WILD ANIMAL DETECTION AND ALERT SYSTEM

A PROJECT REPORT

Submitted by

CHITRANKSHI RATHOUR –19BCE10419
JAISHREE SHARMA – 19BCE10307
DIVYAM KHARE- 19BCG10013
ISHITA JAIN - 19BEC1422

TABLE OF CONTENTS

INTRODUCTION

1.1 Overview

The Internet of Things is transforming our physical world into a complex and dynamic system of connected devices on an unprecedented scale.
Advances in technology are making possible a more widespread adoption of IoT, from pill-shaped micro-cameras that can pinpoint thousands of images within the body, to smart sensors that can assess crop conditions on a farm, to the smart home devices that are becoming increasingly popular.

Examining wild animals in their natural environment is an essential task in ecosystem. Due to the enormous growth in human inhabitants and the increase in hunt of economic development makes excessive exploitation of natural resources, fast, innovative and significant changes in the Earth's ecosystems. An expanding region of the land surface has been changed by human activity, modifying natural life populace, habitat and behavior. More fatally, many wild animals on the Earth have disappeared, and many species are locomoted into new places where they can disturb all natural and human resources. Other existing techniques are there but are not effective due to several reasons.

In this paper, we proposed a computer vision-based solution for agriculture security from wild animals. Detecting and tracking wild animals near farm fields is done with the help of camera placed in fields and the information is sent to respective authorities including the forest department and this helps the respective authorities to take immediate actions to protect farm fields from wild animals and vice versa.

1.2 Purpose

The purpose of this project is to detect any wild animal in the range of the camera and intimate immediately to the owner or the services.
There will be a camera integrated in the premises. In the live video streaming the images will be capture at some time intervals and given as an input to clarifai service. If any wild animals are detected the images will be captured and sent to IBM COS. Authorities can see the images through web app. The people living nearby areas will be intimated. If any animal is detected authorities will be notified.

LITERATURE SURVEY

Existing Problem:

One of the tasks is analysis and interpretation of visual scenes by computers. Visual scene analysis is a high-level task that acquire knowledge from videos or digital images, which comes under the domain of computer vision. Computer vision deals with image data (such as digital images, a sequence of images, multi-view images, etc.) and information to form decisions. Detection of objects from different scenes is a prime requirement for various computer vision applications. Human visual system is one such example which can easily detect and recognize one class of object from another class of object. Object detection is widely used for automatic analysis of digital data, Human-Computer Interaction (HCI), automated processes, smart vehicles and wild animal detection. Among different researches, applications of object detection are in the domain of car detection, face detection, image retrieval and video surveillance. The main contribution of this paper is for analyzing the different object detection methods in various environments.

Proposed Solutions:

Fang, Y., discussed a technique to move animal detection by taking benefit of global patterns of pixel motion. In the dataset, where animals make obvious movement against the background, motion vectors of every pixel were estimated by applying optical flow techniques. A coarse segmentation then eliminates most parts of the background via applying a pixel velocity threshold. Using the segmented regions, another threshold was used to filter out negative candidates, which could belong to the background.

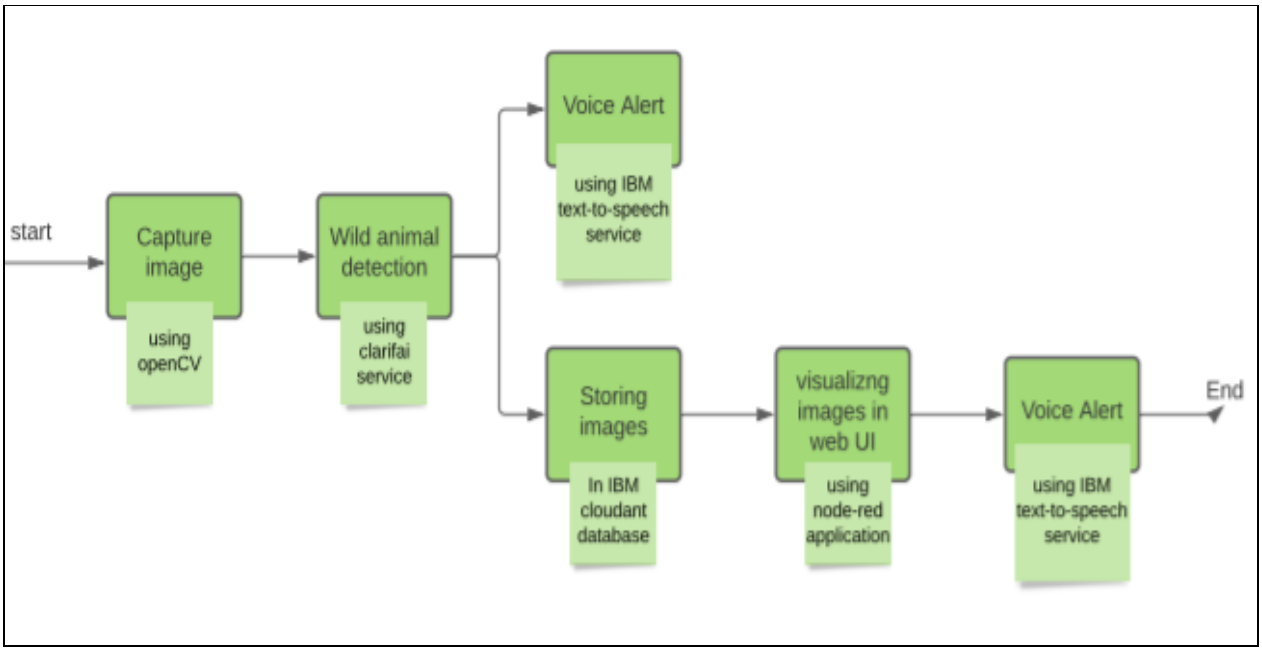
Jaskó, G., presented a system capable of detecting different huge sized wild animals from traffic scenes. Visual data was obtained from a camera with monocular color vision. The objective was to analyze the traffic scene image, to

locate the regions of interest and to correctly classify them for discovering the animals that were on the road and might cause an accident. A saliency map was generated from the traffic scene image using intensity, color and orientation features. The salient regions of this map were assumed to be regions of interest. A database was compiled from a large number of images containing various four-legged wild animals. Relevant features were extracted from these and were utilized for training Support Vector Machine (SVM) classifiers.

Gupta, P., & Verma, G. K. proposed a technique for detection of visual wild animals in images by dictionary learning. Discriminative Feature-oriented Dictionary Learning was utilized for learning discriminative features of positive images, that have animals present in positive class, in addition to of negative images that do not have animals present in that class. The system was created dictionaries that were class-specific and was capable of automatic feature extraction by example training image samples. The proposed approach was learned these dictionaries through positive (animal class and negative background class) sparse representation of image samples.

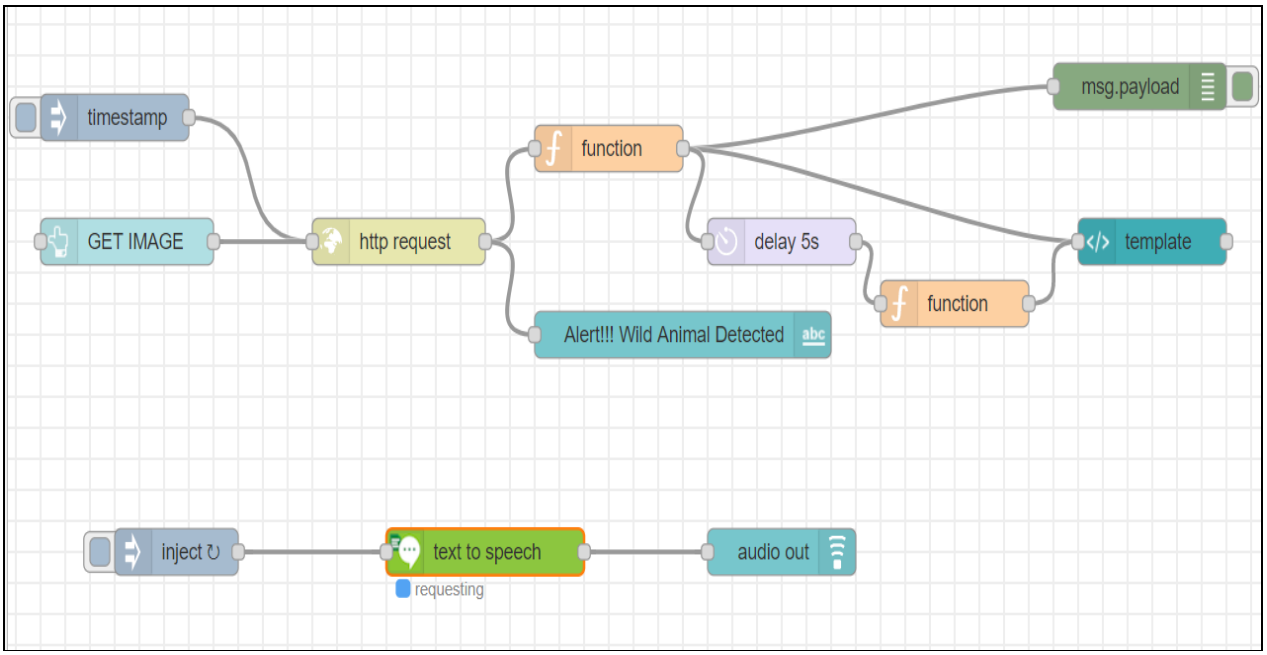
THEORITICAL ANALYSIS

3.1 Block Diagram



3.2 Software Designing

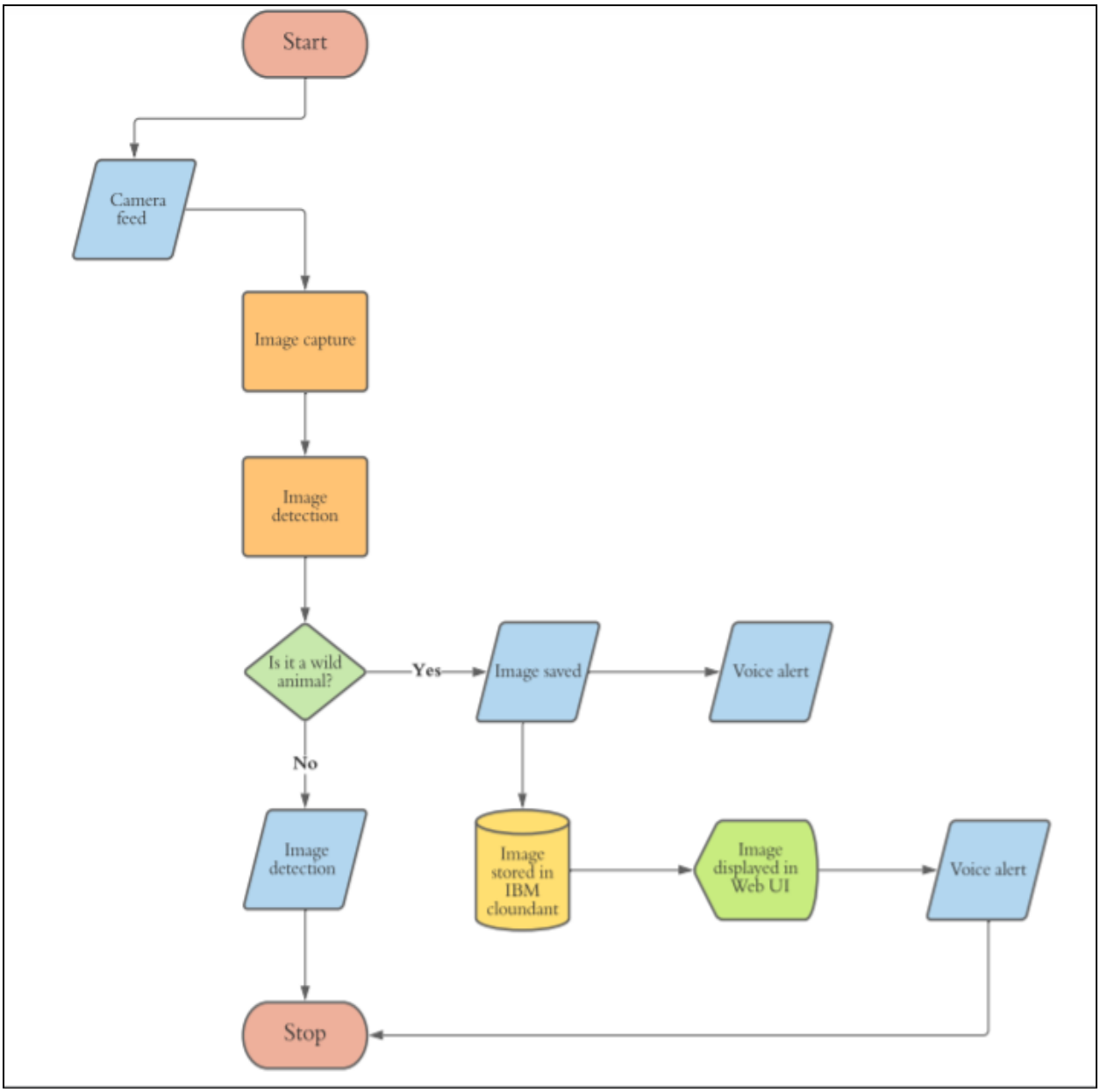
This snapshot is from the node-red application where we used these nodes to visualize the images and sending voice alerts to the Web UI.IBM cloud has many services which we used in our project in order to get the required result. We used Node-red to make web UI and text-to-speech service for voice alerts. Other than this we've used IBM cloudant database service to store images whenever a wild animal is detected.



EXPERIMENTAL INVESTIGATIONS

Our project used live video recognition so we used a video in order to test its working in which we experimented it with different animals. We showed some other animal pictures but it only detected the wild animals. The voice alert will only work after any wild animal is detected.

FLOW CHART


Start
Camera feed
Image capture
Image detection
Is it a wild animal?
Yes
Image saved
Voice alert
No
Image detection
Image stored in IBM cloundant
Image displayed in Web UI
Voice alert
Stop

RESULT

We have achieved the desired output. Our system detects the wild animals and notifies at the same time to the nearby people through voice alerts. We have displayed the images of the last animal detected through web UI.

ADVANTAGES

1. This system is very helpful to reduce the number of accidents that are happening due to vehicle-animal collisions.
2. It helps the farmers by alerting them at the moment so that they can take proper measures to save their crops.
3. It also alerts the forest authorities, if some wild animal is detected in the nearby cities or towns.

DISADVANTAGES

Sometimes the camera may detect inaccurately during night time and alert the owners, whereas the detected thing may not be a wild animal.

APPLICATIONS

1. It is used in highways to prevent animal- vehicle collision.
2. It is used in agriculture fields to prevent the wild animal from destroying the crops.
3. It is also used in the towns and villages that are nearby forests so that if any wild animal comes on the streets, the forest department gets alerted at the very moment.

CONCLUSION AND FUTURE SCOPE

The animals, many of which are already threatened or endangered, are often killed in retaliation or to prevent future conflicts. So, this zone is to be monitored continuously. With regard to this problem, we have made an effort to develop the system which will alert everyone with voice messages whenever a wild animal is detected and also it captures images so that they can be visualized in web UI so that suitable action can be taken.

This work can be further extended by sending an alert in the form of messages on their mobile phones when the animal is detected to the nearby people and forest office. The problem of crop vandalization by wild animals has become a major social problem in current time. In other words, while utilizing his/her crop production, every farmer should be aware and take into consideration the fact that animals are living beings and need to be protected from any potential suffering. It requires urgent attention and an effective solution.

Thus, this project carries a great social relevance as it will help farmers in protecting their fields and save them from significant financial losses and will save them from the unproductive efforts that they endure for the protection of their fields. Furthermore, it can be used to reduce human wildlife conflict and also animal accidents.

BIBLIOGRAPHY

1. https://www.researchgate.net/publication/281537804_Real_Time_Animal_Detection_System_using_HAAR_Like_Feature

2. http://www.jcreview.com/fulltext/jcr070185.pdf

3. https://smartbridge.teachable.com/purchase?product_id=3226410

APPENDIX

11.1 Source code

```
import cv2
import numpy as np
import datetime
from clarifai_grpc.channel.clarifai_channel import ClarifaiChannel
from clarifai_grpc.grpc.api import service_pb2_grpc
import datetime
import ibm_boto3
from ibm_botocore.client import Config, ClientError
from ibmcloudant.cloudant_v1 import CloudantV1
from ibmcloudant import CouchDbSessionAuthenticator
from ibm_cloud_sdk_core.authenticators import BasicAuthenticator
from ibm_watson import TextToSpeechV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
import playsound
import os

stub = service_pb2_grpc.V2Stub(ClarifaiChannel.get_grpc_channel())

from clarifai_grpc.grpc.api import service_pb2, resources_pb2
from clarifai_grpc.grpc.api.status import status_code_pb2

# This is how you authenticate.
metadata = (('authorization', 'Key 261f4f43ded94577b92c9749318d9461'),)

# Constants for IBM COS values
```

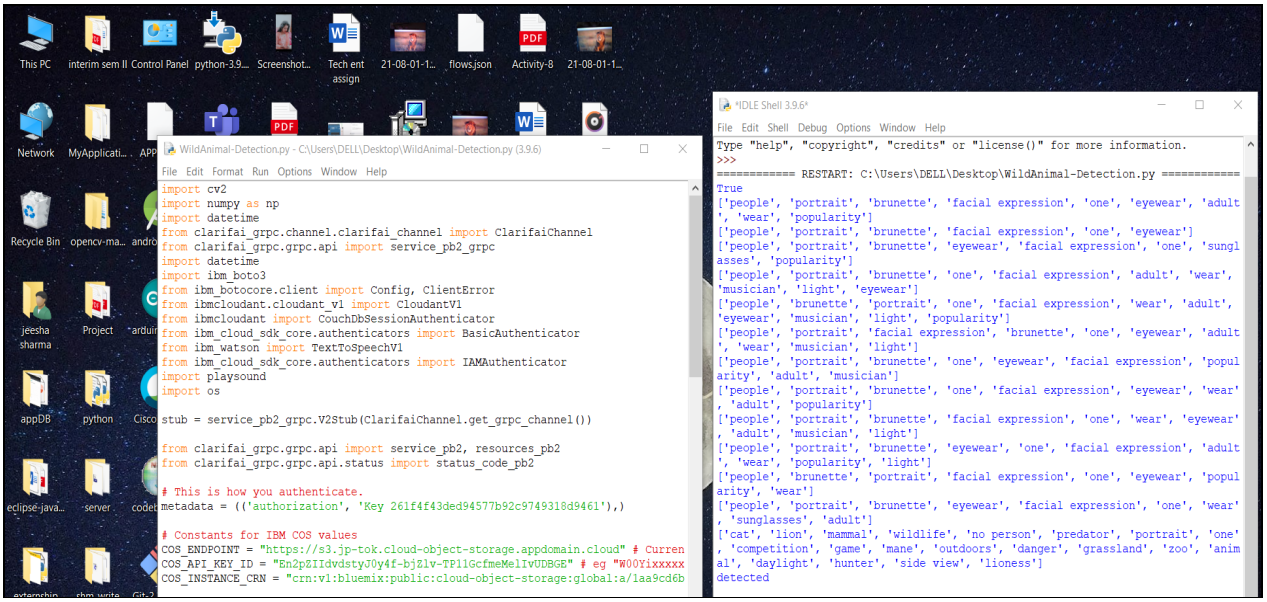
```
COS_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud" # Current list avaiable at https://control.cloud-object-storage.cloud.ibm.com/v2/endpoints
COS_API_KEY_ID = "En2pZIIdvdstyJ0y4f-bjZlv-TP11GcfmeMelIvUDBGE" # eg "W00YixxxxxxxxxxMB-odB-2ySfTrFBIQQWanc--P3byk"
COS_INSTANCE_CRN = "crn:v1:bluemix:public:cloud-object-storage:global:a/1aa9cd6bf5b348718754a22c6924419d:1b23606c-fbe7-4128-b678-f581dbfd38ac::" # eg "crn:v1:bluemix:public:cloud-object-storage:global:a/3bf0d9003xxxxxxxxxx1c3e97696b71c: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_INSTANCE_CRN,
    config=Config(signature_version="oauth"),
    endpoint_url=COS_ENDPOINT
)

authenticator = BasicAuthenticator('apikey-v2-1eqdq09ft2wf7gjqjscocw4lichwljwfj213q5cg809d', 'c4eb7c392f63fbe5d7050b86a867926e')
service = CloudantV1(authenticator=authenticator)
service.set_service_url('https://apikey-v2-1eqdq09ft2wf7gjqjscocw4lichwljwfj213q5cg809d:c4eb7c392f63fbe5d7050b86a867926e@0a7a725c-a76d-4389-bdbb-055690364fe8-bluemix.cloudantnosqldb.appdomain.cloud')

authenticator = IAMAuthenticator('QyNG-8trUbHS0NWYf7lUK5KSO5f4MFbQ6lLWhf8GqwWV')
text_to_speech = TextToSpeechV1(
    authenticator=authenticator
```

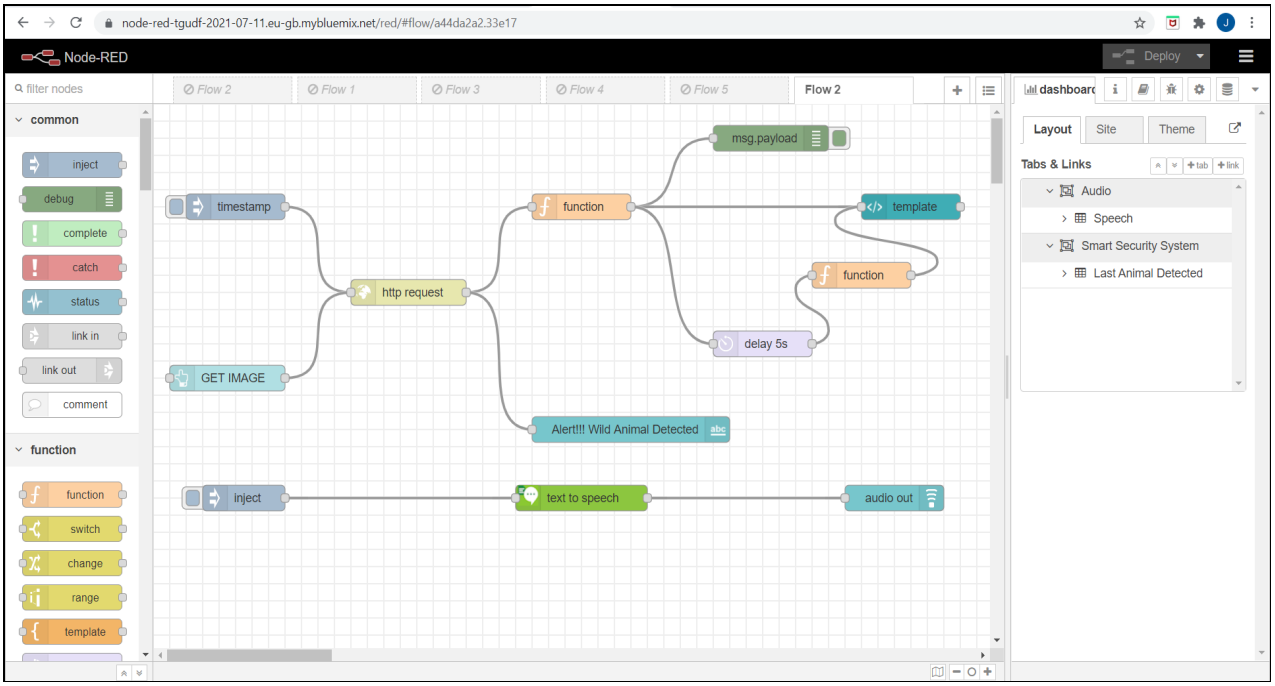
```
)

text_to_speech.set_service_url('https://api.eu-gb.text-to-speech.watson.cloud.ibm.com/instances/b186eb02-3386-482c-bb23-26b90758d879')


bucket = "animals128"
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))
```

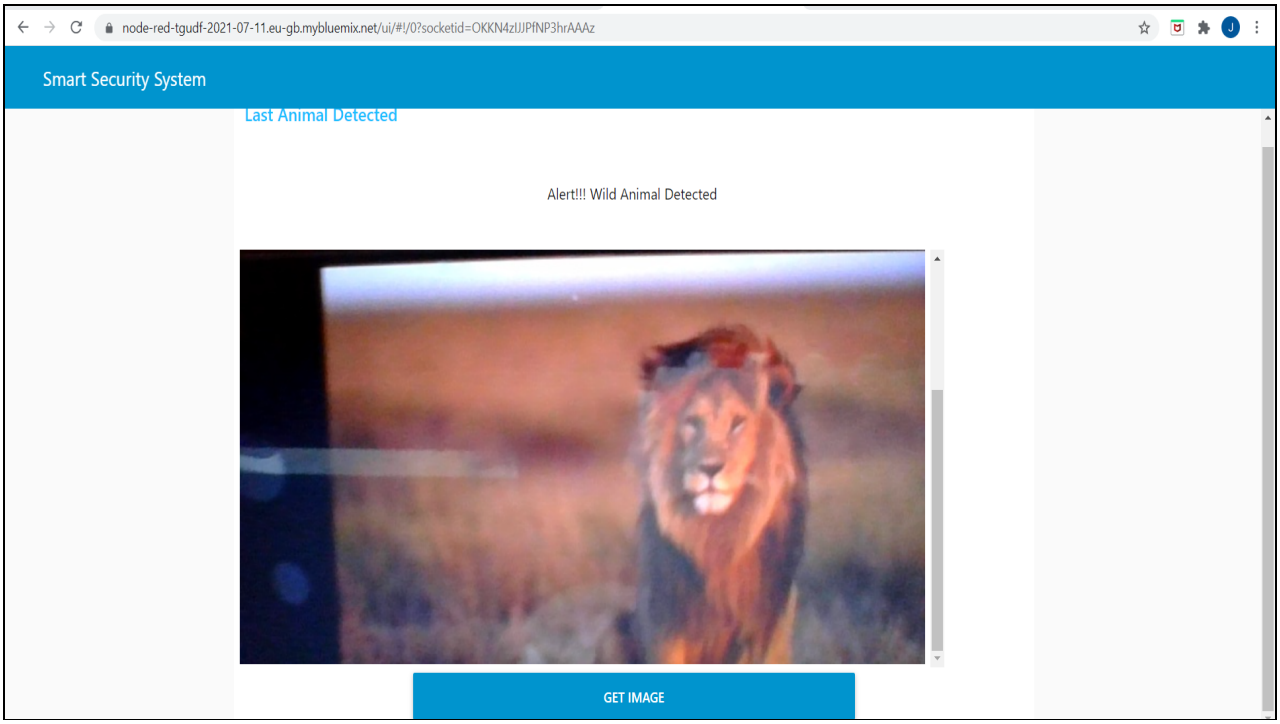
```
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to complete multi-part upload: {0}".format(e))

cap = cv2.VideoCapture(0)

print(cap.isOpened())

while cap.isOpened():
    ret, frame = cap.read()
    #print(ret, frame)
    cv2.imshow('Video', frame)
    cv2.imwrite('new.jpg', frame)
    with open('new.jpg' , "rb") as f:
            file_bytes = f.read()
    request = service_pb2.PostModelOutputsRequest(
            model_id='aaa03c23b3724a16a56b629203edc62c',

inputs=[resources_pb2.Input(data=resources_pb2.Data(image=resources_pb2.Imag
e(base64=file_bytes)))])
    response = stub.PostModelOutputs(request, metadata=metadata)

    if response.status.code != status_code_pb2.SUCCESS:
        raise Exception("Request failed, status code: " + str(response.status.code))



    a= []
    for concept in response.outputs[0].data.concepts:
        if(concept.value > 0.8):
            a.append(concept.name)
    print(a)
```

```
    for i in a:
        if(i=="animal" and "wild"):
            print("detected")
            with open('hello.mp3', 'wb') as audio_file:
                audio_file.write(text_to_speech.synthesize('Alert wild animal detected
beaware and
safe',voice='en-GB_KateV3Voice',accept='audio/mp3').get_result().content)

            playsound.playsound('hello.mp3')
            os.remove('hello.mp3')
            print("stopped")

            while(True):
                name = datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
                cv2.imwrite(name+".jpg", frame)
                multi_part_upload(bucket, name+'.jpg', name+'.jpg')
                json_document={"link":COS_ENDPOINT+'/'+bucket+'/'+name+'.jpg'}
                response = service.post_document(db='animal',
document=json_document).get_result()

            Key=cv2.waitKey(1)
            if Key==ord('q'):
                cap.release()
                cv2.destroyAllWindows()
                break
```

11.2 UI Output Screenshot






node-red-tgudf-2021-07-11.eu-gb.mybluemix.net/ui/#!/0?socketid=OKKN4zIJJPfNP3hrAAAz
Smart Security System
Last Animal Detected
Alert!!! Wild Animal Detected
GET IMAGE