved
- The integration process is easy and flawless
- High accuracy allows avoiding false identification
- Facial Recognition System is fully automated
- Time fraud will be excluded

DISADVANTAGES :

- Difficulties with data processing and storing
- Troubles with images size and quality
- Strong influence of the camera angle
- Poor Image Quality Limits Facial Recognition's Effectiveness
- Different Face Angles Can Throw Off Facial Recognition's Reliability
- Face recognition is less effective if facial expressions vary

APPLICATIONS :

- Used in educational institutions like schools and colleges
- Used in meetings

CONCLUSION :

Thus, the aim of this paper is to capture the video of the students, convert it into frames, relate it with the database to ensure their presence or absence, mark

attendance to the particular student to maintain the record. The Automated Classroom Attendance System helps in increasing the accuracy and speed ultimately achieve the high-precision real-time attendance to meet the need for automatic classroom evaluation.

FUTURE SCOPE :

- Automated Attendance System can be implemented in larger areas like in a seminar hall where it helps in sensing the presence of many people.
- Sometimes the poor lighting condition of the classroom may affect image quality which indirectly degrades system performance, this can be overcome in the latter stage by improving the quality of the video or by using some algorithms

BIBILOGRAPHY

1. N.Sudhakar Reddy, M.V.Sumanth, S.Suresh Babu, "A Counterpart Approach to Attendance and Feedback System using Machine Learning Techniques",Journal of Emerging Technologies and Innovative Research (JETIR), Volume 5, Issue 12, Dec 2018.

2. Dan Wang, Rong Fu, Zuying Luo, "Classroom Attendance Auto-management Based on Deep Learning",Advances in Social Science, Education and Humanities Research, volume 123,ICESAME 2017.

3. Akshara Jadhav, Akshay Jadhav, Tushar Ladhe, Krishna Yeolekar, "Automated Attendance System Using Face Recognition", International Research Journal of Engineering and Technology (IRJET), Volume 4, Issue 1, Jan 2017.

4. B Prabhavathi, V Tanuja, V Madhu Viswanatham and M Rajashekhara Babu, "A smart technique for attendance system to recognize faces through parallelism", IOP Conf. Series: Materials Science and Engineering 263, 2017.

5. Prajakta Lad, Sonali More, Simran Parkhe, Priyanka Nikam, Dipalee Chaudhari, " Student Attendance System Using Iris Detection", IJARIIE-ISSN(O)-2395-4396, Vol-3 Issue-2 2017.

6. Samuel Lukas, Aditya Rama Mitra, Ririn Ikana Desanti, Dion Krisnadi, "Student

Attendance System in Classroom Using Face Recognition Technique", Conference Paper DOI: 10.1109/ICTC.2016.7763360, Oct 2016.

7. K.Senthamil Selvi, P.Chitrakala, A.Antony Jenitha, "Face Recognition Based Attendance Marking System", IJCSMC, Vol. 3, Issue. 2, February 2014.

8. Yohei KAWAGUCHI, Tetsuo SHOJI, Weijane LIN, Koh KAKUSHO, Michihiko MINOH, "Face Recognition-based Lecture Attendance System", Oct 2014.

9. Shireesha Chintalapati, M.V. Raghunadh, "Automated Attendance Management System Based On Face Recognition Algorithms", IEEE International Conference on Computational Intelligence and Computing Research, 2013.

Appendix:-

The growth of Internet of Things have given rise to New possibilities in terms of smart cities. Smart attendence management systems have always been at the core of Multi National Compamies. In this project, we address the issue of attendence and present an IoT based web application smart attendence system. The system that recognises the person and attendence given to that person . The efforts made in this project are indented to improve the attendence system in the schools also and thereby aiming to enhance the quality of life of its people. In our system user can view the attendence.

Source code:-

```
# -*- coding: utf-8 -*-

import datetime
```

```
import ibm_boto3
from ibm_botocore.client import Config, ClientError
import cv2
import numpy as np
import sys
import ibmiotf.application
import ibmiotf.device
import random
import time
import json
from ibm_watson import VisualRecognitionV3
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
from cloudant.client import Cloudant
from cloudant.error import CloudantException
from cloudant.result import Result, ResultByKey
#Provide your IBM Watson Device Credentials
organization = "uxl664"
deviceType = "iot"
deviceId = "015"
authMethod = "token"
authToken = "6303898175"
```

```
def myCommandCallback(cmd):
        print("Command received: %s" % cmd.data)
        print(cmd.data['command'])


        if(cmd.data['command']=="open"):
                print("door open")
                if(cmd.data['command']=="close"):
                print("door close")
        if(cmd.data['command']=="present"):
                print("present")
        if(cmd.data['command']=="absent"):
                print("absent")
try:
deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,
"auth-method": authMethod, "auth-token": authToken}
deviceCli = ibmiotf.device.Client(deviceOptions)
#.....
except Exception as e:
print("Caught exception connecting device: %s" % str(e))
sys.exit()
# Connect and send a datapoint "hello" with value "world" into the cloud as an
event of type "greeting" 10 times
deviceCli.connect()
face_classifier=cv2.CascadeClassifier("haar-face.xml")
```

```
#It will read the first frame/image of the video

video=cv2.VideoCapture('video3.mp4')

COS_ENDPOINT = "https://s3.us.cloud-object-storage.appdomain.cloud" # Current list avaiable at https://control.cloud-object-storage.cloud.ibm.com/v2/endpoints

COS_API_KEY_ID = "75EFmi6-6DPQVMknKBchXOdQQqa2mHmorb7nWGQPVzwj" # eg "W00YiRnLW4a3fTjMB-oiB-2ySfTrFBIQQWanc--P3byk"

COS_AUTH_ENDPOINT = "https://iam.cloud.ibm.com/identity/token"

COS_RESOURCE_CRN = "crn:v1:bluemix:public:cloud-object-storage:global:a/e7faebd9d761480bb961b30b6eda9b91:92440559-bf42-49b9-8ea2-267eed9d2bda::"


client = Cloudant("8e7237b7-402b-4171-a550-9af9eda65bf6-bluemix", "9dc4ca6821a0f866d4527339c89a0e15323c8e95b6a0bc15d2cd28df6df6ced9", url="https://8e7237b7-402b-4171-a550-9af9eda65bf6-bluemix:9dc4ca6821a0f866d4527339c89a0e15323c8e95b6a0bc15d2cd28df6df6ced9@8e7237b7-402b-4171-a550-9af9eda65bf6-bluemix.cloudantnosqldb.appdomain.cloud")

client.connect()

database_name = "doorbell"


# Create resource

cos = ibm_boto3.resource("s3",

    ibm_api_key_id=COS_API_KEY_ID,

    ibm_service_instance_id=COS_RESOURCE_CRN,

    ibm_auth_endpoint=COS_AUTH_ENDPOINT,

    config=Config(signature_version="oauth"),

    endpoint_url=COS_ENDPOINT
```

```
)
def multi_part_upload(bucket_name, item_name, file_path):
    try:
        print("Starting file transfer for {0} to bucket: {1}\n".format(item_name, bucket_name))
        # set 5 MB chunks
        part_size = 1024 * 1024 * 5


        # set threadhold to 15 MB
        file_threshold = 1024 * 1024 * 15
        # set the transfer threshold and chunk size
        transfer_config = ibm_boto3.s3.transfer.TransferConfig(
            multipart_threshold=file_threshold,
            multipart_chunksize=part_size
        )
        # the upload_fileobj method will automatically execute a multi-part upload
        # in 5 MB chunks for all files over 15 MB
        with open(file_path, "rb") as file_data:
            cos.Object(bucket_name, item_name).upload_fileobj(
                Fileobj=file_data,
                Config=transfer_config
            )
        print("Transfer for {0} Complete!\n".format(item_name))
    except ClientError as be:
```

```
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to complete multi-part upload: {0}".format(e))
authenticator = IAMAuthenticator('p94o5iYZ4Yy_mJBsjKJMDWOcpgQpKb1ThS667EYS5GiL')
visual_recognition = VisualRecognitionV3(
    version='2018-03-19',
    authenticator=authenticator
)
visual_recognition.set_service_url('https://api.us-south.visual-recognition.watson.cloud.ibm.com/instances/5da92653-bc4d-400d-8163-5c0602441914')
while True:
    #capture the first frame
    check,frame=video.read()
    gray=cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    #detect the faces from the video using detectMultiScale function
    faces=face_classifier.detectMultiScale(gray,1.3,5)
    #drawing rectangle boundries for the detected face
    for(x,y,w,h) in faces:
        cv2.rectangle(frame, (x,y), (x+w,y+h), (127,0,255), 2)
        cv2.imshow('Face detection', frame)
        picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
        picname=picname+".jpg"
        pic=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
```

```
        datetym=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
        cv2.imwrite(picname,frame)
        with open(picname, 'rb') as images_file:
            classes = visual_recognition.classify(
                images_file=images_file,
threshold='0.6',classifier_ids='DefaultCustomModel_1112676152').get_result()
            print(json.dumps(classes, indent=2))
            person1=classes['images'][0]['classifiers'][0]['classes'][0]['class']
            print("person1")
            print(person1)
            if(person1=="sachin.zip"):
                person1="Sachin"
        person=1
        my_database = client.create_database(database_name)

multi_part_upload("project-donotdelete-pr-7irtazedott6ek",picname,pic+".jpg")

        if my_database.exists():
            print("'{database_name}' successfully created.")
            json_document = {
                "_id": pic,

"link":COS_ENDPOINT+"/project-donotdelete-pr-7irtazedott6ek/"+picname
                }
            new_document = my_database.create_document(json_document)
```

```
            if new_document.exists():
                print("Document '{new_document}' successfully created.")
        time.sleep(1)
        p=34
     data = {"d":{ 'present' : p, 'person': person1,'datetym':datetym}}
        #print data
        def myOnPublishCallback():
            print ("Published data to IBM Watson")
        success = deviceCli.publishEvent("Data", "json", data, qos=0,
on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(1)
        deviceCli.commandCallback = myCommandCallback
        person=0
    #waitKey(1)- for every 1 millisecond new frame will be captured
    Key=cv2.waitKey(1)
    if Key==ord('q'):
        #release the camera
            video.release()
        #destroy all windows
            cv2.destroyAllWindows()
            break
deviceCli.disconnect()
```