INTRODUCTION

1.1 Overview

Indian biodiversity is home to different kinds of animals.These animals enter urban and rural areas in search of food.Human animal conflict is a major problem where enormous amount of resources is lost and human life is in danger.Most of the agriculture activities take place in rural areas.Due to this people lose their crops, livestock, property, and sometimes lives too.Due to fear of wild animals, people attack and kill the animals .So this zone is to be monitored continuously to prevent entry of wild animals.With regard to this problem, we have made an effort to develop the system which can easily identify the entering of the animal in that regional limit through the cameras surrounded in that region.These cameras captures the images in certain intervals and then send it for visual recognition.When these images are visually recognized and if there is any presence of wild animal,it will give an alert to the forest department authorities.The live streaming of those cameras will be presented in the forest office.Whenever a animal is detected , voice commands will be sent to the villagers inorder to alert them and at the same time notifications will be sent to forest department through messages.Captured image of the wild animal will also be sent to the forest officials and local people.

1.2 Purpose

Animal warning systems are very applicable to rural areas as these areas often have a lower population.Most of the rural areas have forest attached to them.Human animal conflict arises due to the enroachment and poaching, humans move into the forest to satisfy their livelihood,and claiming of land for agricultural activities and rapid industrialization causes spreading of urban ground and animals enter the nearby areas in search of food and water bodies.These animals destroy the resources and sometimes it also leads to human harm or human death.Out of fear, human beings harm the animals and sometimes kill them too.In recent times the numbers of these kinds of conflicts are increasing. So there should be a system to encounter the wild animals entering into the zones of humans so that the humans and the authorities can be provided with an alerting system to ensure the safety of both humans and animals .So,we came out with an idea of Animal Detection and Alerting System which will detect the entering on the animal in the regional limit through the camera and give alert to the forest department through the messages and alerting the villagers through the voice commands.It will help the forest officers to reach the spot at the right time and rescue the animals.Image of the detected animal will be sent to forest authorities and people

LITERATURE SURVEY

2.1 Existing problem

- CROP DAMAGE is one of the problem faced by farmers due to the entering of wild animals.These animals damage food and crash crops and they also affect the livelihoods of the farmers.Elephants in large groups can destroy large areas of crops in a single night.
- DAMAGE TO OTHER PROPERTY is another problem faced by the people.wild animals damage the crops which are stored after the harvest.The loss of this stored food is considered to be more disruptive to the farmers because these are the food which are cncentrated at a larger space.
- HUMAN DEATH AND INJURY are caused by the wild animals when they enter into field or any area.They attack on humans while farmers where trying to protect their crops from animals.

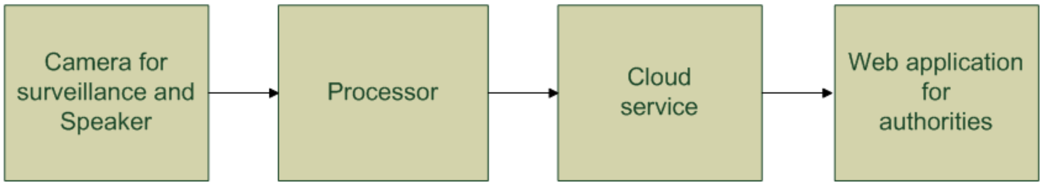
Due to this fear,people start to kill the animals as soon as they enter the village which starts a conflict between the human and wild life.Human-wildlife conflict is a global issue present in urban and rural areas.On the other hand it also becomes a threat posed by wildlife on human life,economic security ,health and property.It also disturbs the ecosystem structure and functions.Due to the conflict between human and wildlife , many species have got extincted .

2.2 Proposed solution

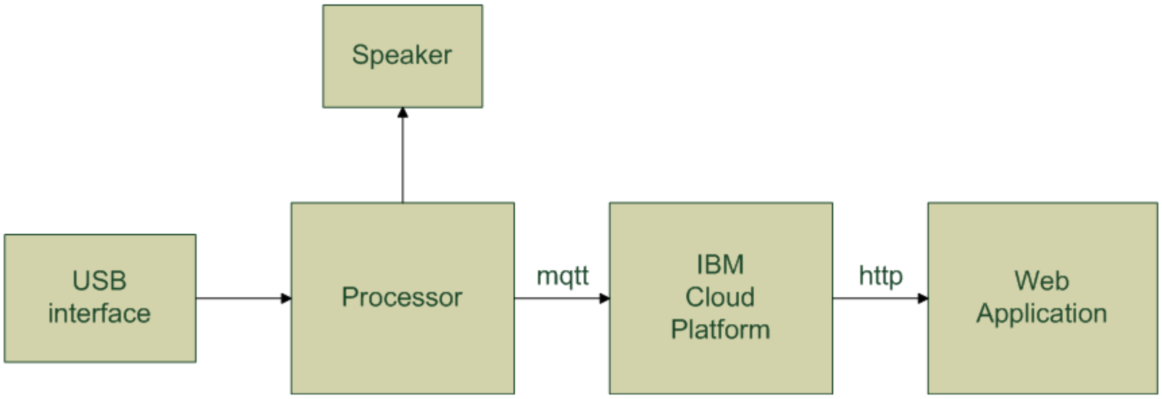
In order to solve the problem of animal attack , destruction of crops and property in the rural areas we came up with the idea of Animal Detection and Alerting System .Urban and Rural areas are surrounded by cameras in the regional limit.Live streaming of these cameras will be presented in the forest office.These cameras capture the images in the certain interval of time .These images will be directed to Visual Recognition.Whenever a animal is detected , it sends an alert to the forest authorities through messages .At the same time it will send a voice command to the villagers alerting about the entering of animals to their regional limit.Through this alerting system,it will be easy task for the forest authorities to get hold of the animal .Due to the alerting voice commands to the people of that region, it will create a awareness among the villagers about the entering of the animal.It will restrict them from going to that area.It will be very helpfull for the authorities to track the location of the wild animal.An important aspect of the work is that it benfits both the animals and local human communities, and actively involves these communities.This is about finding solutions that lead to mutually beneficial co-existence.

THEORETICAL ANALYSIS

3.1 Block diagram


Camera for surveillance and Speaker
Processor
Cloud service
Web application for authorities

3.2 Hardware/Software designing

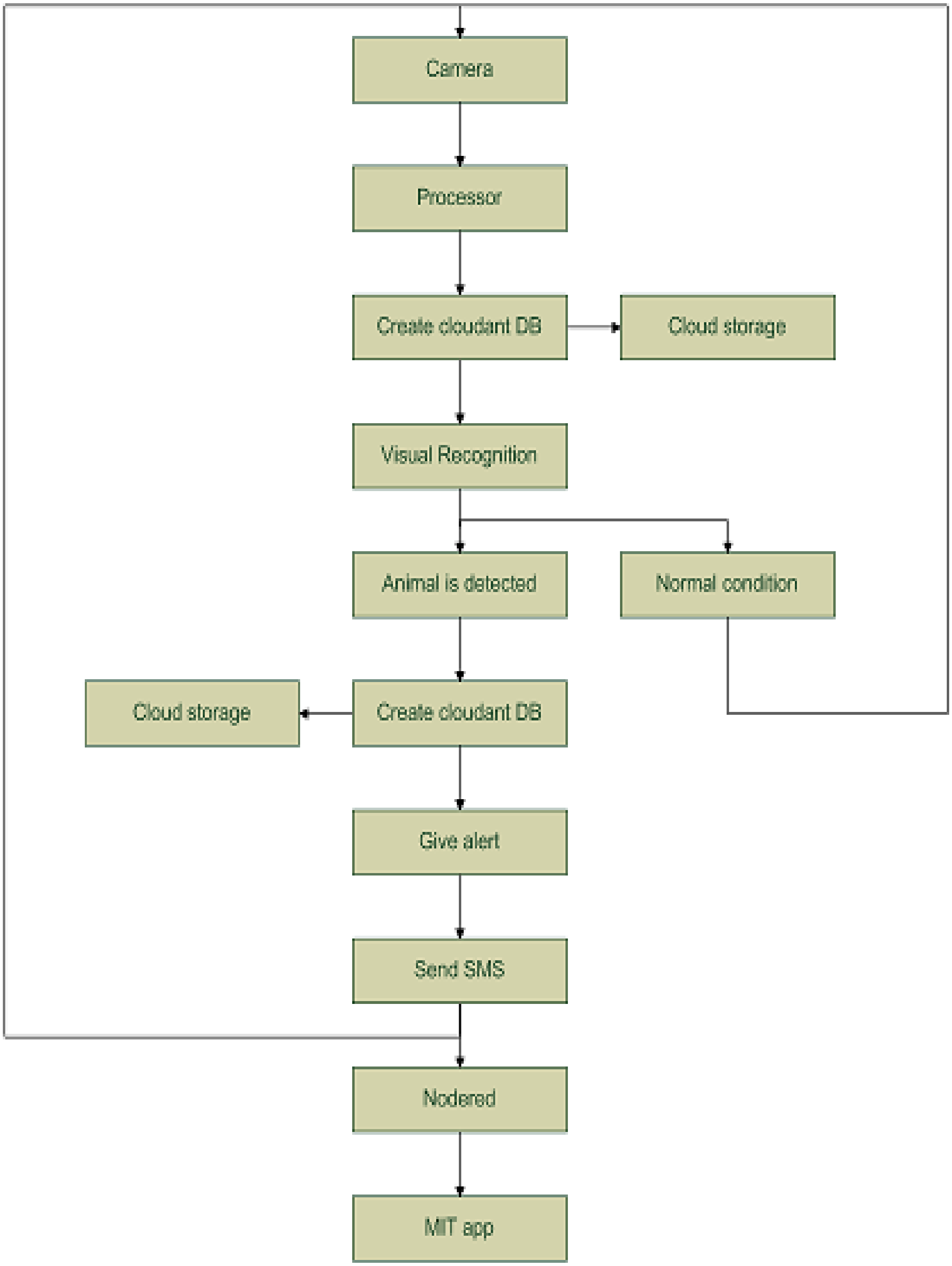


EXPERIMENTAL INVESTIGATIONS

- When we run the code , we used to get index error and the visual recognition was unable to detect the presence of animal infront of the camera. when we went deep into that, we got to know that , we got to know that the animal recognition is possible only when we keep the threshold under a certain range.When we decreased the threshold from 9 to 7, the visual recognition took place correctly and it was able to detect the presence of animal entering the regional limit.

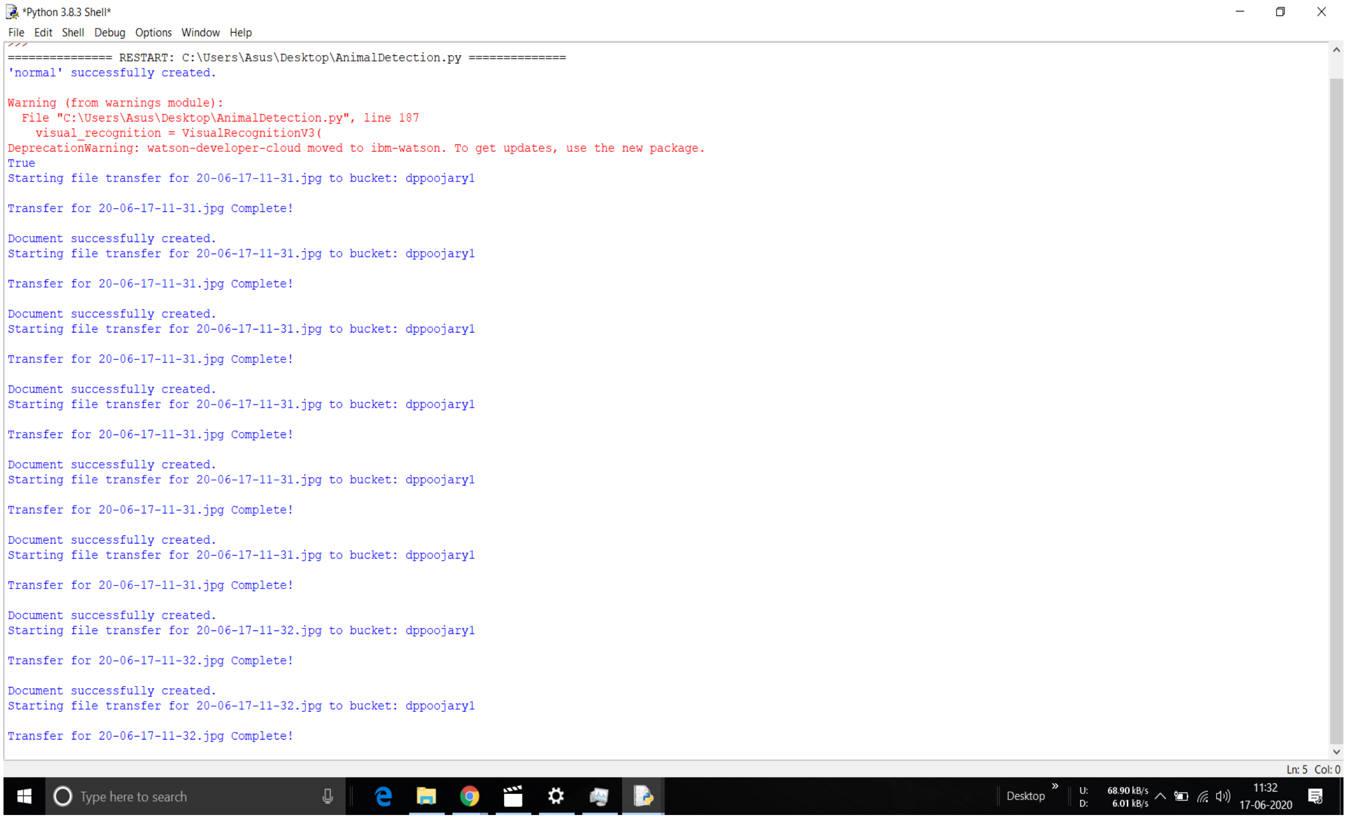
- When we used to run the code, live streaming was working properly and the bucket creation was taking place correctly but it was unable to detect the presence of animal infront of the camera.We thought that there is something wrong in the code but the thinng was that since there was some dust and fog in my webcam ,it wasn't able to detect the presence of animal.

FLOWCHART

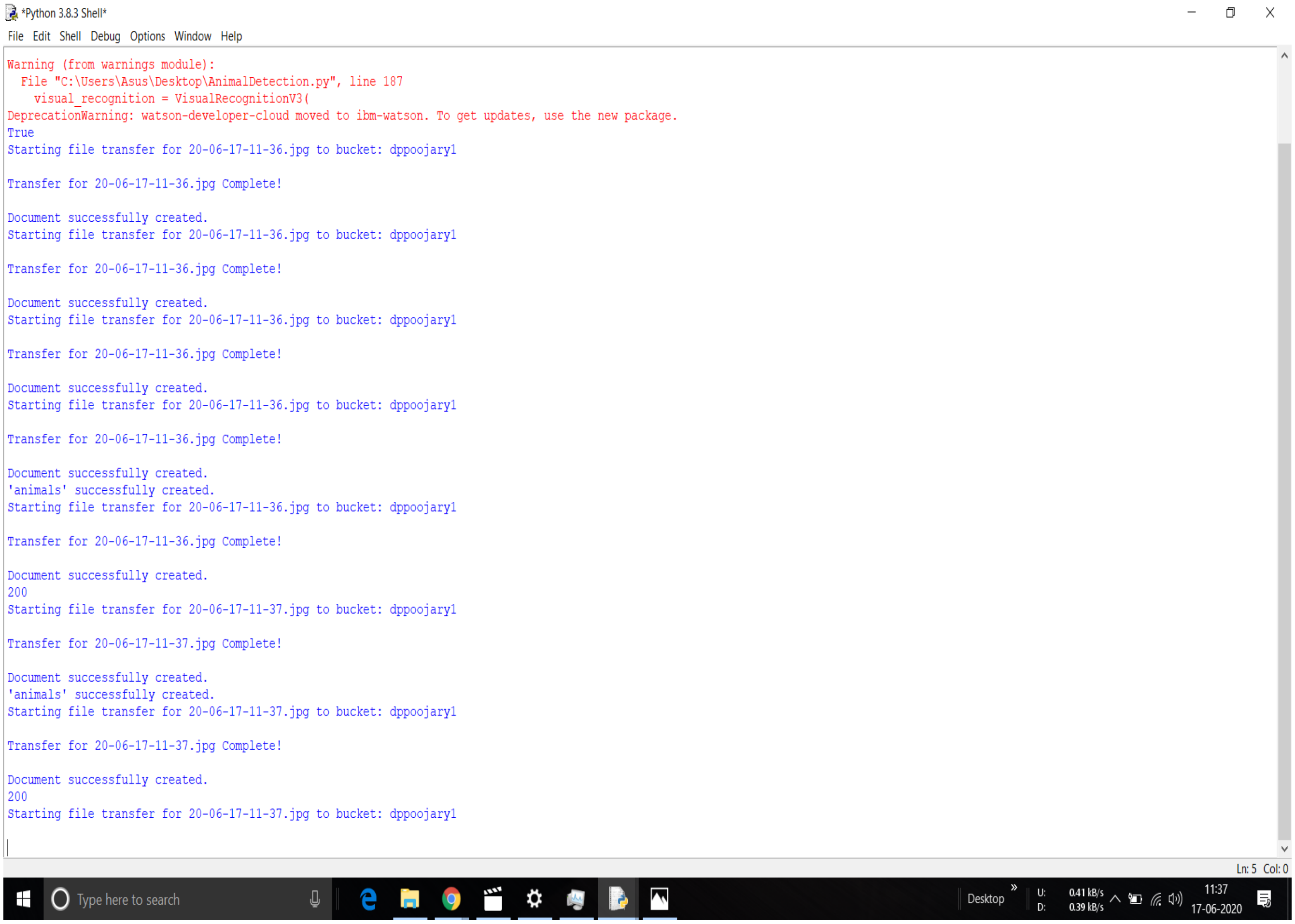

Camera
Processor
Create cloudant DB
Cloud storage
Visual Recognition
Animal is detected
Normal condition
Cloud storage
Create cloudant DB
Give alert
Send SMS
Nodered
MIT app

RESULT

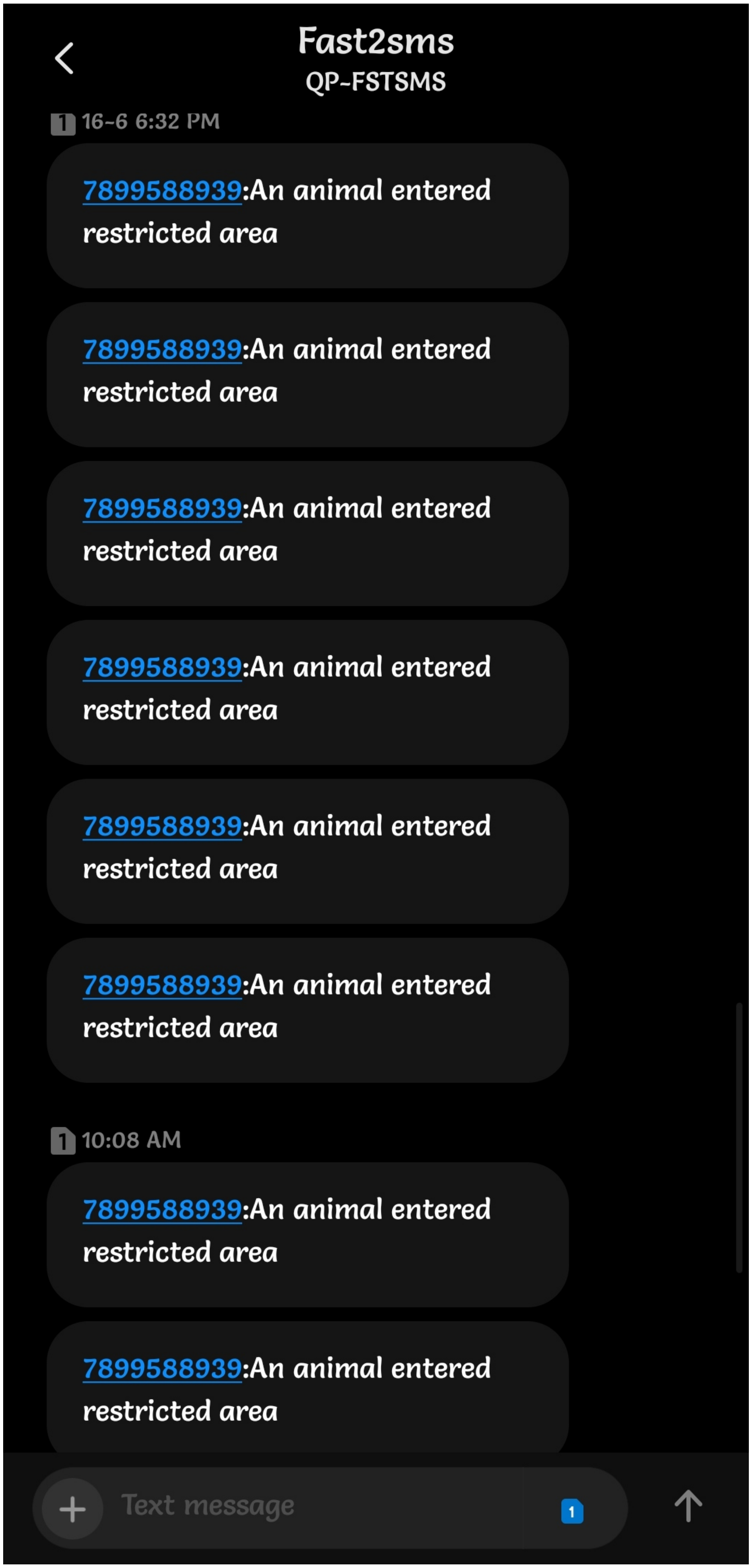
- The installed cameras will be under survilliance 24x7
- The live streaming of these cameras are presented in the forest office
- The installed cameras capture images of the area in certain time intervals
- The captured images are stored in cloudant db through the communication protocols
- The stored images are sent to Visual Recognition
- During visual recognition , if it is recognised as an animal then an alert notification along with the image is sent to the concerned authorities through node-red
- An alert notification is also sent to the local people with the captured image through MIT application



```
>>>
============== RESTART: C:\Users\Asus\Desktop\AnimalDetection.py =============
'normal' successfully created.

Warning (from warnings module):
  File "C:\Users\Asus\Desktop\AnimalDetection.py", line 187
    visual_recognition = VisualRecognitionV3(
DeprecationWarning: watson-developer-cloud moved to ibm-watson. To get updates, use the new package.
True
Starting file transfer for 20-06-17-11-31.jpg to bucket: dppoojary1

Transfer for 20-06-17-11-31.jpg Complete!

Document successfully created.
Starting file transfer for 20-06-17-11-31.jpg to bucket: dppoojary1

Transfer for 20-06-17-11-31.jpg Complete!

Document successfully created.
Starting file transfer for 20-06-17-11-31.jpg to bucket: dppoojary1

Transfer for 20-06-17-11-31.jpg Complete!

Document successfully created.
Starting file transfer for 20-06-17-11-31.jpg to bucket: dppoojary1

Transfer for 20-06-17-11-31.jpg Complete!

Document successfully created.
Starting file transfer for 20-06-17-11-31.jpg to bucket: dppoojary1

Transfer for 20-06-17-11-31.jpg Complete!

Document successfully created.
Starting file transfer for 20-06-17-11-31.jpg to bucket: dppoojary1

Transfer for 20-06-17-11-31.jpg Complete!

Document successfully created.
Starting file transfer for 20-06-17-11-32.jpg to bucket: dppoojary1

Transfer for 20-06-17-11-32.jpg Complete!

Document successfully created.
Starting file transfer for 20-06-17-11-32.jpg to bucket: dppoojary1

Transfer for 20-06-17-11-32.jpg Complete!
```



```
Warning (from warnings module):
  File "C:\Users\Asus\Desktop\AnimalDetection.py", line 187
    visual_recognition = VisualRecognitionV3(
DeprecationWarning: watson-developer-cloud moved to ibm-watson. To get updates, use the new package.
True
Starting file transfer for 20-06-17-11-36.jpg to bucket: dppoojary1

Transfer for 20-06-17-11-36.jpg Complete!

Document successfully created.
Starting file transfer for 20-06-17-11-36.jpg to bucket: dppoojary1

Transfer for 20-06-17-11-36.jpg Complete!

Document successfully created.
Starting file transfer for 20-06-17-11-36.jpg to bucket: dppoojary1

Transfer for 20-06-17-11-36.jpg Complete!

Document successfully created.
Starting file transfer for 20-06-17-11-36.jpg to bucket: dppoojary1

Transfer for 20-06-17-11-36.jpg Complete!

Document successfully created.
'animals' successfully created.
Starting file transfer for 20-06-17-11-36.jpg to bucket: dppoojary1

Transfer for 20-06-17-11-36.jpg Complete!

Document successfully created.
200
Starting file transfer for 20-06-17-11-37.jpg to bucket: dppoojary1

Transfer for 20-06-17-11-37.jpg Complete!

Document successfully created.
'animals' successfully created.
Starting file transfer for 20-06-17-11-37.jpg to bucket: dppoojary1

Transfer for 20-06-17-11-37.jpg Complete!

Document successfully created.
200
Starting file transfer for 20-06-17-11-37.jpg to bucket: dppoojary1
```



Fast2sms
QP-FSTSMS

1 16-6 6:32 PM

7899588939:An animal entered restricted area

7899588939:An animal entered restricted area

7899588939:An animal entered restricted area

7899588939:An animal entered restricted area

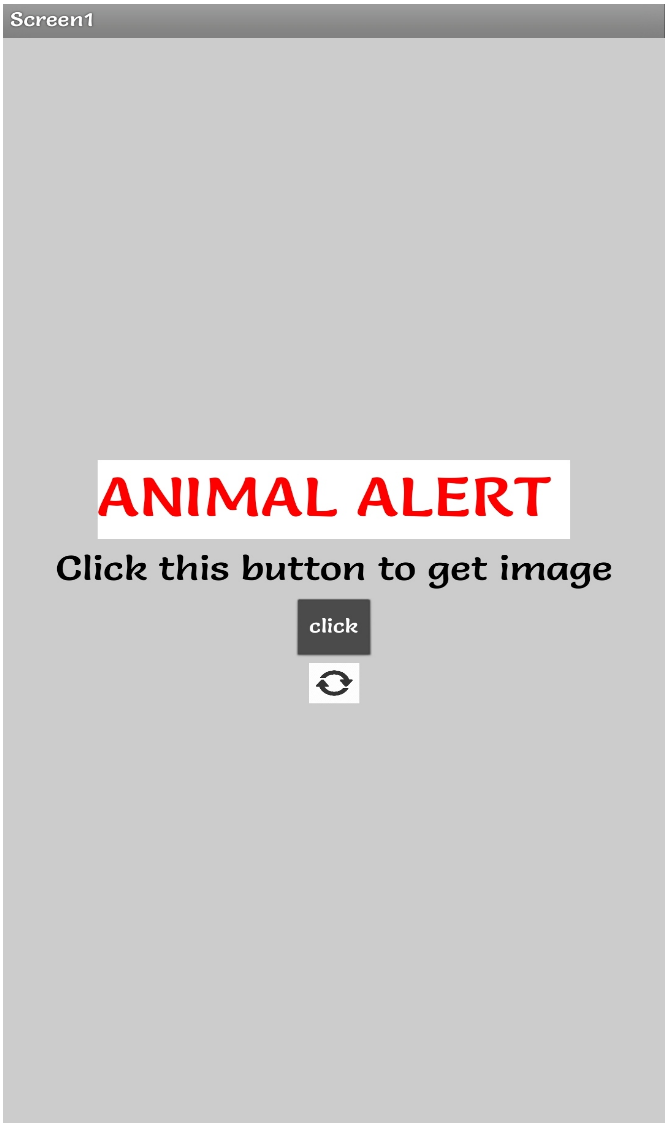
7899588939:An animal entered restricted area

7899588939:An animal entered restricted area

1 10:08 AM

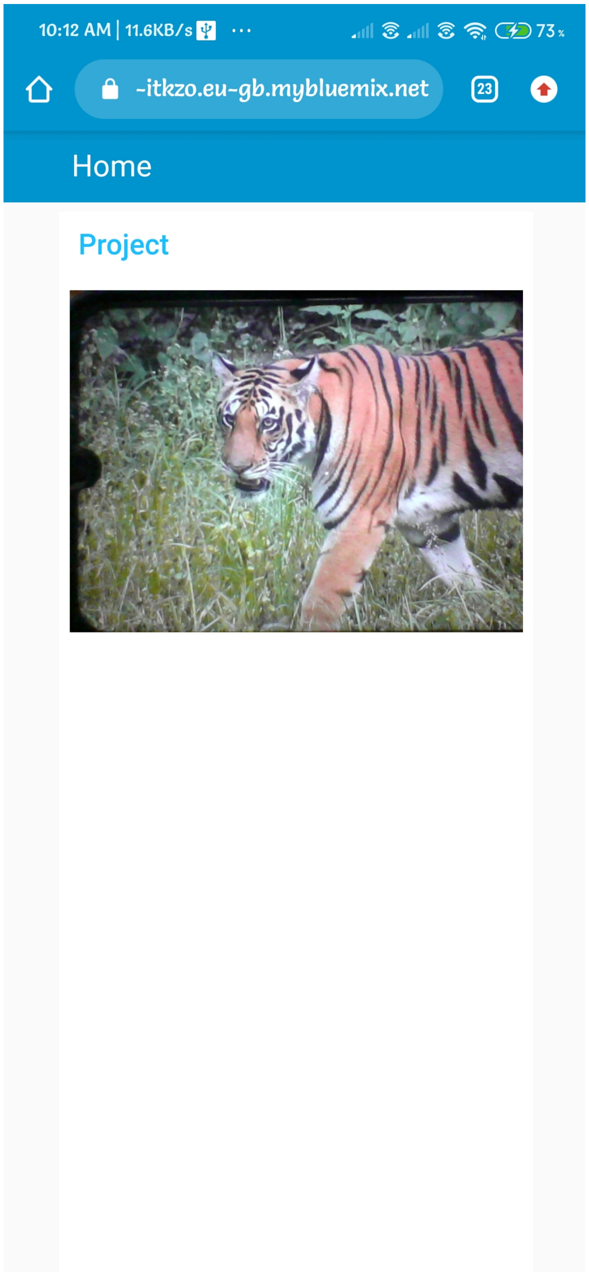
7899588939:An animal entered restricted area

7899588939:An animal entered restricted area



ANIMAL ALERT

Click this button to get image


10:12 AM | 11.6KB/s
73%
-itkzo.eu-gb.mybluemix.net
23
Home
Project

ADVANTAGES AND DISADVANTAGES

The advantages of animal detection and alerting system are:

- It is easy to work(monitor) with security camera as long as we have power supply source closeby
- It will be easy to gather the evidences
- Monitoring is very useful in maintaining a peaceful environment
- As an alert notification system tool, the the information will reach the people in an instant,thus the recipient will be able to take the necessory action
- You can always get instantly updated.The information will be reached directly to each individual
- Can access the data anywhere,anytime.This allows for easy collaborationand sharing among users in multiple locations
- Cloud computing delivers faster and more accurate retrievals of appliactions and data.With less downtime, it is the most efficient recovery plan
- To reduce the destruction of crops, property and livestock made my the wild animals
- This system is harmless and doesn't injure animals and human beings in any way

The disadvantages are as follows:

- The system must be able to operate continuously with fog that occurs occasionally
- The system can be costly,cost of the set up depends on features and number of cameras and montoring system we buy
- Most of the monitoring applications are costly.Those with cheaper rates don't work as per the needs, but those that are perfect and outstanding in delivering the desired result are expensive.
- We should have proper network connection in region under conservation, else it will be hard to reach the people during emergency
- Notification also need to be connected to a data source at all times, so they can easily waste battery life on phones and drain mobile plans
- Cloud technologies always prone to an outage and other technical issues.Even , the best cloud service provider companies may face this type of trouble despite maintaning high standards of maintenance
- Downtime should be considered while working with cloud computing.That's because your cloud provider may face power loss, low internet connectively,service maintenance etc

APPLICATIONS

- With video capabilities, security systems include cameras that need to process data and transform it into information that can help us identify threats and prevent them from turning into reality
- We might even collect information about the species with a granularity unseen before
- It will help in the prevention of illigal wildlife crime
- This system helps us to keep away wild animals from the farmlands
- Using this system it is possible to do animal migration traking
- This system can be used to conduct the census of wild animals

CONCLUSION

Human animal conflict has become a major problem where enormous amount of resources are lost and both the human life and wildlife is in danger.Animal detection and alerting systems helps to detect the entering of wild animals into certain regions.The main aim here is to protect both the wildlife and humans.when any animal is detected, it alerts the concerned authorities about the danger through the notification.At the same time it sends an voice command to the villagers alerting them about the danger.It helps the concerned authorities to reach the spot in a very less time and it also alerts the villagers to stay at home and be safe.So there is no point of harm to both wild animal and humans.

FUTURE SCOPE

- In addition to providing protection this system differentiates between human and a wild animal using visual recognition system
- We used classifier to distinguish between wild animal and humans
- The animals can also be detected during night time by using night vision cameras
- We have to overcome the problem of fog during certain seasons
- We have to overcome the high maintanence cost,low internet connectivity and cost of cameras

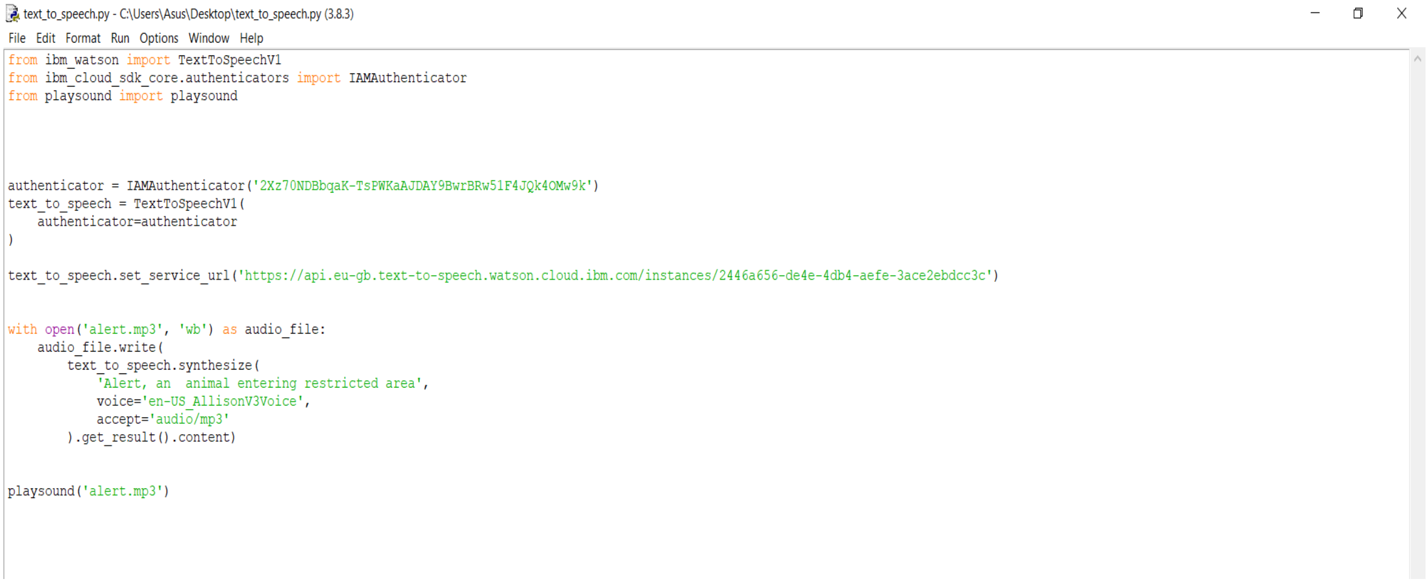
BIBILOGRAPHY

- Kenneth A. Lambert,The Fundamentals of Python :First Programs,2011,Cengage Learning
- Charles Dierbach,"Introduction to Computer Science Using Python",1st Edition,Wiley India Pvt Ltd.
- Reema Thareja,"Python Programming Using Problem Solving Approach",Oxford University Press,2017
- Internet Of Things, Systems and Applications by Jamil Y.Khan
- Getting Started With Enterprise Internet of Things:Design Approaches and Software Architecture Models

APPENDIX

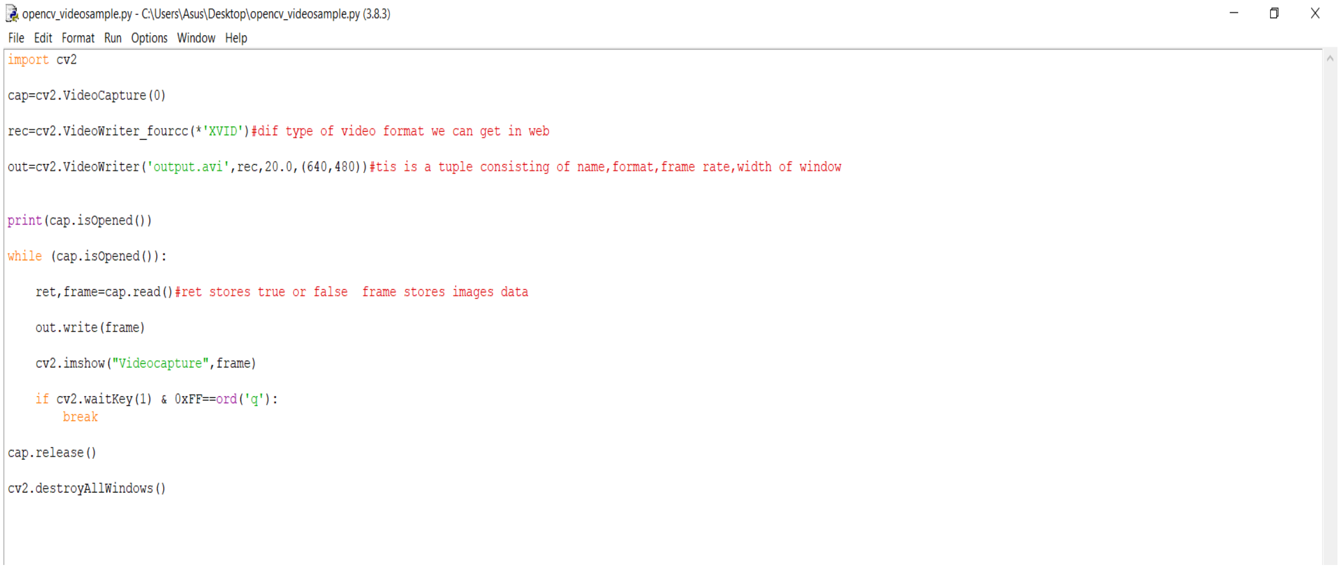
A.Source code

Text To Speech


text_to_speech.py - C:\Users\Asus\Desktop\text_to_speech.py (3.8.3)
File Edit Format Run Options Window Help

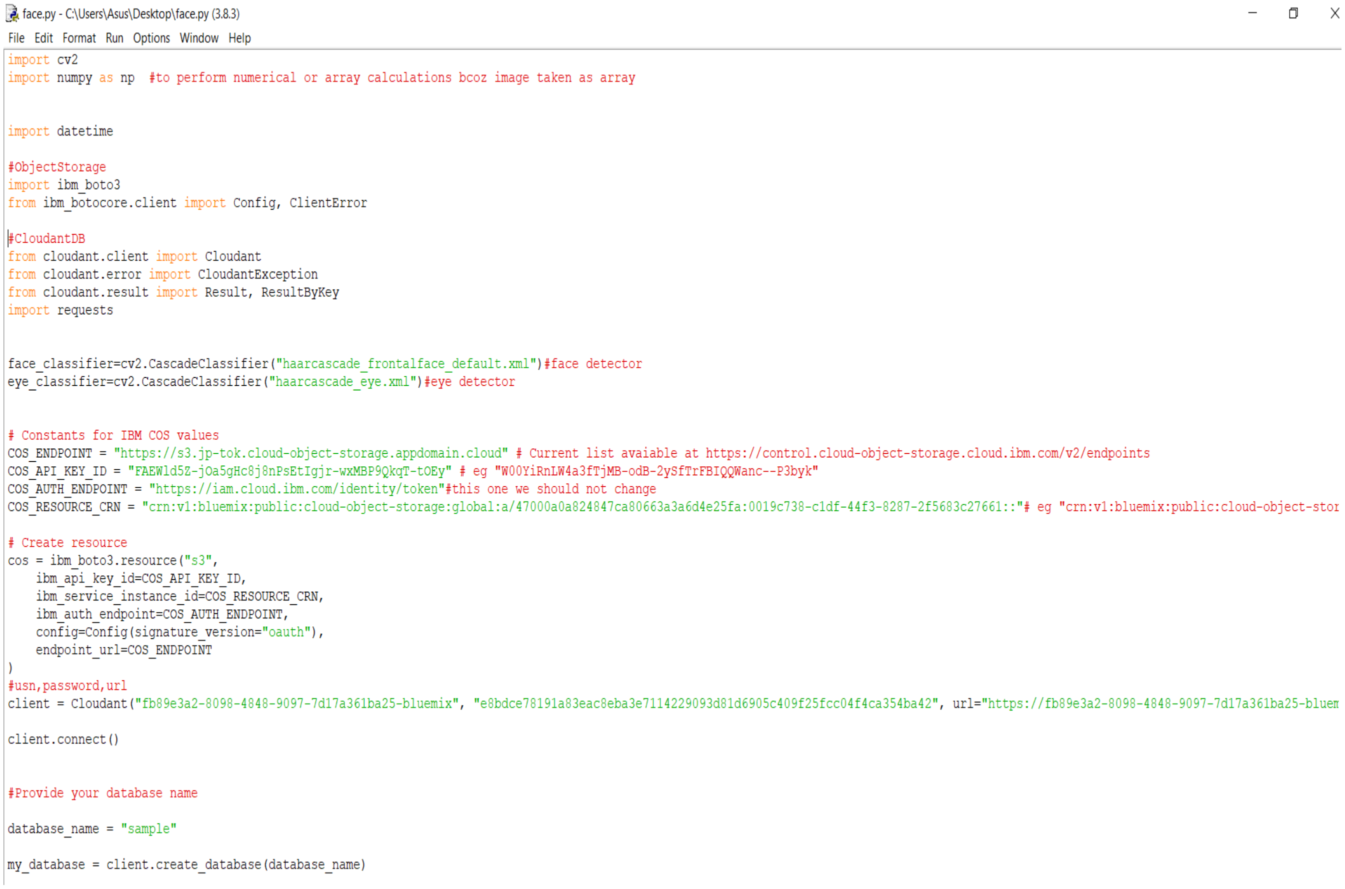
```
from ibm_watson import TextToSpeechV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
from playsound import playsound




authenticator = IAMAuthenticator('2Xz70NDBbqaK-TsPWKaAJDAY9BwrBRw51F4JQk4OMw9k')
text_to_speech = TextToSpeechV1(
    authenticator=authenticator
)

text_to_speech.set_service_url('https://api.eu-gb.text-to-speech.watson.cloud.ibm.com/instances/2446a656-de4e-4db4-aefe-3ace2ebdcc3c')


with open('alert.mp3', 'wb') as audio_file:
    audio_file.write(
        text_to_speech.synthesize(
            'Alert, an  animal entering restricted area',
            voice='en-US_AllisonV3Voice',
            accept='audio/mp3'
        ).get_result().content)


playsound('alert.mp3')
```

Open CV

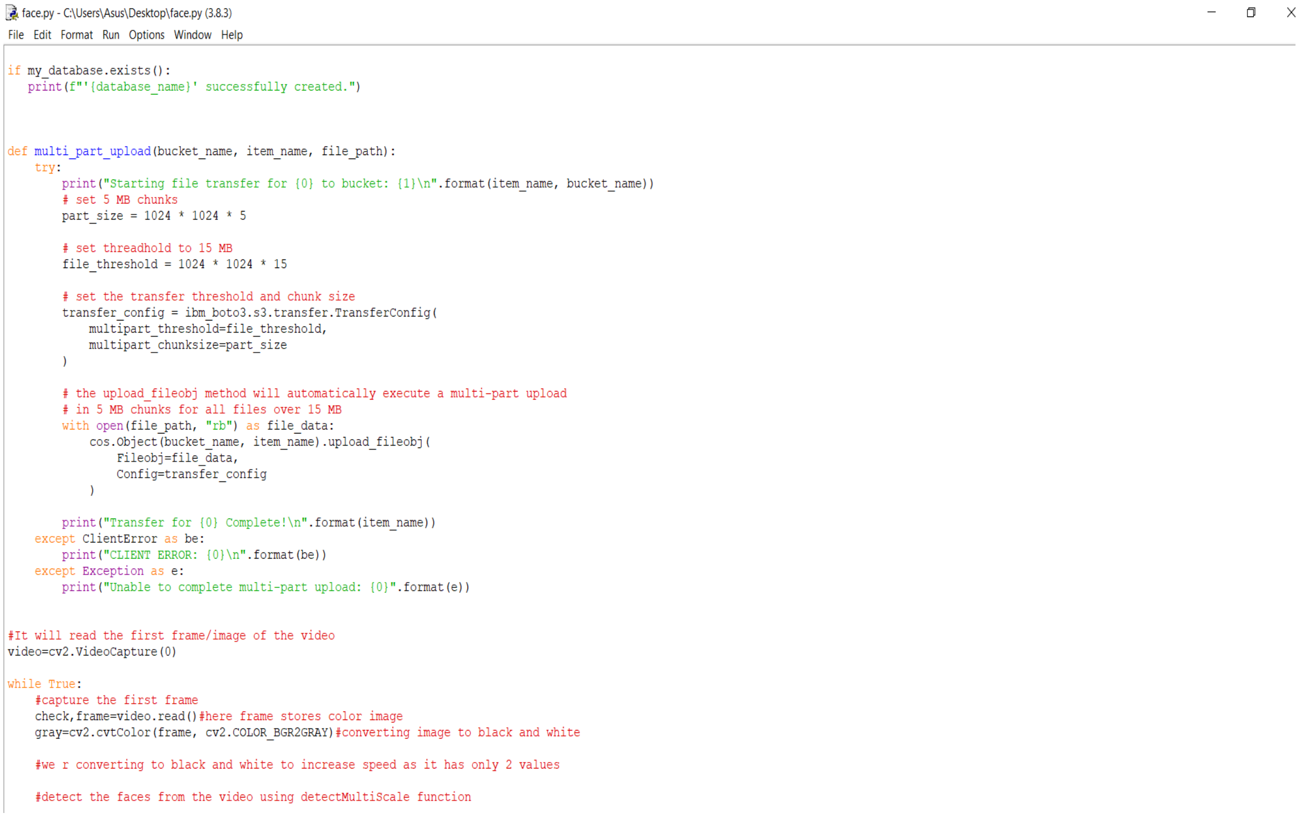

opencv_videosample.py - C:\Users\Asus\Desktop\opencv_videosample.py (3.8.3)
File Edit Format Run Options Window Help

```
import cv2

cap=cv2.VideoCapture(0)

rec=cv2.VideoWriter_fourcc(*'XVID')#dif type of video format we can get in web

out=cv2.VideoWriter('output.avi',rec,20.0,(640,480))#tis is a tuple consisting of name,format,frame rate,width of window


print(cap.isOpened())

while (cap.isOpened()):

    ret,frame=cap.read()#ret stores true or false  frame stores images data

    out.write(frame)

    cv2.imshow("Videocapture",frame)

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

cap.release()

cv2.destroyAllWindows()
```

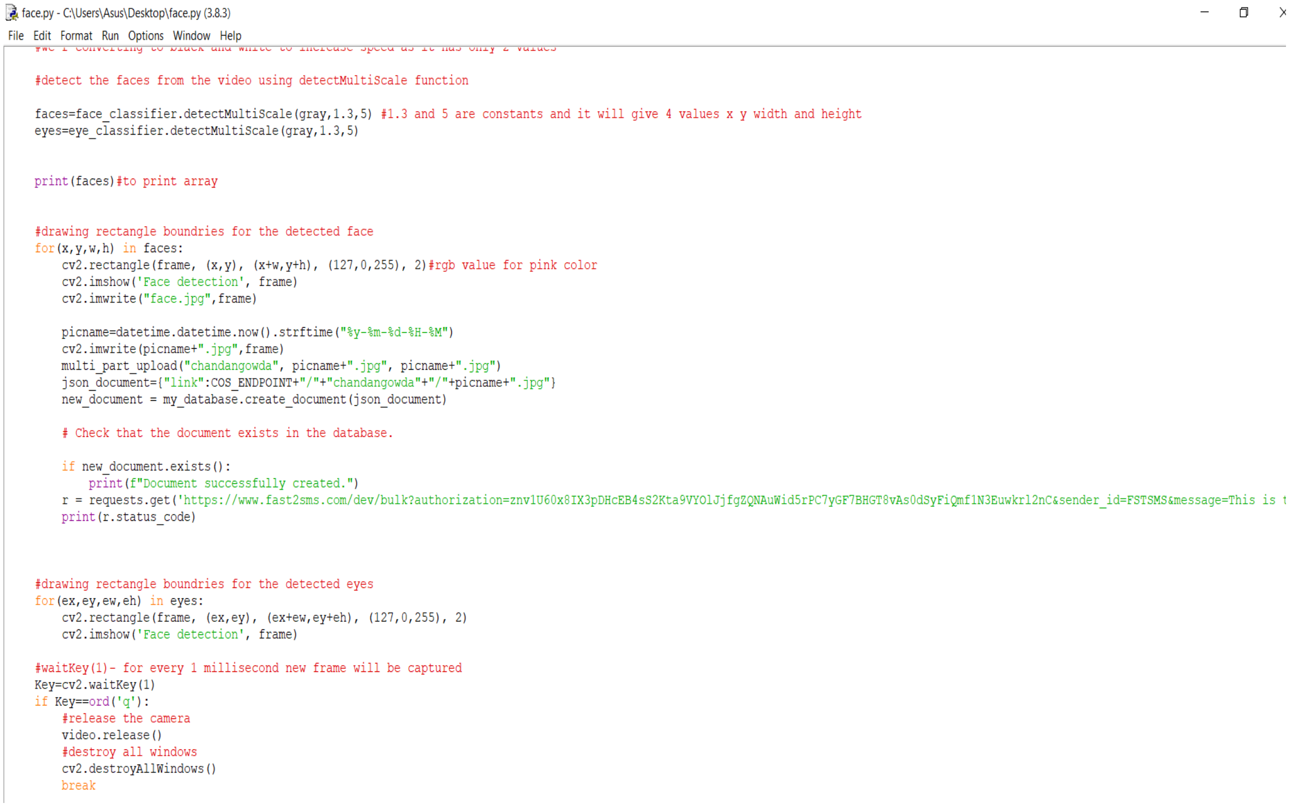
Animal detection and alerting system


face.py - C:\Users\Asus\Desktop\face.py (3.8.3)
File Edit Format Run Options Window Help

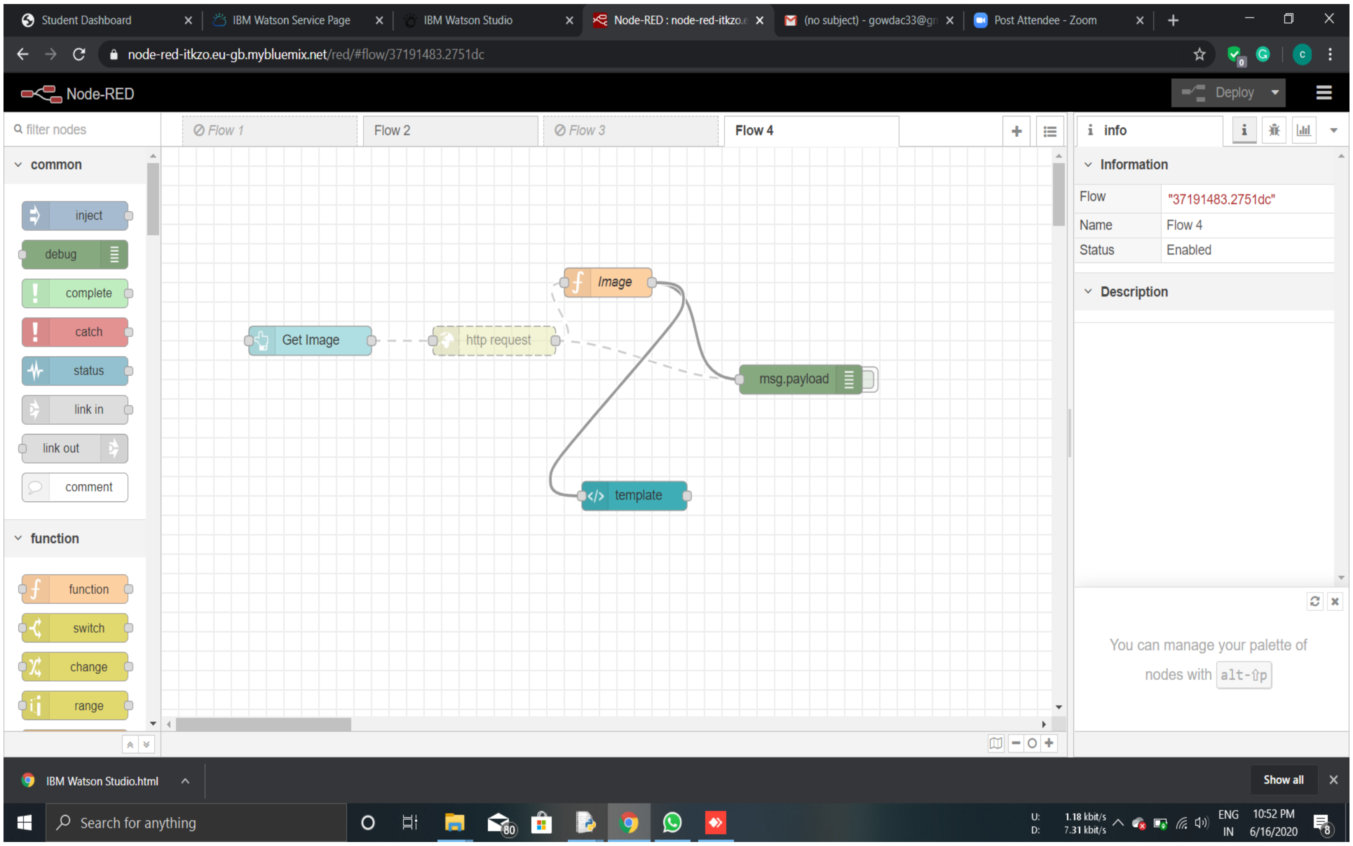
```
import cv2
import numpy as np  #to perform numerical or array calculations bcoz image taken as array


import datetime

#ObjectStorage
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


face_classifier=cv2.CascadeClassifier("haarcascade_frontalface_default.xml")#face detector
eye_classifier=cv2.CascadeClassifier("haarcascade_eye.xml")#eye detector


# 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 = "FAEWld5Z-jOa5gHc8j8nPsEtIgjr-wxMBP9QkqT-tOEy" # eg "W00YiRnLW4a3fTjMB-odB-2ySfTrFBIQQWanc--P3byk"
COS_AUTH_ENDPOINT = "https://iam.cloud.ibm.com/identity/token"#this one we should not change
COS_RESOURCE_CRN = "crn:v1:bluemix:public:cloud-object-storage:global:a/47000a0a824847ca80663a3a6d4e25fa:0019c738-c1df-44f3-8287-2f5683c27661::"# eg "crn:v1:bluemix:public:cloud-object-stor

# 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
)
#usn,password,url
client = Cloudant("fb89e3a2-8098-4848-9097-7d17a361ba25-bluemix", "e8bdce78191a83eac8eba3e7114229093d81d6905c409f25fcc04f4ca354ba42", url="https://fb89e3a2-8098-4848-9097-7d17a361ba25-bluem

client.connect()


#Provide your database name

database_name = "sample"

my_database = client.create_database(database_name)
```


face.py - C:\Users\Asus\Desktop\face.py (3.8.3)
File Edit Format Run Options Window Help

```
if my_database.exists():
   print(f"'{database_name}' successfully created.")



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))


#It will read the first frame/image of the video
video=cv2.VideoCapture(0)

while True:
    #capture the first frame
    check,frame=video.read()#here frame stores color image
    gray=cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)#converting image to black and white

    #we r converting to black and white to increase speed as it has only 2 values

    #detect the faces from the video using detectMultiScale function
```


face.py - C:\Users\Asus\Desktop\face.py (3.8.3)
File Edit Format Run Options Window Help

```
    #we r converting to black and white to increase speed as it has only 2 values

    #detect the faces from the video using detectMultiScale function

    faces=face_classifier.detectMultiScale(gray,1.3,5) #1.3 and 5 are constants and it will give 4 values x y width and height
    eyes=eye_classifier.detectMultiScale(gray,1.3,5)


    print(faces)#to print array


    #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)#rgb value for pink color
        cv2.imshow('Face detection', frame)
        cv2.imwrite("face.jpg",frame)

        picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
        cv2.imwrite(picname+".jpg",frame)
        multi_part_upload("chandangowda", picname+".jpg", picname+".jpg")
        json_document={"link":COS_ENDPOINT+"/"+"chandangowda"+"/"+picname+".jpg"}
        new_document = my_database.create_document(json_document)

        # 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=znv1U60x8IX3pDHcEB4sS2Kta9VYOlJjfgZQNAuWid5rPC7yGF7BHGT8vAs0dSyFiQmf1N3Euwkrl2nC&sender_id=FSTSMS&message=This is t
        print(r.status_code)



    #drawing rectangle boundries for the detected eyes
    for(ex,ey,ew,eh) in eyes:
        cv2.rectangle(frame, (ex,ey), (ex+ew,ey+eh), (127,0,255), 2)
        cv2.imshow('Face detection', frame)

    #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
```



Information

Flow	"37191483.2751dc"
Name	Flow 4
Status	Enabled

Description

You can manage your palette of nodes with alt-⇧p