IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE

A PROJECT REPORT

Submitted by

JANANI K - 711719106016

KALAIYARASI N - 711719106020

AKASH P - 711719106003

KARTHICK RAJA T - 711719106021

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

INSTITUTE OF TECHNOLOGY, SARAVANAMPATTI

ANNA UNIVERSITY: CHENNAI 600025

ABSTRACT

India is agriculture sector, on either side, is losing ground every day, affecting the ecosystem\'s output capacity. In order to restore vitality and put agriculture back on a path of higher growth, there is a growing need to resolve the issue. A large-scale agricultural system necessitates a great deal of upkeep, knowledge, and oversight. The IoT is a network of interconnected devices that can transmit and receive data over the internet and carry out tasks without human involvement. Agriculture provides a wealth of data analysis parameters, resulting in increased crop yields. The use of IoT devices in smart farming aids in the modernization of information and communication. For better crop growth moisture, mineral, light and other factors can be assumed. This research looks into a few of these characteristics for data analysis with the goal of assisting users in making better agricultural decisions using IoT. The technique is intended to help farmers increase their agricultural output.

TABLE OF CONTENTS

7.2 FEATURE 2

CHAPTER 1

INTODUCTION

Crops in farms are many times ravaged by local animals like buffaloes, cows, goats, birds etc. This leads to huge losses for the farmers. It is not possible for farmers to barricade entire fields or stay on field 24 hours and guard it. So here we propose automatic crop protection system from animals. This is a microcontroller based system using PIC family microcontroller. This system uses a motion sensor to detect wild animals approaching near the field. In such a case the sensor signals the microcontroller to take action. The microcontroller now sounds an alarm to woo the animals away from the field as well as sends SMS to the farmer so that he may know about the issue and come to the spot in case the animals don't turn away by the alarm. This ensures complete safety of crops from animals thus protecting the farmers loss.

1.1 PROJECT OVERVIEW

The production rate of crops is significantly declining due to natural disasters, animal interventions and plant diseases. Internet of things (IoT) and wireless sensor networks are widely applied in crop field monitoring systems to observe the quality of each plant and the field. This work proposes IoT based crop field protection system (ICFPS) that monitors and protects the crop fields from animal intrusions. This proposed system uses ultrasonic sensors, hyperspectral cameras, voice recorded buzzers and other agriculture sensors to protect the entire crop field. This system uses numerous sensor nodes and cameras for gathering field objects (images and environmental objects). The proposed ICFPS creates deep learning techniques such as recurrent convolutional neural networks (RCNN) and recurrent generative adversarial neural networks (RGAN) for feature extraction, disease detection and field data monitoring practices. This proposed work develops a smart city-based agriculture system using cognitive learning approaches. This proposed system

analyses crop field data and provide automatic alerts regarding animal interferences and crop diseases.

1.2 PURPOSE

This system used in smart crop protection system. The purpose of SCPS is to secure or protect the farm from the theft in the farm or main purpose of this project is to alert the farmer as well as fear the animals with getting harm to animals. The Smart protection system define to help the farmer for the protection of a farm. We have designed this project for the only secure from animals but we this project have the provision to secure from the human begins also. This can achieve by the help of IOT device that we are discuss in this paper. The SCPS work on the battery so that this project can be easily portable and also we are add solar panels and converter modules this can help the battery to charge from solar energy. The IOT device is used to indicate the farmer by a message while someone enter into the farm and we are used SD card module that helps to store a specified sound to fear the animals.

CHAPTER 2

LITERATURE REVIEW

2.1 EXISTING PROBLEM

In existing system, the smart production system defines that this project help to form for the protection of farm. We have designed this project for the only secure from animals but this project has the provisions of to secure from human beings also. This can be achieved by the help of IOT device. The SCPS works on the battery so that this project can be easily portable and also we are added the solar pannels and converter modules. This can help the battery to charge from solar energy. The IOT device is used to indicate the farmer by the message while someone enter into farm and we are used SD card module that helps to store a specified sound to fear the animals.

2.2 REFERENCE

[1] Ipseeta Nanda, Medepalli Swathi, Lizina Khatua, vol.09,DIO:10.1088 issue no:06, published:2021

[2] Shishir Bagal1 , Krunal Mahajan2 , Riya Parate3 , Ekta Zade,vol.07,

Published 2021

[3] Abhilash Lad, Sumitra Nandre, Krishna Raichurkar, Sumit Zarkhande, Published: 2022-0225, vol:06, issue.no:22, ISSN:2321-9653

[4] Tanmay Baranwai, Nitika, Pushpendra Kumar Pateriy, Published:2019 IEEE DOI:10.1109

[5] Dr.N.Suma, Sandra Rhea Samson, Saranya, vol no:5, issue:2ISSN:2321-8169 Published:2020

[6] Moummadi K, Abidar R, Medromi. H vol no:4, Issue no:1, Published:2016

[7] Abdulai. A, W.E. Huffman, 2009 IEEE, DOI:10.1109/MLSP.2015.7324362, vol:07

[8] D. Bettena Sheryl Fernando, R.Ramya Priya , S. Santhoshini

Vol no:06, Issue:04, ISSN: 2395-566x, Published:2020

[9] P.Venkateswara Rao,Ch Siva Rama Krishna, M. Samba Siva Reddy, ISSN:2279-543x, vol no:06, Issue no:06,published:2019

[10] Navyashree.H, Balakrishna K, Abilash M IETE , ISSN:2278-0181, vol no:07, Issue no:02, published:2018

2.3 PROBLEM STATEMENT DEFINITION

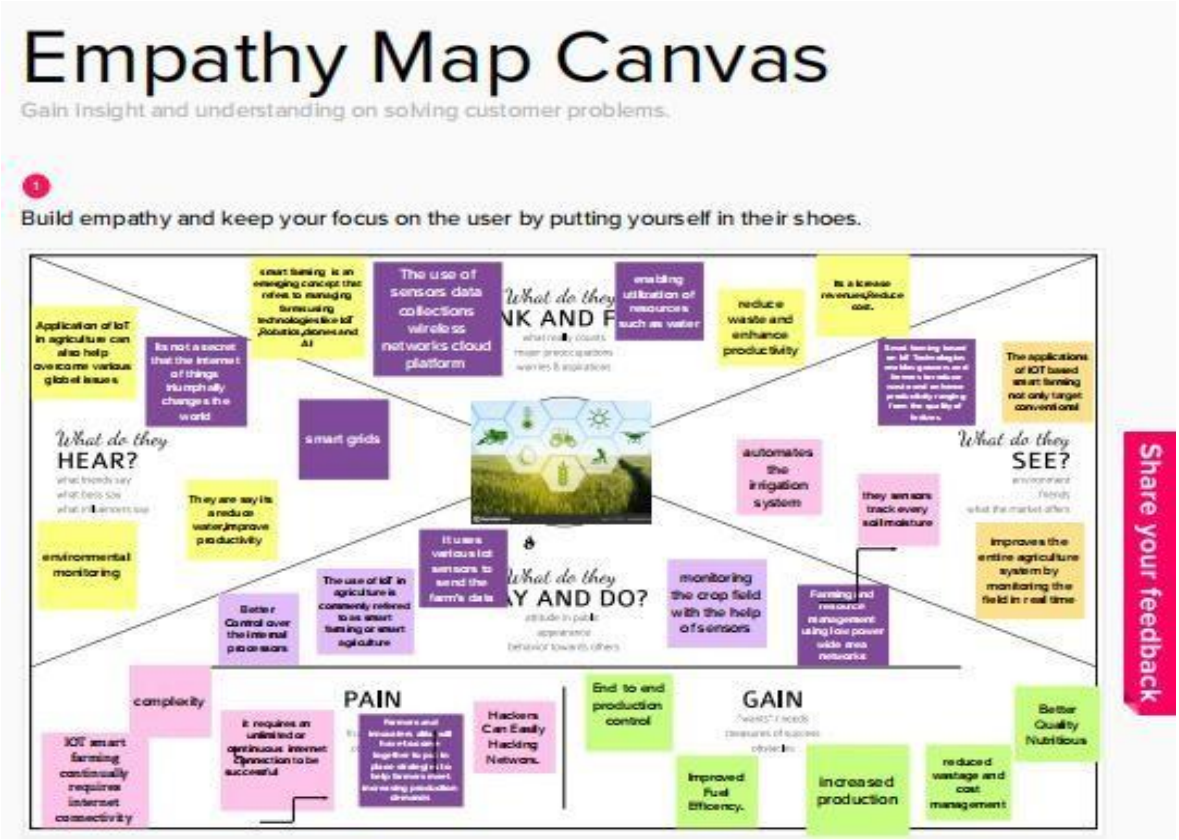
One of the biggest problems farmers face the attack on crops by wild animals in their fields. A damage from the attack significantly and adversely affected the crop yield. Farmers in the village use electrical fencing around the filed to keep away wild animals. But due to many accidents, which have caused the death of farmers as well as animals, this approach is not so appreciated by farmers. As an alternative electrical fencing the farmers keep vigil at night to keep the wild animals away. This is very strenuous task and lack of sleep adversely affect the farmers work during the day time. The damaged caused by the animals to the crops affect total yield of the harvest immensely and the farmers have to suffer the loss in their income because of this. So they can used the IOT based intelligent system that can used to prevent the crop damaged due to wild animals.

CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

An empathy map is a simple, easy-to-digest visual that captures knowledge about auser's behaviors and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



3.2 IDEATION AND BRAINSTROMING

3.3 PROPOSED SOLUTION

S.No	Parameter	Description
1.	• Problem Statement (Problem to be solved)	• Crops are not irrigated properly due to insufficient labour forces. • Improper maintenance of crops against various environmental factors such as temperature climate, topography and soil quality which results in crop destruction. • Requires protecting crops from Wild animals attacks, birds and pests.
2.	• Idea / Solution description	• Moisture sensor is interfaced with Arduino Microcontroller to measure the moisture level in soil and relay is used to turn ON and OFF the motor pump for managing the excess water level. It will be updated to authorities through IOT. • Temperature sensor connected to microcontroller is used to monitor the temperature in the field. • Image processing techniques with IOT is followed for crop protection against animal attacks.
3.	Novelty / Uniqueness	• Automatic crop maintenance and protection using embedded and IOT technology.
4.	Social Impact / Customer Satisfaction	• This proposed system provides many facilities which helps the farmers to maintain the crop field without much loss.
5.	Business Model (Revenue Model)	• This prototype can be developed as product with minimum cost with high performance.
6.	Scalability of the Solution	• This can be developed to a scalable product by using sensors and transmitting the data through Wireless Sensor Network and analyzing the data in cloud and operations is performed using robots.

3.4 PROBLEM SOLUTION FIT

CHAPTER 4

REQUIREMENT ANALTSIS

4.1 FUNCTIONAL REQUIREMENTS

- FR-1 User Registration, Registration through Form Registration through Gmail Registration through Linked IN

- FR-2 User Confirmation, Confirmation via Email Confirmation via OTP

- FR-3 Tracking Expense Helpful insights about money management

- FR-4 Alert Message Give alert mail if the amount exceeds the budget limit

- FR-5 Category This application shall allow users to add categories of their expenses

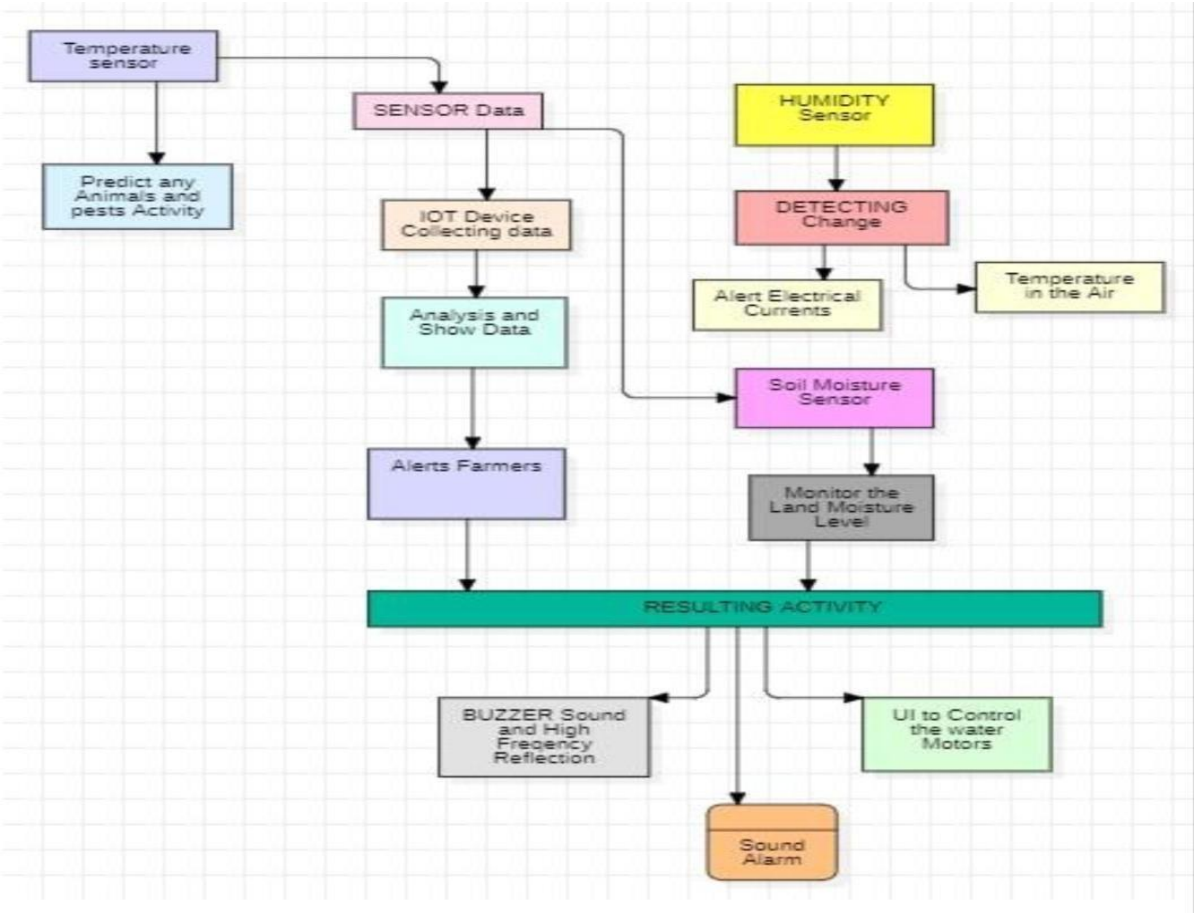
4.2 NON- FUNCTIONAL REQUIREMENT

- NFR-1 Usability You will able to allocate money to different priorities and also help you to cut down on unnecessary spending
- NFR-2 Security More security of the customer data and bank account details.
- NFR-3 Reliability Used to manage his/her expense so that the user is the path of financial stability. It is categorized by week, month, and year and also helps to see more expenses made. Helps to define their own categories.
- NFR-4 Performance The types of expense are categories along with an option. Throughput of the system is increased due to light weight database support.
- NFR-5 Availability Able to track business expense and monitor important for maintaining healthy cash flow. NFR-6 Scalability The ability to appropriately handle increasing demands

CHAPTER 5

5.1 DATA FLOW DIAGRAMS

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is store.

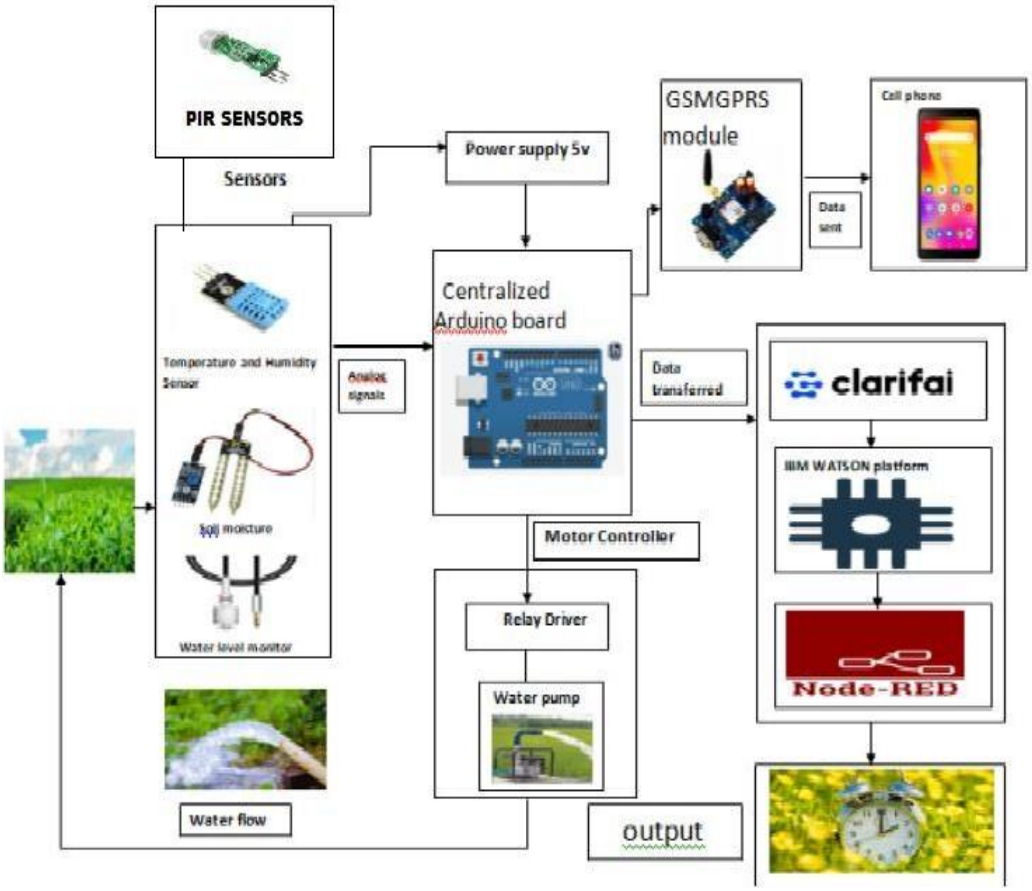


5.2 SOLUTION ARCHITECTURE

Solution architecture is a complex process – with many sub-processes – thatbridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behaviour, and other aspects of the softwareto project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, anddelivered.

5.3 SOLUTION ARCHITECTUTRE DIAGRAM



CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint1	Sensor Data(python script)	USN-1	The Data of sensor which are feed to the Raspberry pi, Here we are using python script to generate a random sensor data.	3	High	JANANI K (Team leader)
Sprint1	Automation (python script)	USN-2	Some activities are made to automation to overcome insufficient of labourforce in the field.Hence that also included in python script to implement automation	5	High	JANANI K (Team leader) KALAIYARASI N (Team Member)
Sprint2	IBM IOT platform	USN-3	To Send the raspberry pi data to IOT platform, we create an IBM IOTplatform and connectthe raspberry pi to thedevice created in IBMIOT.	5	High	JANANI K (Team Leader) AKASH P (Team Member) KARTHICK RAJA T (Team Member)
Sprint3	Node RED service	USN-4	To access the IBM IOT platform from external application or from external UI Node red service is established.	5	High	JANANI K (Team Leader)
Sprint3	API Key	USN-5	To protect the IBM IOTplatform creating an API Key .		High	AKASH P (Team Member)

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	8	6 Days	24 Oct 2022	29 Oct 2022	8	29 Oct 2022
Sprint-2	5	6 Days	31 Oct 2022	05 Nov 2022	5	05 Nov 2022
Sprint-3	8	6 Days	07 Nov 2022	12 Nov 2022	8	12 Nov 2022
Sprint-4	8	6 Days	14 Nov 2022	19 Nov 2022	8	19 Nov 2022

7. CODING AND SOLUTIONING

7.1 FEATURES

Feature 1: Detect the Temperature

Feature 2: Detect the Humidity

Feature 3: Detect the Moisture

Feature 4: Detect the Animals

Codes:

PYTHON CODE TO IBM CODE

```
import cv2
import numpy as np
import wiotp.sdk.device
import playsound
import random
import time
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
from clarifai_grpc.channel.clarifai_channel import ClarifaiChannel
from clarifai_grpc.grpc.api import service_pb2_grpc
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 bc885e5165d74ef48f42f6f6a2c9eb87'),)
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 = " f6Ap-ct18m07S9UZL7XPbAF7170ome PLLUQOzqmnAzb5"
# eg "W00YiRnLW4a3fTj MB-odB-2ySfTrFBIQQ'Wanc -- P3byk"
COS_AUTH_ENDPOINT = "https://iam.cloud.ibm.com/identity/token"
COS_RESOURCE_CRN = "crn:v1:bluemix:public:cloudantnosqldb:eu-
gb:a/d43aa7d0631b400e9283084df08f9f60:502851d6-a240-4b22-8d4b-
3642ed2bc3a8::" # eg "crn:vl:bluemix:public:cloud-object-
storage:global:a/6b644a3fda97448b888c23eeef263ed6:199ab1e5-0d9d-420f-8e4a-
98d868c04368 ::"
clientdb = Cloudant("apikey-v2-1wveoo6739lo7qj5cy7kqtpfsku8dumxlvp6dy62rwu2",
"64455b04f35e5d5f9b4fc25bb38904af", url = "https://apikey-v2-
1wveoo6739lo7qj5cy7kqtpfsku8dumxlvp6dy62rwu2:64455b04f35e5d5f9b4fc25bb38
904af@de3c99da-899c-43cb-9aa5-b6b3fdc4cc16-
bluemix.cloudantnosqldb.appdomain.cloud",
  username = "apikey-v2-1wveoo6739lo7qj5cy7kqtpfsku8dumxlvp6dy62rwu2")
clientdb.connect()
#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))
def myCommandCallback(cmd) :
        print("Command received: %s" % cmd.data)
        command=cmd.data[ ' command']
        print(command)
        if(command =='lighton'):
            print('lighton')
        elif(command =='lightoff'):
            print('lightoff')
        elif(command =='motoron'):
            print('motoron')
        elif(command =='motoroff') :
            print('motoroff')
myConfig = {
"identity": {
"orgId": "hj5fmy",
"typeId": "NodeMCU",
"deviceId": "12345"
},
"auth": {
"token": "12345678"
}
}
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()
database_name = "sample"
my_database = clientdb.create_database(database_name)
if my_database.exists():
  print(f"¹ {database_name} ' successfully created.")
cap=cv2.VideoCapture('garden.mp4')
if(cap.isOpened()== True) :
  print('File opened')
else:
  print('File not found')
while(cap.isOpened()) :
   ret, frame=cap.read()
   gray = cv2.cvt5Color(frame, cv2.COLOR_BGR2GRAY)
   ims = cv2.resize(frame, (960, 540))
   cv2.imwrite(' ex.jpg' ,ims)
   with open("ex.jpg", "rb") as f:
```

```
      file_bytes = f.read()
   # This is the model ID of a publicly available General model. You may use any
other public or custom model ID.
   request = service_pb2.PostModelOutputsRequest(
      model_id='aaa03c23b3724a16a56b629203edc62c',
      inputs=[resources_pb2.Input(data=resources_pb2.Data(image=resources_pb2.I
mage(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))
   detect=Falss
   for concept in response.outputs[0] .data.concepts:
     #print('%12s: %.2f¹ % (concept.name, concept.value))
     if(concept.value>0.98):
        #print(concept.name)
        if(concept.name =="animal") :
            print("Alert! Alert! animal detected")
            playsound.playsound('alert.mp3')
            playsound.playsound('alert.mp3')
            picname=datetime.datetime.now() . strftime("%Y-%m-%d-%H-%M")
            cv2.imwrite(picname+ ' . jpg',frame)
            multi_part_upload('gnaneshwar' , picname+ ' .jpg' , picname+ ' . jpg')
            json_document={"link":COS_ENDPOINT+'/'+'gnaneshwar'+'/'+picname+'
.jpg'}
            new_document = my_database.create_document(json_document)
            if new_document.exists():
                print(f"Document successfully created.")
            time.sleep(5)
            detect=True
   moist=random.randint(0,100)
   humidity=random.randint(0,100)
   myData={ 'Animal' : detect, 'moisture' :moist, 'humidity':humidity}
   print(myData)
   if(humidity!=None):
       client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
       print("Publish Ok ..")
   client.commandCallback = myCommandCallback
   cv2.imshow('frame ' , ims)
   if cv2.waitKey(1) & 0xFF == ord('q'):
       break
client.disconnect()
cap.release()
cv2.destroyAllWindows()
```

8.TESTING:

8.1 TESTING

- PYTHON CODE TO IBM
- IoT SENSOR OUTPUT
- IBM CLOUD TO NODE RED OUTPUT

8.2 USER ACCEPTANCE TESTING

8.1 PURPOSE OF DOCUMENT

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

8.2 DEFECT ANALYSIS

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6

Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

8.3 TEST CASE ANALYSIS

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3

Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9.RESULT

ADVANTAGES

- Farmers can monitor the health of farm animals closely, even if they are physicallydistant.
- Smart farming systems reduce waste, improve productivity and enable management ofa greater number of resources through remote sensing.
- High reliance.
- Enhanced Security.

DISADVANTAGES

- Farms are located in remote areas and are far from access to the internet.
- A farmer needs to have access to crop data reliably at any time from any location, soconnection issues would cause an advanced monitoring system to be useless.
- High Cost
- Equipment needed to implement IoT in agriculture is expensive.

CONCLUSION

The problem of crop vandalization by wild animals and fire has become a major socialproblem in current time. It requires urgent attention as no effective solution exists till date for this problem. Thus, this project carries a great social relevance as it aims to address this problem. This project will help farmers in protecting their orchards and fields and save them from significant financial losses and will save them from the unproductive efforts that they endure for the protection their fields. This will also help them in achieving better crop yields thus leading to their economic well being.

FUTURE SCOPE

Study and analysis of the developed Crop protection systems for its cost effectiveness with the development of Arduino based variable frequency Ultrasonic bird deterrent circuit. outline of the crop damage caused by a particular Wild animal if the behavioral features of the with the reduced cost in the smart phones.

APPENDIX

SOURCE CODE

The source code has been uploaded in git hub.

GITHUB LINK- https://github.com/IBM-EPBL/IBM-Project-11713-1659340068