## horizontal line

INTELLIGENT CROP PROTECTION

# Goals

1. Detecting the different animals and birds using IBM visual recognition service for protecting the crop from damage.
2. Whenever any birds and animals are detected the image will be captured and for keeping away them from the crop we can keep a siren and we can blink the LEDs.
3. Whenever there is low soil moisture the admins will be alerted and they can control the motors using the mobile application.

Internship project assigned by:



Team Name: Techie-Tribe

Team members:

K.HARISH

S.SURYA MOHAN

KEVIN

# Introduction

Human-animal conflicts arise due to the diminishing natural resources and habitats as a result of the rise of encroachments, poaching, urbanization, and industrialization. As a result, the loss in yield due to animal attacks has been on a steady increase in the past few years. Even though agriculture is a vast sector, the average farmer would find it difficult to invest in technological solutions to prevent loss in yield, especially for security purposes. We propose a solar-powered, IoT based intelligent system that can be used to prevent crop damage due to wild animal attacks. The system implements IoT technology along with simple sensors. It makes use of four junction boxes that are central to the system architecture. The junction boxes contain MCU, XBee module, RTC, and GSM. In addition, each box hosts either two lasers or two LDR sensors. The GSM module enables the system to send a message to the farmer when the system is triggered by the animal intrusion. The junction boxes will be placed in each corner in such a way that the laser transmitter in one junction box and the LDR of the neighboring junction box are in the line of sight. The proposed system can improve the yield of crops and in turn help farmers to increase their earning. This system indirectly helps the farmers to have a good sleep during the night as there is no need for them to keep patrolling their fields all through the night. This project is funded by IEEE SIGHT.

## Overview

Surveillance plays a major role in many fields be it at home, hospitals, schools, public places, farmlands etc. It helps us to monitor a certain area and prevent theft and also provides proof of evidence. In the case of farmlands or agricultural lands surveillance is very important to prevent unauthorized people from gaining access to the area as well as to protect the area from animals. Various methods aim only at surveillance which is mainly for human intruders, but we tend to forget that the main enemies of such farmers are the animals that destroy the crops. This leads to a poor yield of crops and significant financial loss to the owners of the farmland. This problem is so pronounced that sometimes the farmers decide to leave the areas barren due to such frequent animal attacks. This system helps us to keep away such wild animals from the farmlands as well as provides surveillance functionality.

## 

## 

## 

## Purpose

To intelligently monitor the temperature, humidity of your farm at fingertips by using sensors and cloud services. And also protect your farm by continuously watching it and get notified when an animal or bird enters your farm.

# Literature survey

T.Gayathri et al, [10] proposed the system for monitoring the growing status of the corn (maize) plant continuously and intimate the agriculturist using wireless sensor network (WSN). But in practice, cultivator faces too much effort in the farmland. This paper makes eases the work of the farmer in cultivated land through the usage of different kind of sensors. The two LDR sensors are interfaced with PIC16F877A microcontroller whereas its top array receives solar radiation for supply current and the bottom of the LDR array is for measuring leaf area index (LAI). The humidity sensor will compute the moisture level in the corn field, if the level decreases, then it automatically switches ON the DC motor. All the particulars of farmland are sent to the farmer through GSM and revel in the LCD screen. The temperature sensor will find the intensity of heat present in the soil. PH sensor is used to find the soil

alkalinity which is essential for plant nutrition.

V Nainwal, et al, [16] Sensors are used to detect the presence of objects in the surveillance area and the information is collected over time to extract the event of interest. The information gathered by the surveillance camera i.e., video or still images could be used for further analysis and detection of the intruding object. This system does not utilize advanced techniques for alerting the owner of that area.

Sneha Nahatkar et al, [1] proposed a home embedded surveillance system which evaluates the development of a low cost security system using small PIR (Pyroelectric Infrared) sensor built around a microcontroller with ultra-low alert power. The system senses the signal generated by PIR sensor detecting the presence of individuals not at thermal equilibrium with the surrounding environment. On detecting the presence of any unauthorized person in any specific time interval, it triggers an alarm & sets up a call to a predefined number through a GSM modem. After the MCU sends the sensor signals to the embedded system, the program starts the Web camera which then captures the images which can be viewed and analysed later.

Puja G, Mohammad Umair Bagali proposed the system. This project is based on surveillance with an animal ward-off system employed in farmlands in order to prevent crop vandalization by wild animals. In addition to providing protection this system distinguishes between an intruder and an authorized person using RFID’s, various PIR sensors are deployed in the area to detect any motion and hence turns ON a camera when movement is detected, thereby providing real time monitoring. Automation of certain methods used to prevent the wild animals from entering the farmlands and destroying the crops, an electronic firecracker.

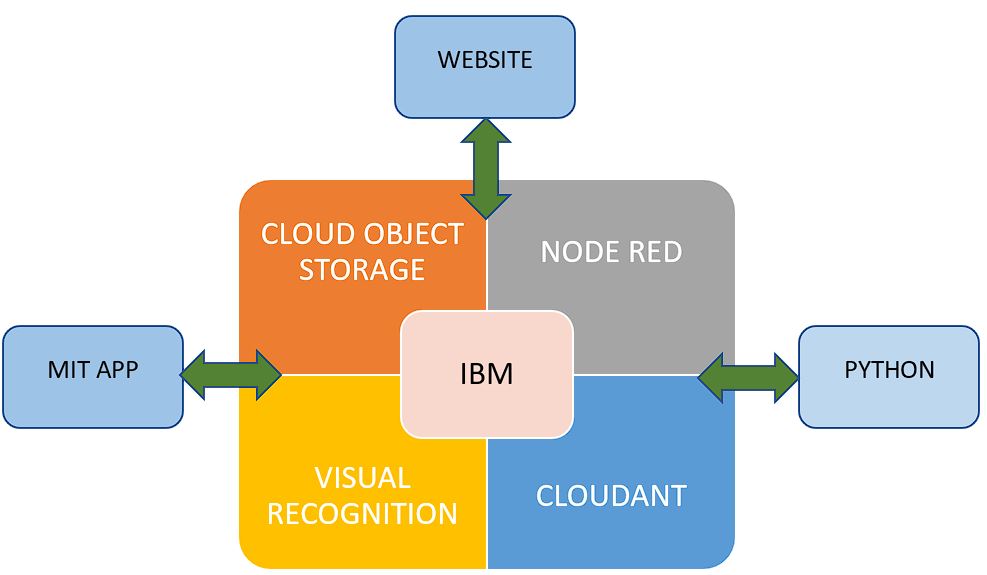
## Existing problem

The existing systems mainly provide the surveillance functionality. Also these systems don’t provide protection from wild animals, especially in such an application area. They also need to take actions based on the on the type of animal that tries to enter the area, as different methods are adopted to prevent different animals from entering such restricted areas. Also the farmers resort to the other methods by erecting human puppets and effigies in their farms, which is ineffective in warding off the wild animals, though is useful to some extent to ward off birds .The other commonly used methods by the farmers in order to prevent the crop vandalization by animals include building physical barriers, use of electric fences and manual surveillance and various such exhaustive and dangerous methods.

## Proposed solution

The proposed system uses IBM cloud services which form the main heart of the system and the different sensors and camera are interfaced with it. The python continuously runs the camera and sends the images to the IBM visual recognition service. If the service recognizes any animal or bird in the image then it alerts the farm owner by an SMS and also sends the captured image the mobile application. By the app we can also monitor the live temperature and humidity levels in the farm.

# Block diagram



# SOFTWARE DESIGNING

## IBM cloud services

## Python

## Mit app inventor

# IBM cloud services

IBM provides various services required by IoT applications. It provides visual recognition service to identify the objects and living things. We have used this service to identify the animals and birds on our farm. IBM also provides cloud storage of images by cloud object storage service in which we have uploaded the images of our farm. It also provides cloudant service to store the text such as temperature and humidity values. We have also used the Node Red services provided by IBM to create a website and also to care of the flow of data.

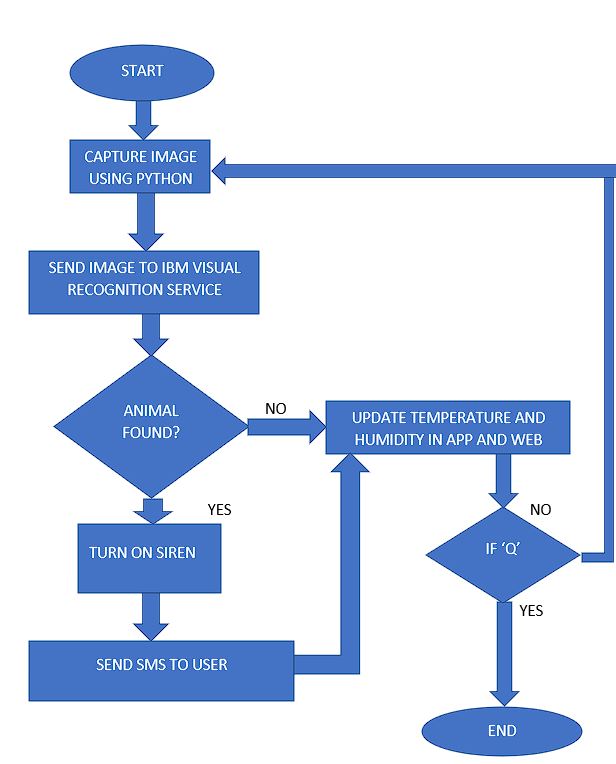
# Python

Python is primarily used to continuously monitor the farm by the camera using cv2 package. By using the credentials of IBM service we send the images to IBM visual recognition service using python. And also send the temperature and humidity values to cloudant. We use Twilio package the alert the user about the animal attack on their farm through SMS.

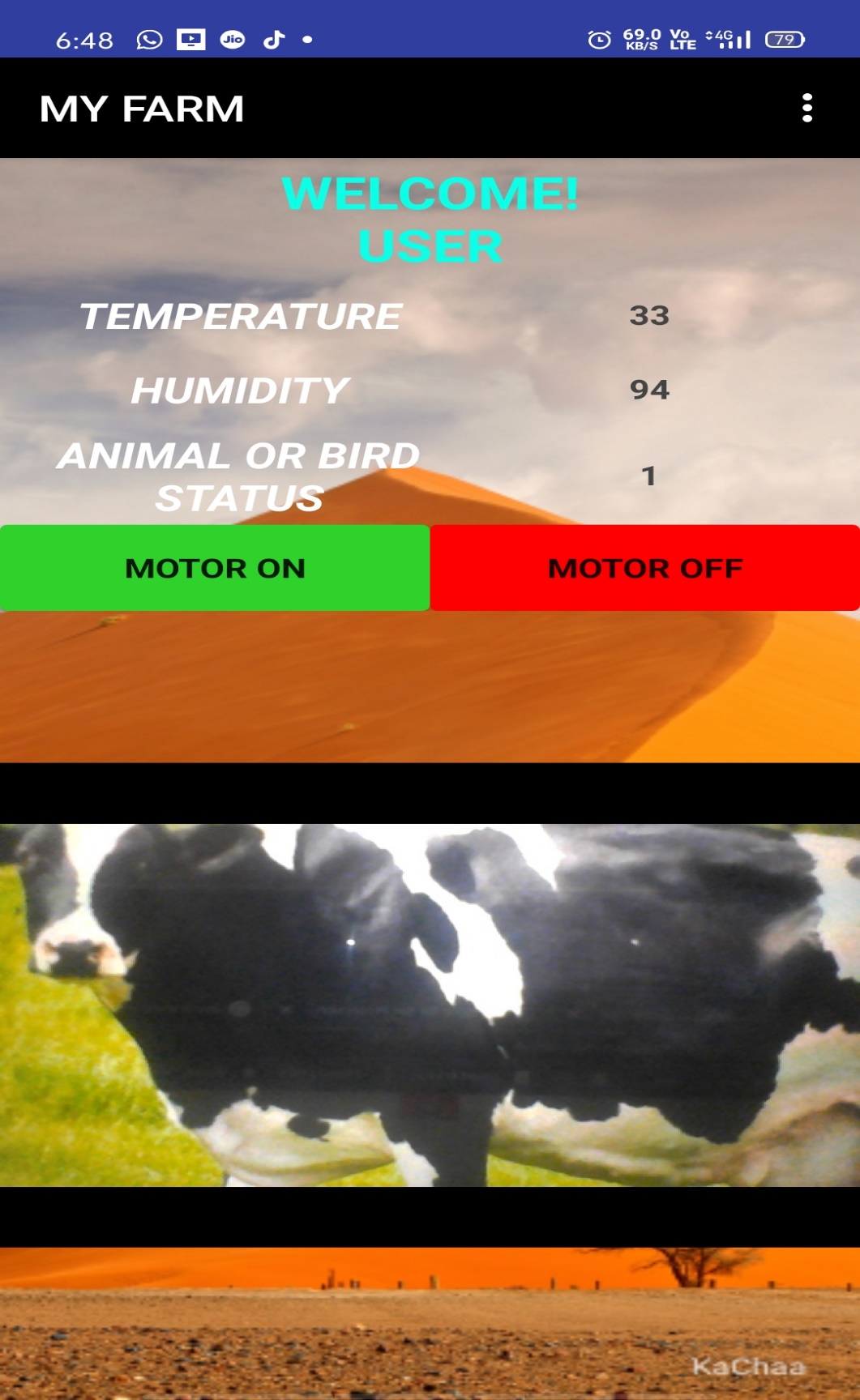
# Mit app inventor

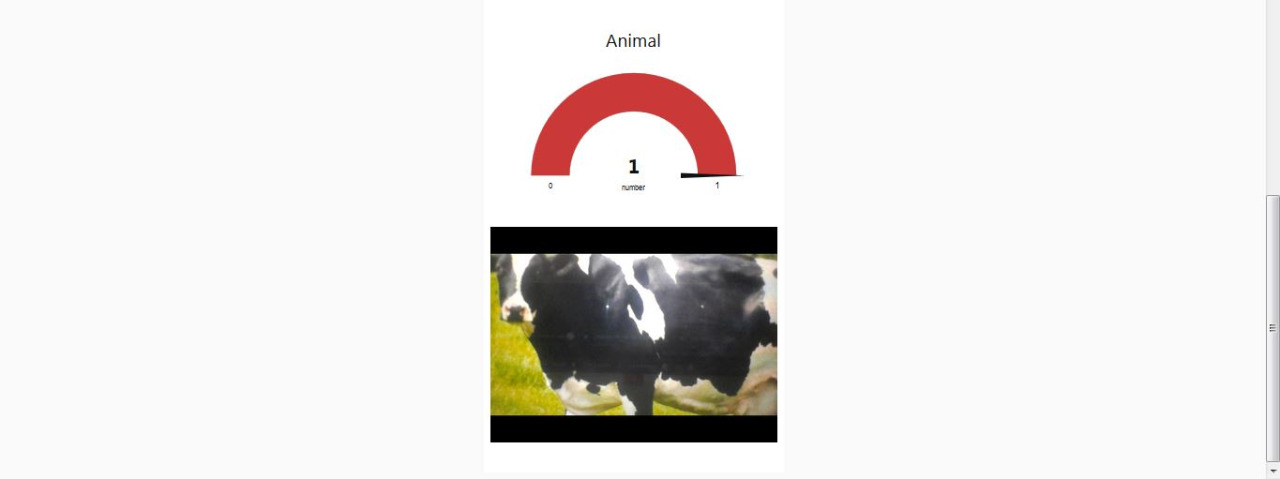
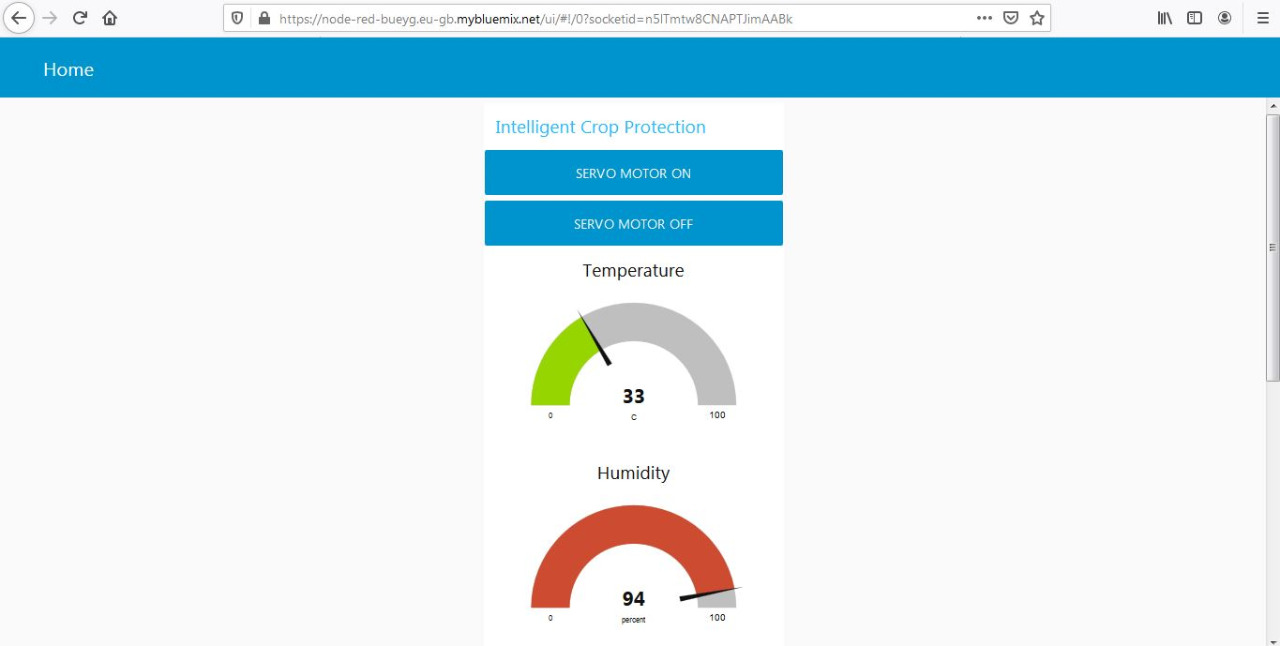
This platform is used to create an android application. This website provides a simple block system to just drag and drop procedures to make the user feel comfortable. We have designed our app on this website which receives the temperature, humidity and also images of the farm of the user.

# FLOWCHART



# RESULT





# ADVANTAGES

This system is very effective and carries following features and merits in comparison to the other solutions that exist in the current time.

1. Effective, accurate and adaptive:-

This system is very effective in driving off the animals from the fields and keeping them away. It accurately determines the presence of animals in the fields and sounds the buzzer. It does not sound the buzzer due to the presence of a human being or due to some random motion. The ultrasonic buzzer is very effective against animals and causes no noise pollution.

2. Requires no human supervision:-

This system requires almost no human supervision, except for the task of switching the system on and off. The system is capable of turning the buzzers on automatically and warding off the animals thus protecting the fields from any damage.

3. Economical:-

This system is economical as compared to many of the existing solutions like electric fences, brick walls and manual supervision of the fields. Thus it saves a lot of money of the farmer.

4. Real time monitoring:-

This system works in real time to detect the animals in the fields. The system enables the farmer to have a real time view of his fields from any place via internet and even provides manual buzzer controls if the need arises to use them. Thus the farmer is in effective control of the system and can manually sound the buzzer if needed.

5. Causes no harm to animals and humans:-

This system is totally harmless and doesn’t injure animals in any way. It also doesn’t cause any harm to humans. Also this system has a very low power requirement thus reducing the hazards of electric shocks.

# DISADVANTAGES

1. Required fast internet.
2. Cloud storage is costly.

# Applications

1. This can be used to protect the farm.
2. It can be used as surveillance in houses.
3. To track the live temperature and humidity of an area.

# CONCLUSIONS

The problem of crop vandalization by wild animals has become a major social problem in the current time. It requires urgent attention and an effective solution. Thus this project carries a great social relevance as it aims to address this problem. Hence we have designed a smart embedded farmland protection and surveillance based system which is low cost, and also consumes less energy. The main aim is to prevent the loss of crops and to protect the area from intruders and wild animals which pose a major threat to the agricultural areas. Such a system will be helpful to the farmers in protecting their orchards and fields and save them from significant financial losses and also saves them from unproductive efforts that they endure for the protection of their fields. This system will also help them in achieving better crop yields thus leading to their economic wellbeing.

# Future Scope

1. In addition to providing protection this system distinguishes between an intruder and an authorized person using RFID’s.

2. We use Haar feature based cascade classifiers for object detection to distinguish between the animal and human.

3. When such intrusions occur the cameras employed are turned ON which capture an image and start recording the video for some time which will be stored on the SD card as well as stored on cloud i.e. dropbox, the land owner can then view the video on any smart device.

4. If the motion detection is due to an authorized person with a valid RFID, who is mostly a farm worker, his attendance gets recorded automatically.

5. We can design a IOT based application to provide an image and video feed to farmer on any smart device and farmer will be notified when there is an intrusion in the farm by animal along with additional information of humidity and temperature

# BIBLIOGRAPHY:

[1]. Ms. Sneha Nahatkar, Prof. Avinash Gaur, Prof. Tareek M. Pattewa “Design of a Home Embedded Surveillance System with Pyroelectric Infrared Sensor & Ultra-Low Alert Power” International Journal of

Advanced Research in Electronics and Communication Engineering (IJARECE) Volume 1, Issue 3, September 2012.

[2]. M. Sathishkumar1, S.Rajini “Smart Surveillance System Using PIR Sensor Network and GSM” International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 4 Issue 1, January 2015.] S. Sivagamasundari, S. Janani, “Home surveillance system based on MCU and GSM”, International journal of communications and engineering, 2014, volume 06– no.6.

[3]. Padmashree S. Dhake, Sumedha S. Borde, “Embedded Surveillance System Using PIR Sensor”,

International Journal of Advanced Technology in Engineering and Science, www.ijates.com Volume

No.02, Issue No. 03, March 2014.

[4]. Sudhir G. Nikhade, “Wireless Sensor Network System using Raspberry Pi and Zigbee for International

Environmental Monitoring Applications”, 2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, T.N., India. 6 - 8 May 2015. Pp.376-381.

[5]. Pramod P. J, S. V Srikanth, Vivek N, Mahesh U Patil, Sarat Chandra Babu N, “Intelligent Intrusion Detection System (In2DS) using Wireless Sensor Networks”, Proceedings of the 2009 IEEE International Conference on Networking, Sensing and Control, Okayama, Japan, March 26-29, 2009.

# Appendix

## Source code

import json

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

from playsound import playsound

import random

from cloudant.client import Cloudant

from cloudant.error import CloudantException

from cloudant.result import Result, ResultByKey

from ibm\_watson import VisualRecognitionV3

from ibm\_cloud\_sdk\_core.authenticators import IAMAuthenticator

#twillio for sending sms to user

from twilio.rest import Client

account\_sid = 'AC8d25604e8b8c8016030f2f6e3a315241'

auth\_token = 'ff3f493c41df60b00f809af4cf603f14'

FROM\_NUMBER = '+12029329375'

TO\_NUMBER = '+917995868667'

client2 = Client(account\_sid, auth\_token)

#Provide your IBM Watson Device Credentials

organization = "plwr0o"

deviceType = "rsip"

deviceId = "1711"

authMethod = "token"

authToken = "1234567890"

def myCommandCallback(cmd):

print("Command received: %s" % cmd.data)

print(cmd.data['command'])

if(cmd.data['command']=="Servomotoron"):

print("Servo motor on")

if(cmd.data['command']=="Servomotoroff"):

print("Servo motor off")

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()

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 = "JbmF\_lEZF2mWhacw46Ga3S9dAYtBinyX4PHuLJM6q7-\_" # 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/d39ef40c5e3e4377a6e97f47bdc16759:39c97034-7cb0-433c-843b-36d20adae79f::"

authenticator = IAMAuthenticator('AZ9mqzUZP8bIJTl0AzUAuvd7-gyFb1TsbOdAwZNTCO1P')

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/07910fc0-f78a-4e64-b003-d8c8748d5df1')

client = Cloudant("28b096b5-d5c8-4ea1-8ac6-3142ff57803b-bluemix", "5399e404598b8afad48f4e7b6b67bcf3031bb3f201e2979cfb53852596b3c607", url="https://28b096b5-d5c8-4ea1-8ac6-3142ff57803b-bluemix:5399e404598b8afad48f4e7b6b67bcf3031bb3f201e2979cfb53852596b3c607@28b096b5-d5c8-4ea1-8ac6-3142ff57803b-bluemix.cloudantnosqldb.appdomain.cloud")

client.connect()

database\_name = "animal"

picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")

picname=picname+".jpg"

pic=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")

# 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))

cam = cv2.VideoCapture(0)

cv2.namedWindow("Animal")

while True:

ret, frame = cam.read()

animal=0

if not ret:

print("failed to grab frame")

break

cv2.imshow("Animal", frame)

k = cv2.waitKey(1) & 0xFF

if k== ord('q'):

# q pressed

print("q hit, closing...")

break

else:

picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")

picname=picname+".jpg"

pic=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")

cv2.imwrite(picname, frame)

print("{} written!".format(picname))

with open(picname, 'rb') as images\_file:

classes = visual\_recognition.classify(

images\_file=images\_file,

threshold='0.6').get\_result()

print(json.dumps(classes, indent=2))

for i in classes['images'][0]['classifiers'][0]['classes']:

if i['class']=='animal':

animal=1

if animal==1:

print("animal found turning on siren and leds")

playsound('siren.mp3')

print("sending message to user")

message = client2.messages.create(

to=TO\_NUMBER,

from\_=FROM\_NUMBER,

body="Alert! animal has entered into your farm")

print(message.sid)

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

multi\_part\_upload("cloud-object-storage-dsx-cos-standard-60v",picname,pic+".jpg")

if my\_database.exists():

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

json\_document = {

"\_id": pic,

"link":COS\_ENDPOINT+"/cloud-object-storage-dsx-cos-standard-60v/"+picname

}

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

if new\_document.exists():

print("Document '{new\_document}' successfully created.")

else :

print("no animal found")

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

multi\_part\_upload("cloud-object-storage-dsx-cos-standard-60v",picname,"none.jpg")

if my\_database.exists():

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

json\_document = {

"\_id": pic,

"link":COS\_ENDPOINT+"/cloud-object-storage-dsx-cos-standard-60v/"+picname

}

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

if new\_document.exists():

print("Document '{new\_document}' successfully created.")

time.sleep(1)

t=random.randint(1,100)

h=random.randint(1,100)

data = {"d":{ 'temperature' : t, 'humidity': h, 'animal': animal}}

#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

cam.release()

cv2.destroyAllWindows()

deviceCli.disconnect()