**1. INTRODUCTION:**

**1.1 OVERVIEW:-**

*Automated Teller Machine (ATM) are well known device typically used by personal and business financial transactions. Use of technology in the field of security plays an important role in increasing the security as well as reducing the manpower effort. By using this IOT based smart security system we can provide high security as the person is allowed into the ATM if and only if their face is detected.*

**1.2 PURPOSE:-**

*The purpose of this smart security system is to provides high security with intelligence .Avoid theft for ATM machines.And also to identify the robbery quickly.*

**2 .LITERATURE SURVEY:**

**2.1 EXISTING PROBLEM:-**

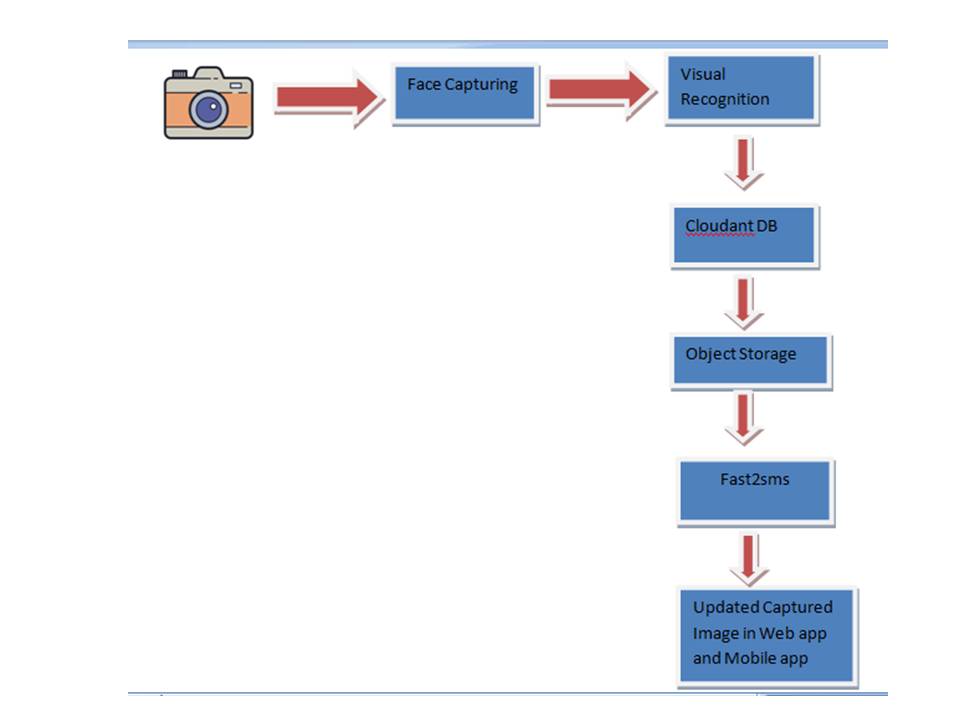
*In today's technically advanced world, autonomous system are gaining rapid popularity .As the automation has been increased and ATM has been installed everywhere. However crime related activities are increased. Eventhough their are CC TV footage these crimes are not reduced because their is no clear image of the entered person and so they cannot be detected this is one of the major problem that we are facing in this real time world.*

**2.2 PROPOSED SOLUTION:-**

*In this project we analyzed what is the problem people faced in the existing technology. This study is going to suggest the method of rapid reaction and minimization of crimes regarding ATM.In this method the ATM doors will be automatically unlock if and only if the person face is detected otherwise the doors will be locked.If any other creatures tries to enter the doors wont open. And this is one of the finest method if any robbery takes place their face be detected and it will stored in the object storage another advantage is whenever a person detected the security guard will receive sms as "authorized person entered" it was developed by using fast to sms* .

**3. THEORITICAL ANALYSIS:**

**3. 1 BLOCK DIAGRAM:-**



**3.2 HARDWARE /SOFTWARE DESIGNING:-**

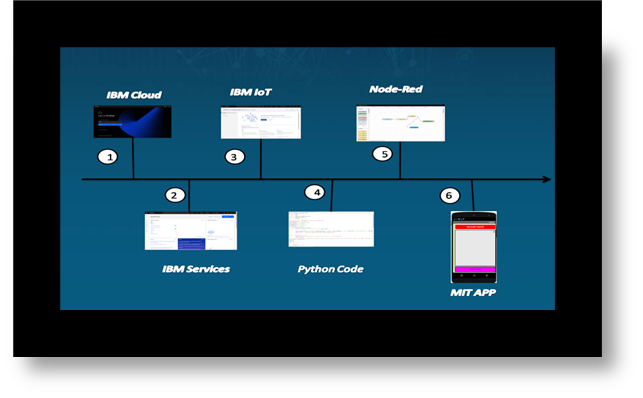
The software part deals in programming.Code is written and compiled the source code which has been written in the python language.In this project node red service is used for building a web app and with the help of MIT app inverter we designed our app.

**4. EXPERIMENTAL INVESTIGATION**:

I*n this experimental study we are going to know that if a person want to*

*enter into the ATM first of all ,his face need to be detected and captured then the doors will be automatically unlocked at the same time the respective person will receive SMS as "Authorized person entered".The data will be stored in the cloudant DB and object storage in bucket.If any other creature like cat tries to enter the doors will be automatically locked and the respective person will receive SMS as"unauthorized person entered".In this experiment the IBM Watson IOT is a cognitive system that learns from and infuses intelligence into a physical world.*

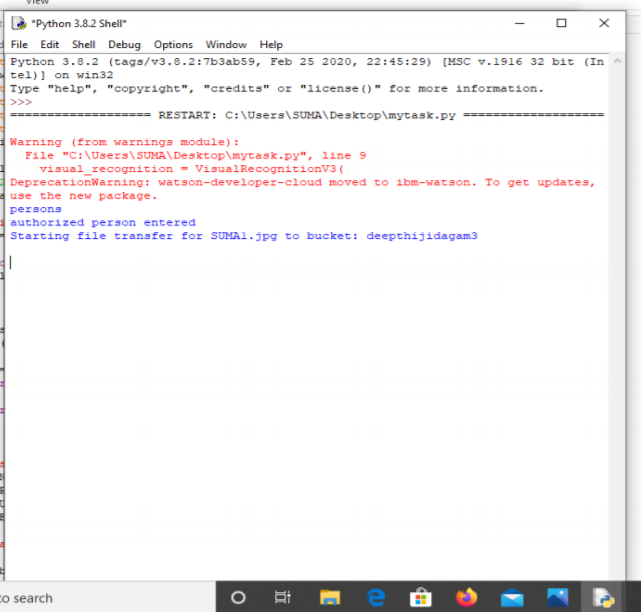
**5.FLOW CHART:-**



**6. RESULT:-**

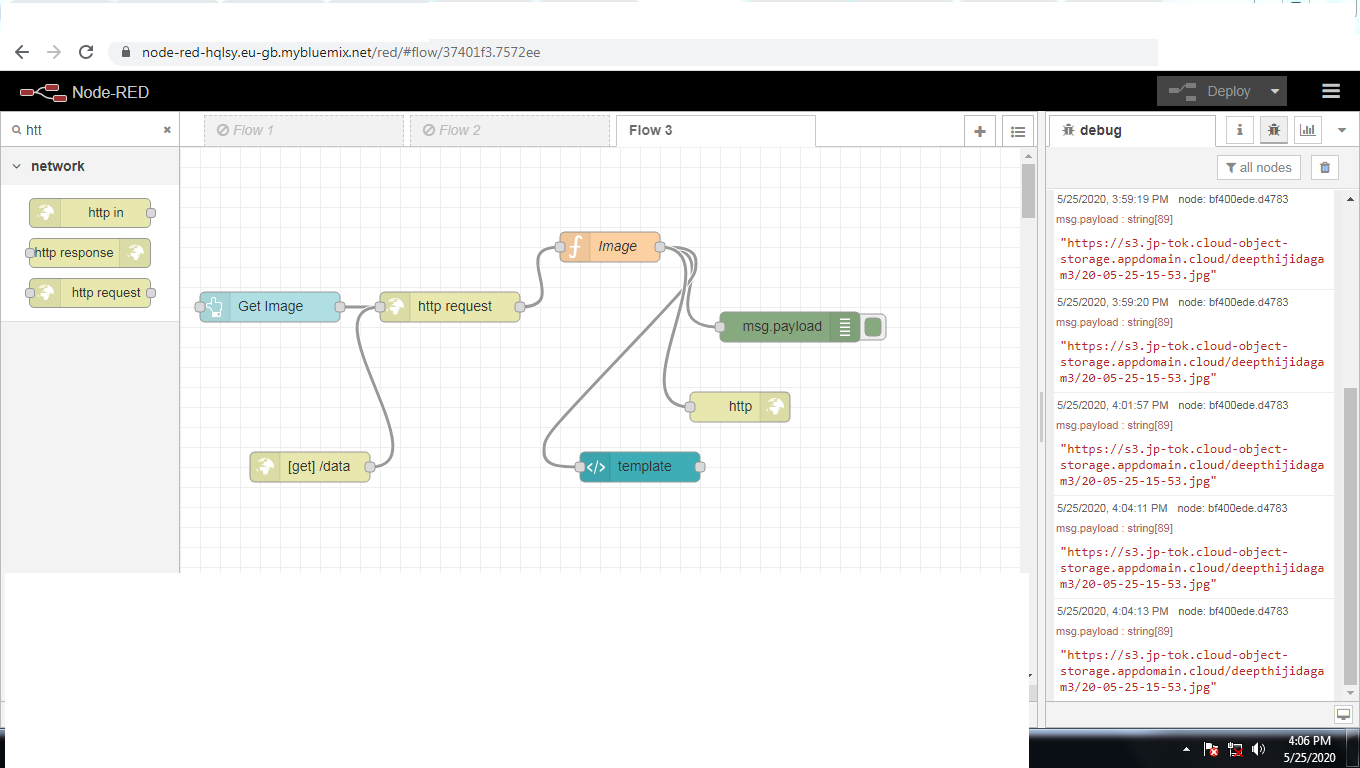
*Finally the image which is detected will be stored in the object storage and by using fast to SMS the respective person will receive the SMS immediately after a person face is detected.This is a smart idea to avoid theft forATM machines and also to identify the robber*y.

**EXECUTION:**

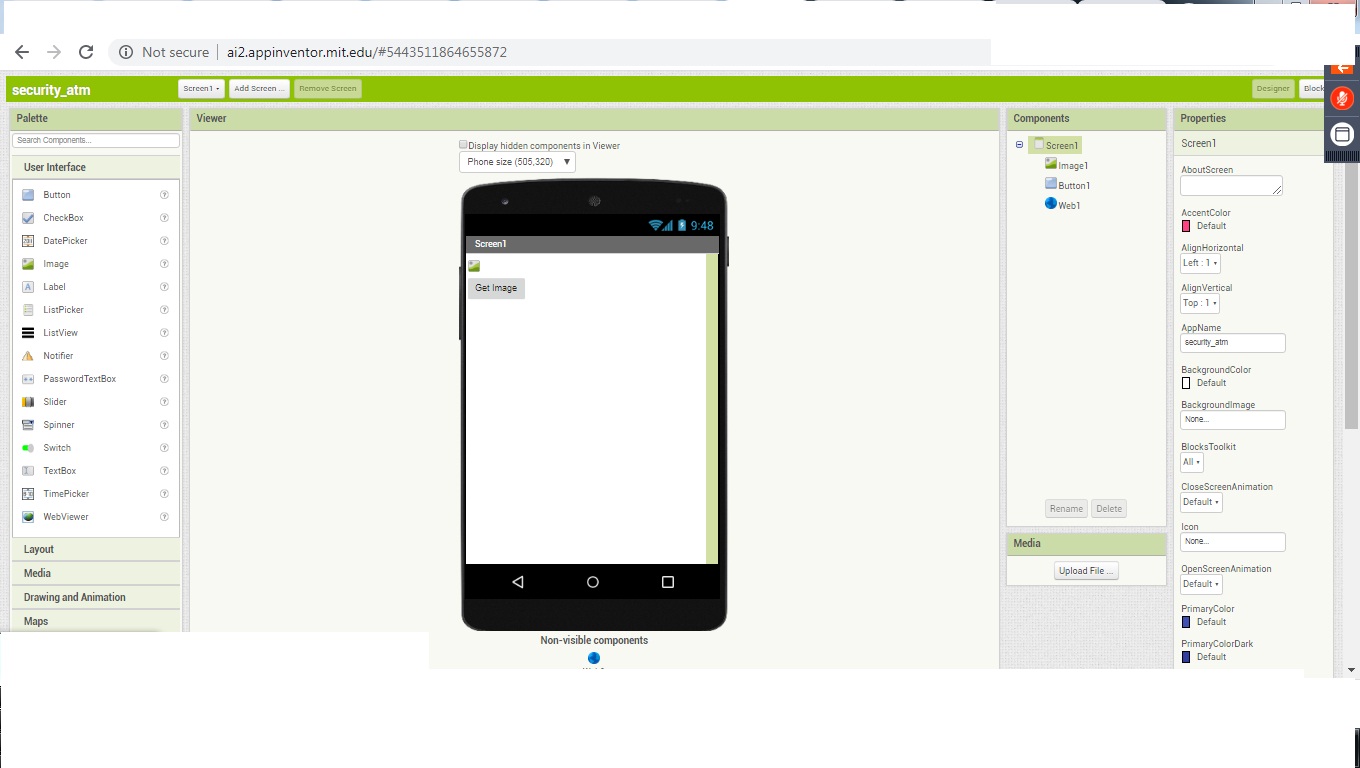




**NODE RED:**



**MIT APP :**



**7.ADVANTAGES & DISADVANTAGES :**

**Advantages:**

* *provide smart security in ATM.*
* *easy way for fastest investigation whenever robbery takes place.*
* *Intimates the respective person when ever persons entered into the ATM.*

**Disadvantages:**

* *unable to stop robbery.*
* *hacking the respective person(security guard) details.*
* *the doors will be in locked position if persons are simultaneously entering and returning from ATM.*

**8. APPLICATIONS:**

*Banks,Shopping malls, Jewellery shops,and Government Office which deal with confidential issues .*

**9. CONCLUSION:**

*The project "Smart Security System in ATM " has been successfully designed and implemented.It has been developed by integrating all the features of IBM cloud Services and with the help of growing technology project has been successfully implemented.At last ,this project overcomes the problem and provides the easiest way to secure the ATM.*

**FUTURE SCOPE:**

*1.* *This smart security system can be deployed to make the entire system more user -friendly and efficient.*

*2. Highly secure protocols can be deployed to ensure their is not security breach.*

**10. BIBILOGRAPHY:**

1 *Lwin,H,Khaing,A,Tun,H.2015 Automatic door access system using face detection.*

*2 "Advanced security model for detecting in ATM transaction"Vivek V.jog Assistant professor dept of comp.Engg.SKNCOE.*

*3 Kannan,P(2013) and Ms P. Meenakshi Vidhya "Desigin and Implementation of security based ATM monitoring system*".

**11. APPENDIX:**

**SOURCE CODE:**

*import json*

*from watson\_developer\_cloud import VisualRecognitionV3*

*import cv2*

*import numpy as np*

*import datetime*

*import ibm\_boto3*

*from ibm\_botocore.client import Config, ClientError*

*#CloudantDB*

*from cloudant.client import Cloudant*

*from cloudant.error import CloudantException*

*from cloudant.result import Result, ResultByKey*

*import requests*

*visual\_recognition = VisualRecognitionV3(*

*'2018-03-19',*

*iam\_apikey='tDYM20kS5sD2jkAApAOrb83RdBO0I2nl-rCuVp10gsJH')*

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

*video=cv2.VideoCapture(0)*

*def vis(x):*

*with open('./'+x, 'rb') as images\_file:*

*classes = visual\_recognition.classify(*

*images\_file,*

*threshold='0.6',*

*classifier\_ids='DefaultCustomModel\_877446449').get\_result()*

*a=classes['images'][0]['classifiers'][0]['classes'][0]['class']*

*print(a)*

*if a=='persons' :*

*print("authorized person entered")*

*else:*

*print("unauthorized person not entered")*

*# Constants for IBM COS values*

*COS\_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud" # Current list avaiable at https://control.cloud-object-storage.cloud.ibm.com/v2/endpoints*

*COS\_API\_KEY\_ID = "\_08kgDODk4BV2-e0fsG5SX-cbLuIAkweyYLqf1gQny62" # eg "W00YiRnLW4a3fTjMB-odB-2ySfTrFBIQQWanc--P3byk"*

*COS\_AUTH\_ENDPOINT = "https://iam.cloud.ibm.com/identity/token"*

*COS\_RESOURCE\_CRN = "crn:v1:bluemix:public:cloud-object-storage:global:a/b03790c764f74139bbf0115be6990c81:4ae1d3b9-2de4-408a-a343-4536b529d067::" # eg "crn:v1:bluemix:public:cloud-object-storage:global:a/3bf0d9003abfb5d29761c3e97696b71c:d6f04d83-6c4f-4a62-a165-696756d63903::"*

*# 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*

*)*

*#Provide CloudantDB credentials such as username,password and url*

*client = Cloudant("959488fa-a1c4-4c85-8129-ed1ee8360c62-bluemix", "3d9d719012c25813207ab56cb5ac36c7b057529f14f42a83ab6a95e9da1e2e4b", url="https://959488fa-a1c4-4c85-8129-ed1ee8360c62-bluemix:3d9d719012c25813207ab56cb5ac36c7b057529f14f42a83ab6a95e9da1e2e4b@959488fa-a1c4-4c85-8129-ed1ee8360c62-bluemix.cloudantnosqldb.appdomain.cloud")*

*client.connect()*

*#Provide your database name*

*database\_name = "atm1"*

*my\_database = client.create\_database(database\_name)*

*if my\_database.exists():*

*print("'{database\_name}' successfully created.")*

*face\_classifier=cv2.CascadeClassifier("haarcascade\_frontalface\_default.xml")*

*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))*

*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)*

*print(faces)*

*#drawing rectangle boundries for the detected face*

*#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")*

*x="picname.jpg"*

*cv2.imwrite(x,frame)*

*multi\_part\_upload("deepthijidagam3", x, x)*

*json\_document={"link":COS\_ENDPOINT+"/"+"deepthijidagam3"+"/"+x}*

*new\_document = my\_database.create\_document(json\_document)*

*vis(x)*

*# Check that the document exists in the database.*

*if new\_document.exists():*

*print(f"Document successfully created.")*

*r = requests.get("https://www.fast2sms.com/dev/bulk?authorization=OMyK5jnSDx9CG40kTNihZ6szEpYRqBPJaQAdr7v1bHg2cmLfoUgiV2jnM75hLRKcC6QAS9ePqOWBJ3dy&sender\_id=FSTSMS&message=Some one at door&language=english&route=p&numbers=9949599804")*

*print(r.status\_code)*

*#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*