1.Introduction:

1.1.Overview:

Intelligent crop protection helps us to whether there's an entry of an animal, and also we can know whether temperature of the atmosphere,humidity in the atmosphere and moisture so that we can manage the texture of the soil.

1.2.Purpose:

Farmers have issues to monitor the farm all the time. So inorder to overcome these issues we have come across this idea where the farmer can know the temperature,humidity and moisture content. Also, he can know whether there is an entry of any kind of animal into the field to protect the crops from them.

2.Literature Survey:INTELLIGENT CROP PROTECTION

2.1 Existing Problem:

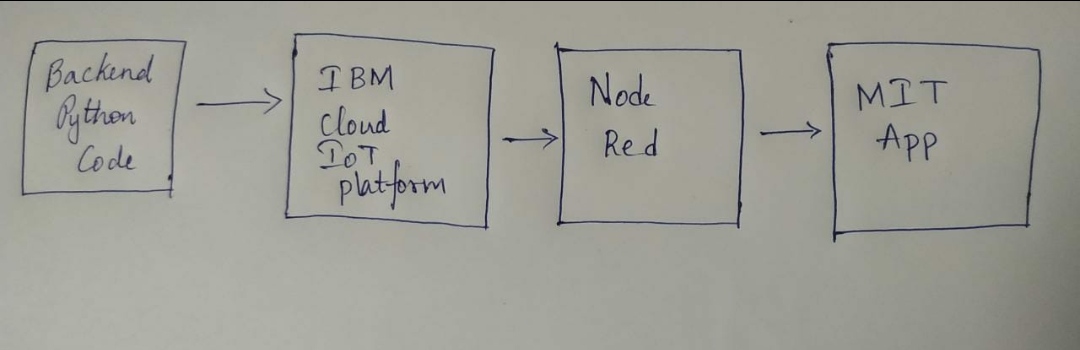
Farmers had numerous complaints about the entry of animals into their farms and the crops are eaten by them.Also the major complaint about temperature, humidity and mositure content where it became difficult to for them to overcome and conquer a good crop.

2.2. Proposed Solution:

Intelligent Crop Protection helps the farmers to overcome the problems they are facing with animals(which certainly ate away most of the crops) , temperature ,humidity and moisture issues. They can have a happy farming and can get a blessed yeild. Thus can have a easy farming with no issues.

3. Theoretical Analysis:

3.1. Block Diagram:

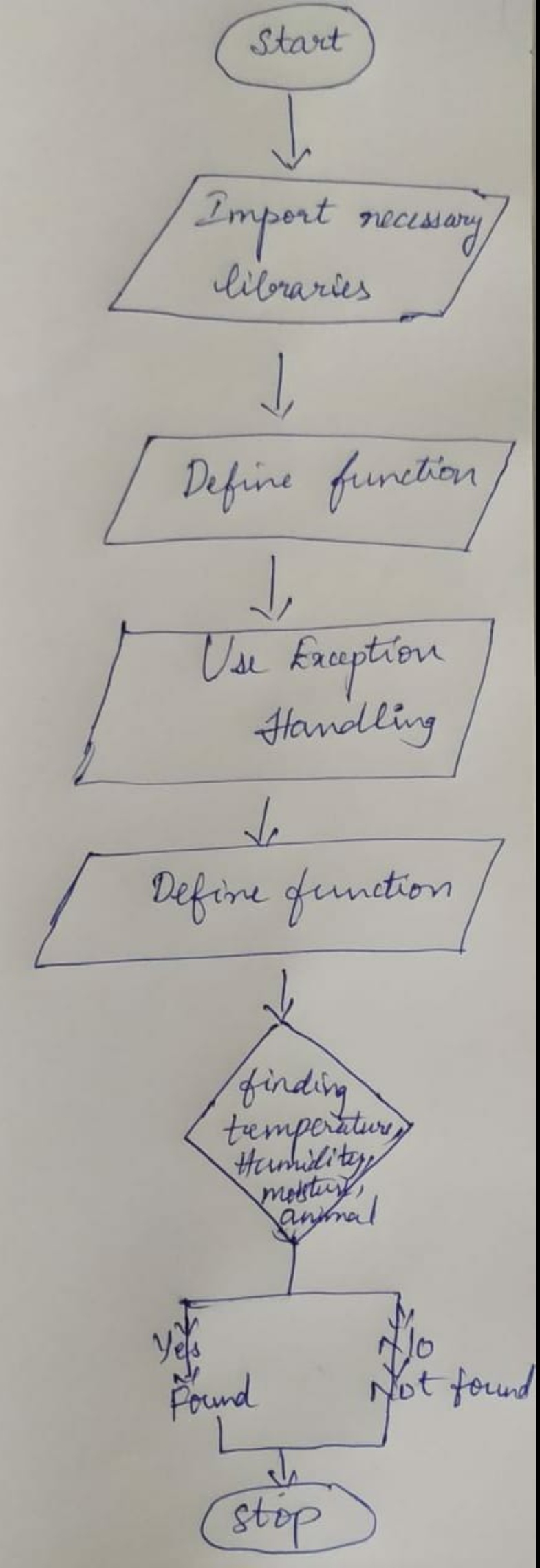


Firstly, the backend code is written in IDLE python 3.8.3. The output information is sent to IBM IoT platform via IBM cloud. This information is sent to Node red and from Node red,it is sent MIT app(the one which we have to create). That app shows us the information about temperature,humidity,moisture and shows whether the any animal is found or not through a camera.

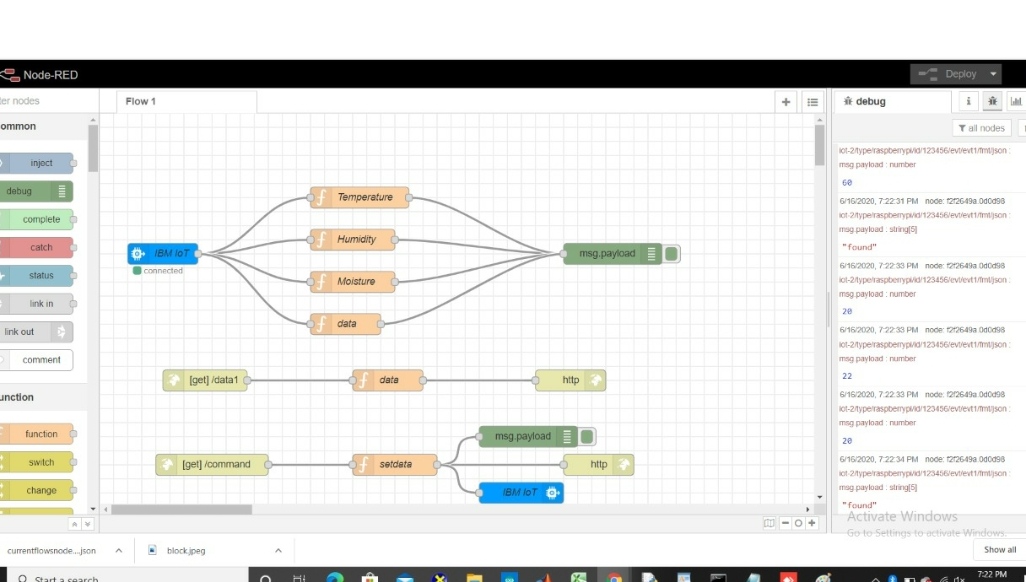
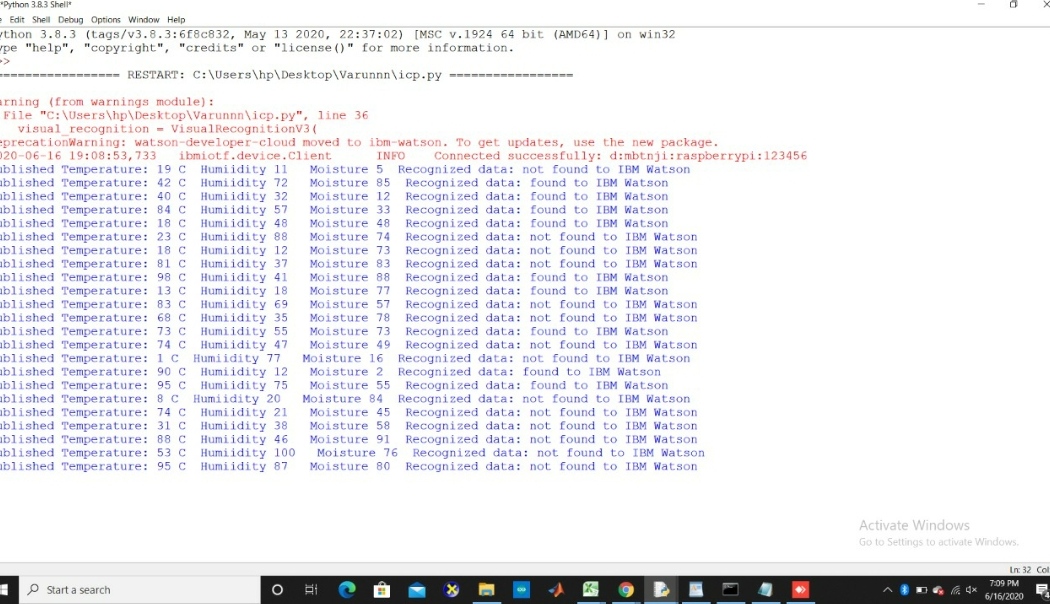
4. Experimental Investigations:

The soil moisture, temperature and humidity are known to us by sensors in the practical applications and we can also know whether if there is an entry of any animal with the help of cameras installed in the locality.

5.Flowchart:



6.Result:



7.1.Advantages:

i. It gives a good crop.

ii. Such things help us advance the technology in every aspect.

7.2.Disadvantages:

i. It is so much difficult to teach every farmer about the technology where the cannot understand in few circumstances, and can neglect it

8. Applications:

It is used to know the pH levels, moisture content, temperature and humidity of the atmosphere.

9. Conclusion:

Thus, technology is not only used in IT sectors but also, in a every sector. Advancement of technology is always a gift.

10. Future Scope:

Intelligent Crop Protection can help us get a good crop. In times of crisis, good crop is essential for any human to survive, so in such times instead of getting a bad crop, we can get a good crop and survive.

Appendix:

import json

from watson\_developer\_cloud import VisualRecognitionV3

import ibmiotf.application

import ibmiotf.device

import time

import sys

import random

import cv2

import numpy as np

import datetime

organization = "mbtnji"

deviceType = "raspberrypi"

deviceId = "123456"

authMethod = "token"

authToken = "12345678"

def myCommandCallback(cmd):

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

if cmd.data['command']=='motoron':

print("motor is ON")

elif cmd.data['command']=='motoroff':

print("motor is OFF")#Commands

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()

visual\_recognition = VisualRecognitionV3(

'2018-03-19',

iam\_apikey='ouINj9jbE4PKtUUAyGUVNNyWuzH5Ho2IXZ\_VlIPjpxmQ')

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times

deviceCli.connect()

# Disconnect the device and application from the cloud

#deviceCli.disconnect()

def vis(ifile):

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

a = visual\_recognition.classify(

images\_file,

threshold='0.6',

classifier\_ids='default').get\_result()

b=a["images"][0]["classifiers"][0]["classes"]

k=[]

for i in b:

k.append(i["class"])

for j in k:

if j=="bird":

a="found"

break

elif j=="animal":

a="found"

break

else:

a="not found"

return a

video=cv2.VideoCapture(0)

while True:

#capture the first frame

check,frame=video.read()

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

cv2.imshow('face detection', frame)

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

cv2.imwrite(picname+".jpg",frame)

f=picname+'.jpg'

a=vis(f)

temp = random.randint(0,100)

hum = random.randint(0,100)

moist = random.randint(0,100)

#Send Temperature & Humidity to IBM Watson

data = { 'Temperature': temp, 'Humidity': hum, 'Moisture': moist, 'data' : a }

#print (data)

def myOnPublishCallback():

print("Published Temperature: %s C " % temp, "Humiidity %s " % hum, "Moisture %s " % moist, "Recognized data: %s" % a, "to IBM Watson")

#time.sleep(5)

success = deviceCli.publishEvent("evt1", "json", data, qos=0, on\_publish=myOnPublishCallback)

if not success:

print("Not connected to IoTF")

deviceCli.commandCallback = myCommandCallback

#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