ANIMAL DETECTION IN FARMS USING OPENCV

A Project Report for

CSE 6363 MACHINE LEARNING

SPRING 2024

By

GROUP – 6

MUNI SAI KALYAN TEJA DUDI - 1002104402

RICHITHAREDDY GORLAGUNTA - 1002126204

SPANDANA KOLLIPARA - 1002126792

NANDINI NIDUMOLU - 1002081505

Under the Esteemed Guidance of

Prof. YINGYING ZHU

Professor, Department of Computer Science

UNIVERSITY OF
TEXAS
ARLINGTON

DEPARTMENT OF COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON

APRIL 2024

TABLE OF CONTENTS

ABSTRACT

Agriculture plays a huge role, Animal intrusion in farms causes huge losses in agricultural revenue which a farmer cannot bear, especially if they have small farming areas as the majority of the farmers. Agriculture is the most important sector of Economy but the issue of damage to crops by wild animals has turned into an important social issue in current occasions. So far, many of the farmers rely on guards to guard their crops which increases the overhead costs. But, due to current climate conditions, crop failure rate has increased dramatically. Debt in the agricultural sector has increased tremendously. In these situations, a farmer cannot expect further destruction of crops, and neither can afford to increase costs in farming. Computer Vision is being increasingly applied in the agricultural field for higher productivity by automating tasks. We propose an AI based system which monitors the field using cameras for any intrusion by the animals and alerts the farmer or can even take certain actions on its own.

CHAPTER 1 INTRODUCTION

1.1 INTRODUCTION

Agriculture meets the food demands of the population and provides various raw materials for different industries. Interference of animals in agricultural lands causes a huge loss of crops. Crop damage due to raiding wild animals has become a major issue of concern these days. Animals like wild boars, macaques, porcupines, deer, monkeys, and bears are extremely destructive and have also caused human casualties on certain occasions. Small farmers can even lose up to half of their yield to animals and they cannot take any harsh measures due to the strict wildlife laws. Human-elephant conflict is rising intensely as elephants are a highly conflict prone wildlife species, especially in India. Thus, there is need for a system to detect any intrusion which can help the farmers to drive away these animals as soon as they learn about their intrusion.

Computer vision is applicable to many fields like medical fields, robotics, remote sensing, machine vision, content-based image retrieval. Computer vision solves many problems in different disciplines. Computer vision also applied in the security field to perform automatic surveillance and access control and attendance management. The computer vision can be applied in the agriculture field in many ways like disease detection of a tree by examining leaves or flowers or fruits and quality control of agricultural products.

Computer vision techniques can be applied in order to provide security from wild animals in agriculture. In agriculture fields near to forest areas have a severe threat from wild animals, which attacks regularly on farms. These attacks cause huge damage to agricultural crops and subsequently cause significant financial losses to farmers.

Some measures are taken by the farmers by installing electrical fences on the farms, big flood lights in the farm. Some even resort to hiring guards. Installing an electrical fence is much costlier than equipping huge farms and kills so many animals, which is even illegal in certain places and affects biodiversity. Other existing techniques also are not effective due to several reasons, cost being one of them.

We proposed a new and cost-effective solution for agriculture security from animals. It is a proactive solution which gives alerts to the farmers when animals come

near to the farms. It also causes certain sirens to be played whenever any animals are detected and is directed towards the animal to scare them away. Here, we are implementing a solution that recognizes animals when they are captured on camera.

1.2 PROBLEM STATEMENT / MOTIVATION

Agriculture plays a huge part, beast intrusion in granges causes huge losses in agrarian profit which a planter cannot bear, especially if they've small husbandry areas as maturity of the growers in India. Computer Vision are being decreasingly applied in agrarian field for advanced productivity by automating tasks. We propose an AI grounded system which monitors the field using cameras for any intrusion by the creatures and cautions the planter or can indeed take certain conduct on its own.

1.3 AIM & OBJECTIVE

- The main aim of the project is to detect animals in the field.
- We propose an AI based system which monitors the field using cameras for any intrusion by the animals and alerts the farmer or can even take certain actions on its own.
- To provide improved prediction estimate and range.
- To enhance the prediction power and speed of accuracy & provides immediate location tracking of where animal occurs.

1.4 SCOPE

- Automatic animal intrusion allows farmers to increase yield and revenue.
- Industry relying on agriculture in their business can have better control on the supply.
- AgriTech industries can depend more of AI so that they can handle the labour issue during seasonality which is common due to the default nature of agriculture.
- Government agencies and Policy makers can also utilize the system to better ensure the crop supply and prevent high inflation of crop prices.

CHAPTER 2 SURVEY OF THE LITERATURE

LITERATURE REVIEW

[1] Parikh, M. et al. "Wild-Animal Recognition in Agriculture Farms Using WCOHOG for Agro-Security." (2017).

Computer Vision is applied in the agriculture field for food grading, disease identification of plants and agro-farms security. Huge crop damage is caused by the wild animal attacks on the agriculture farms. Here are some traditional techniques followed by the local farmers, but which are not effective. This problem can be solved using computer vision techniques. In this paper, we proposed an algorithm to detect animals in each image. WCoHOG is a Histogram oriented gradients-based feature vector with better accuracy. It is an extension of Co-occurrence Histograms of Oriented Gradients (CoHOG). In this paper LIBLINEAR classifier is used to get better accuracy for high dimensional data. The experiments were conducted on two benchmark datasets called Wild-Anim and CamaraTrap dataset. Experimental results prove that W-CoHOG performs better than existing state of the art methods.

Summary: This journal discusses Computer Vision and its application in the field of agricultural safety.

[2] S. Yadahalli, A. Parmar and A. Deshpande, "Smart Intrusion Detection System for Crop Protection by using Arduino," *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, 2020, pp. 405-408, doi: 10.1109/ICIRCA48905.2020.9182868.

Agriculture is still one of the most crucial sectors of the Indian economy. It is important for human survival as well as economic growth. Traditional systems like humanoid scarecrows are used even today in an agricultural field to stop birds and animals from disturbing and feeding on growing crops. Thus, this paper focuses on proposing a system which detects the intruders, monitors any malicious activity and then reports it to the owner of the system. It acts as an adaptable system which provides a practicable system to the farmers for ensuring complete safety of their farmlands from any attacks or trespassing activities.

Summary: This journal helps us in understanding the use of AI and its integration with motion detector and IR sensor to the agricultural field.

[3] Howard, Andrew & Zhu, Menglong & Chen, Bo & Kalenichenko, Dmitry & Wang, Weijun & Weyand, Tobias & Andreetto, Marco & Adam, Hartwig. (2017). "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications."

We present a class of efficient models called MobileNets for mobile and embedded vision applications. We introduce two simples global hyperparameters that efficiently tradeoff between latency and accuracy. These hyper-parameters allow the model builder to choose the right sized model for their application based on the constraints of the problem. We present extensive experiments on resource and accuracy tradeoffs and show strong performance compared to other popular models on ImageNet classification. We then demonstrate the effectiveness of MobileNets across a wide range of applications and use cases including object detection, finegrain classification, face attributes and large-scale geo-localization.

Summary: In this paper, we learn about MobileNet SSD (Single Shot Detection) which we have extensively used in our project.

[4] Deshpande, Abhinav. (2016). Design and Implementation of an Intelligent Security System for Farm Protection from Wild Animals. 5. pp.2319-7064.

Crops are vulnerable to wild animals. Therefore, it is very important to monitor the nearby presence of animals. In this paper, we propose a method to protect farms from wild animals via ubiquitous wired network devices, which is applied to farm along with traditional methods to improve the protection performance. Operational amplifier circuits are utilized mainly for the detection of animal intrusion from the outside of farms. The proposed monitoring scheme is to provide early warning about possible intrusion and damage by wild animals.

Summary: In this paper, we learn about animal detection and protection of crops using embedded systems.

CHAPTER 3 SYSTEM ANALYSIS

3.1 EXISTING METHOD

Since most of the farms in India are small, most farmers rely on medieval techniques like using a scare crow or relying on guards to monitor crops. More recently, crops are also being protected using electric fencing, but it can be highly cost inefficient which a small farmer cannot afford. Even if they can afford it, in most cases it is illegal to use such fences which governments use as a measure to conserve the wildlife populations. Also, in busy seasons like the harvesting time, it can get difficult to have a guard, guarding and monitoring the crops from animals.

3.1.1 DISADVANTAGES

- High cost.
- Prone to seasonality.
- Requires costly equipments and infrastructure.
- Are highly inefficient.

3.2 PROPOSED METHOD

We propose an AI based surveillance system to detect and monitor the presence of any animal. A camera can be placed conveniently at location(s) where any possible animal might enter from. The system uses computer vision using OpenCV to process the feed from the camera. Pre-trained model MobileNet SSD (Single Shot Detector) is used to detect the animals in the farms. The model is trained on MS COCO image dataset. A siren is fired on detecting an animal which can act as a deterrent to the animal. It can also notify the farmer so that he/she can take the concerned action as required in time.

3.2.1 ADVANTAGES

- Lower cost to operate and they are very efficient & they are efficient.
- Seasonality does not affect at all.
- Does not require very high performance or costly hardware(s).

3.2.2 APPLICATIONS

- Helps increasing the yield in farming by preventing animals from entering.
- Higher revenue can be generated due to an increase in yield.
- Farmers don't need to hire any guards anymore, which saves money.
- AgriTech industries can reduce the effects of seasonality of agriculture in their business by relying on AI.

3.3 MODULES

- Upload(live)
- View
- Preprocessing
- The Model
- Prediction
- User Interface

3.3.1 Upload (Live)

Upload a video as a live feed using a webcam (or any camera attached in a farm).

```
print("Starting camera feed...")
vs=cv2.VideoCapture(0, cv2.CAP_DSHOW)
#vs=VideoStream(src=0).start()
```

Output

fig 3.3.1 Window of starting camera feed (live)

3.3.2 View

Video can be viewed live in a dialog box.

```
#Show the frames
cv2.imshow("Frame", frame)
count.append(det)
```

Output

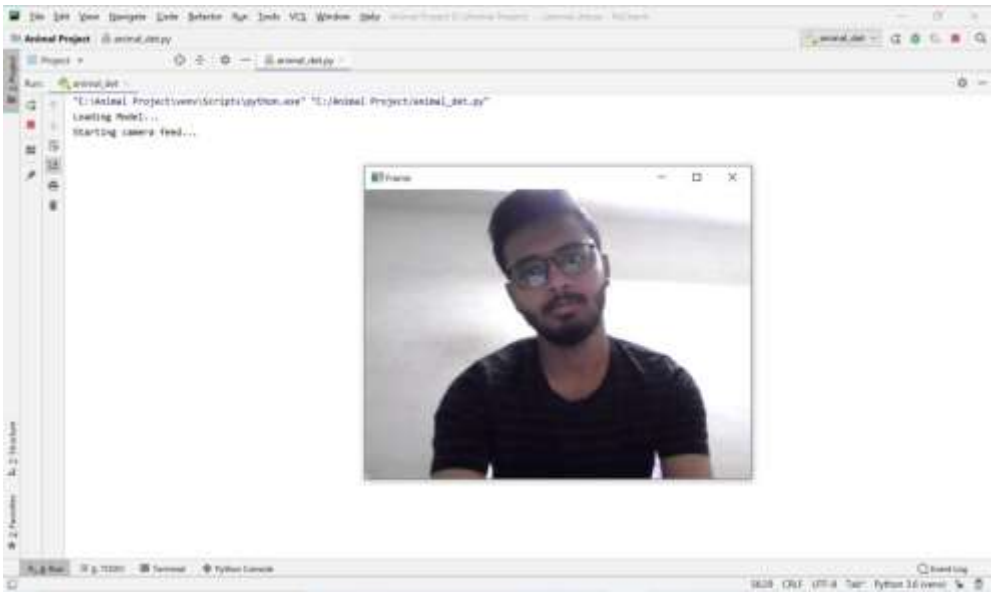


fig 3.3.2 Viewing a man in input live video streaming

3.3.3 Preprocessing

Data Preprocessing is a technique that is used to convert raw data into a clean data set. Cleaning the data refers to removing the null values, filling the null values with meaningful value, removing duplicate values, removing outliers, removing unwanted attributes. If a dataset contains any categorical records means convert those categorical variables to numerical values.

In this case, we are taking a live video feed in the form of images and resizing them to a standard size.We use MobileNet SSD pretrained model which identifies features in any image using a Convolution Neural Network (CNN) model.

```
#Read frame by frame
```

```
frame = imutils.resize(frame, width=500)
#Take the frame dimentions and convert it to a blob
(h, w) = frame.shape[:2]
blob = cv2.dnn.blobFromImage(cv2.resize(frame, (300,300)), 0.007843,
(300,300), 127.5)
```

Output

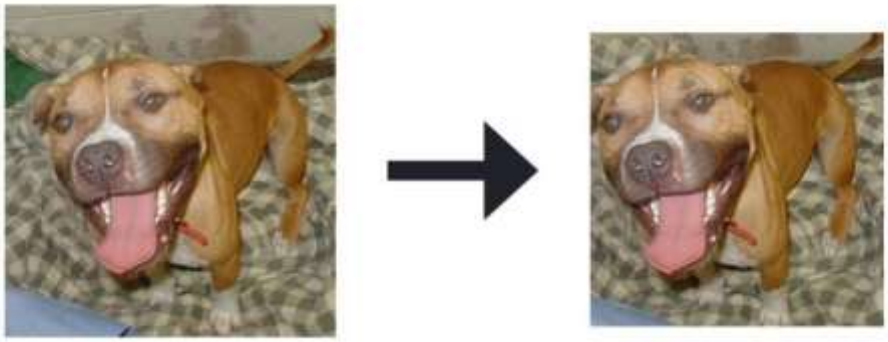


fig 3.3.3 Input image and Preprocessing image

3.3.4 The Model Training

- SSD (Single Shot Detector) is a popular algorithm in object detection.
- It's generally faster than RCNN.
- SSD has two components: a backbone model and SSD head. *Backbone* model usually is a pre-trained image classification network as a feature extractor.
- Here, we will use MobileNet SSD model to detect the objects.
- Here, VGG Net is used as a backbone model to extract the features from the images.
- Convolution layers (CNN) are then used for object detection in the images using the feature map generated by VGG net layer.
- The model is able to detect multiple objects in any given image.
- For the purpose of classification, the model uses softmax in the last layer.
- Softmax takes in a vector of numbers and converts them to probabilities which are then used for image generating results.

- Softmax converts logits into probabilities by taking the exponents from every output and then normalize each of these numbers by the sum of such exponents, such that the entire output vector adds up to

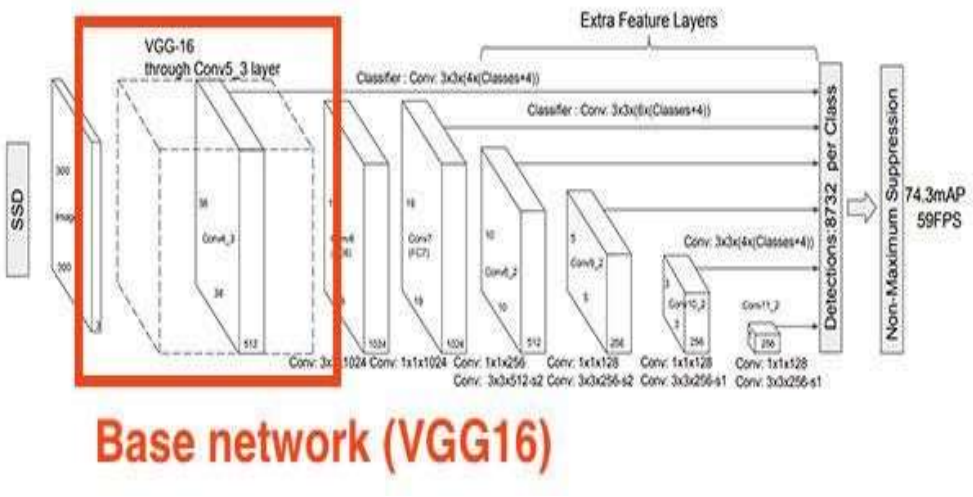


Fig (3.3.4.1) The Model training (base network VGG16)

```
protext=r"DNN_Object_Detection-master/DNN_Object_Detection-
master/MobileNetSSD_deploy.prototxt.txt"
model=r"DNN_Object_Detection-master/DNN_Object_Detection-
master/MobileNetSSD_deploy.caffemodel"

# initialize the list of class labels MobileNet SSD was trained to
# detect, then generate a set of bounding box colors for each class
CLASSES = ["background", "aeroplane", "bicycle", "bird", "boat",
"bottle", "bus", "car", "cat", "chair", "cow", "diningtable", "dog",
"horse", "motorbike", "person", "pottedplant", "sheep", "sofa",
"train", "tvmonitor"]
REQ_CLASSES=["bird","cat","cow","dog","horse","sheep"]
COLORS = np.random.uniform(0, 255, size=(len(CLASSES), 3))

#Load model
print("Loading Model...")
net=cv2.dnn.readNetFromCaffe(protext, model)
print("Starting camera feed...")

vs=cv2.VideoCapture(0, cv2.CAP_DSHOW)
#vs=VideoStream(src=0).start()
time.sleep(2)
fps=FPS().start()
```

Output

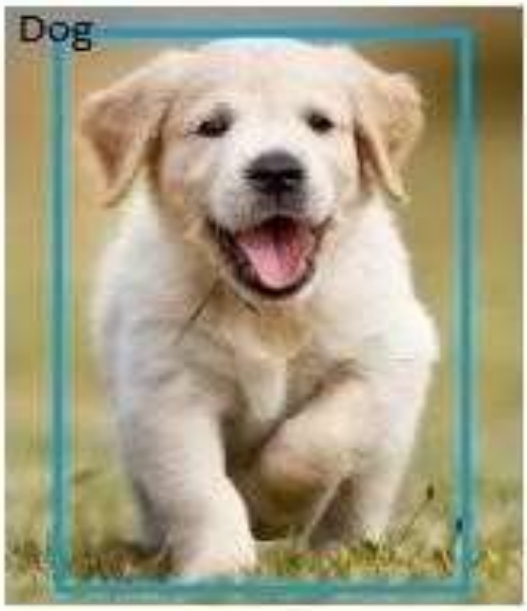


fig 3.3.4.2 Object detected image (dog)

3.3.5 Prediction

A live video feed is taken frame by frame as individual images. These images the then fed into the model after preprocessing to detect animals (if any exists).

```
detections = net.forward()
#frame detection flag
det=0
for i in np.arange(0, detections.shape[2]):
  #Probability associated with predictions
  confidence = detections[0,0,i,2]
  if confidence > conf_thresh:
    #Extract the class labels and dimentions of bounding box
    idx = int(detections[0,0,i,1])
    box = detections[0,0,i,3:7]*np.array([w,h,w,h])
    (startX, startY, endX, endY) = box.astype("int")
```

Output

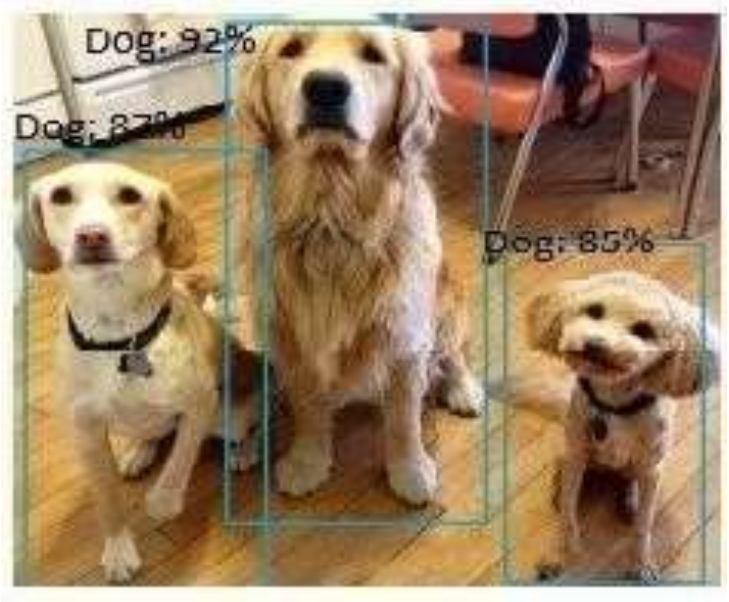


fig 3.3.5 Predicting the object is animal or not

3.3.6 User Interface

A dialog box opens while taking in the live video feed. The frames or images from the video are used to detect objects. The objects are then bounded in a bounding box along with a label and the probability of success is also displayed in there. A siren is then played if any animal is detected for a while.

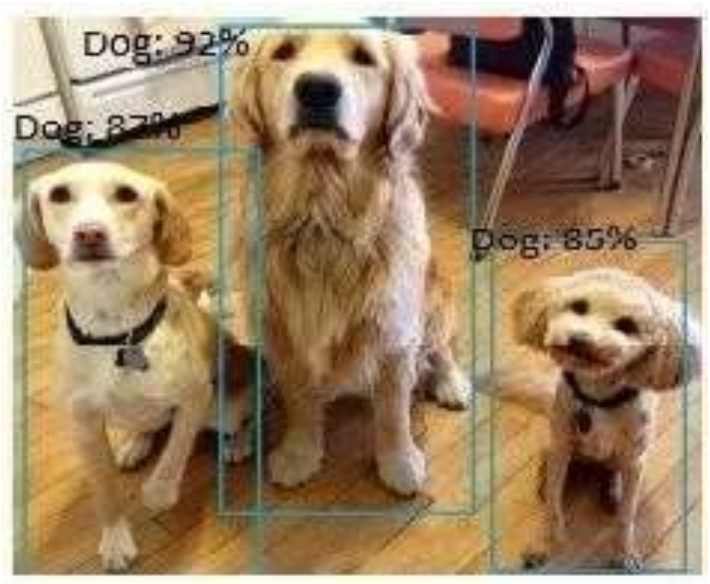
```
#Alerts only if at least 15 frames sucessfully detects animals in the last 36 frames
(appx 2 sec)
  #if flag==1 and len(count) > c+36:
  # flag=0
  if flag==1 and len(count) > c +(11*18):
    flag=0
  if Counter(count[len(count)-36:])[1] > 15 and flag==0:
    print(f"Animal Intrusion Alert...!!! {len(count)}")
    playsound.playsound(siren_loc, block=False)
    flag=1
    c=len(count)

  key = cv2.waitKey(1)
```

```
  if key == ord("q"):
      break
  fps.update()

fps.stop()
print("Elapsed time: {:.2f}".format(fps.elapsed()))
print("Approximate FPS: {:.2f}".format(fps.fps()))
vs.release()
cv2.destroyAllWindows()
#vs.stop()
```

Output



3.3.6 fig Siren played and Alert given to the farmer

CHAPTER 4 SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

Here in this Architecture a camera will start the work which is placed in different locations in the farm (camera or it can be even webcam). Then the live stream is taken as input and through OpenCV the live captured images/videos is checked frame by frame. The SSD MobileNet which is developed by MS COCO dataset detects animals. If the animal is detected an alert will be given to the farmer if it is not animal, it continues to check images frame by frame until animal is detected.

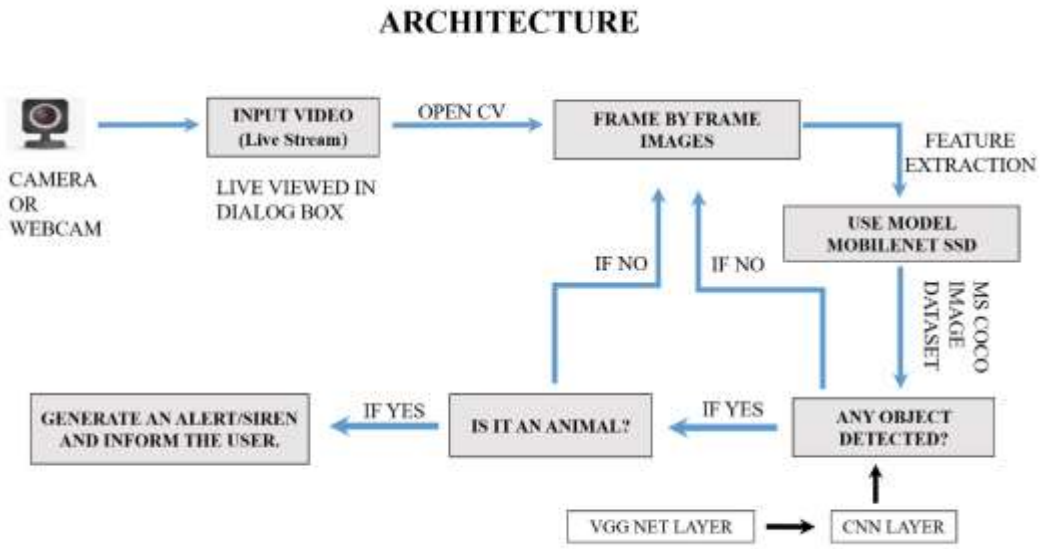


fig 4.1 System Architecture

4.2 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized generalpurpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to or associated with UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

UML is a very important part of developing objects-oriented software and the software development process. UML uses mostly graphical notations to express the design of software projects.

GOALS

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

4.2.1 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

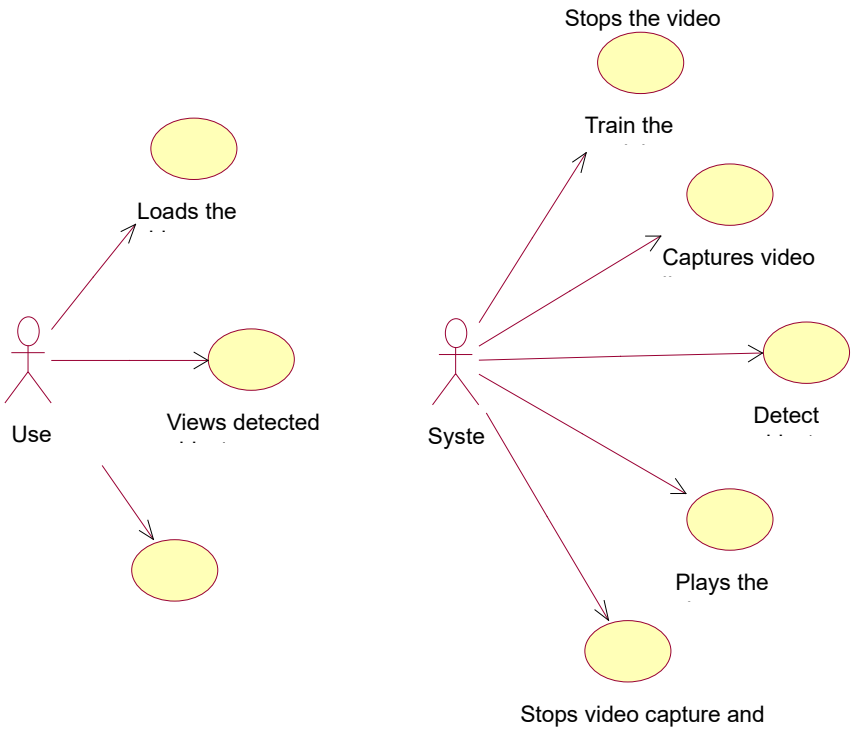


fig 4.2.1 Use Case diagram

4.2.2 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

fig 4.2.2 Class diagram

4.2.3 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

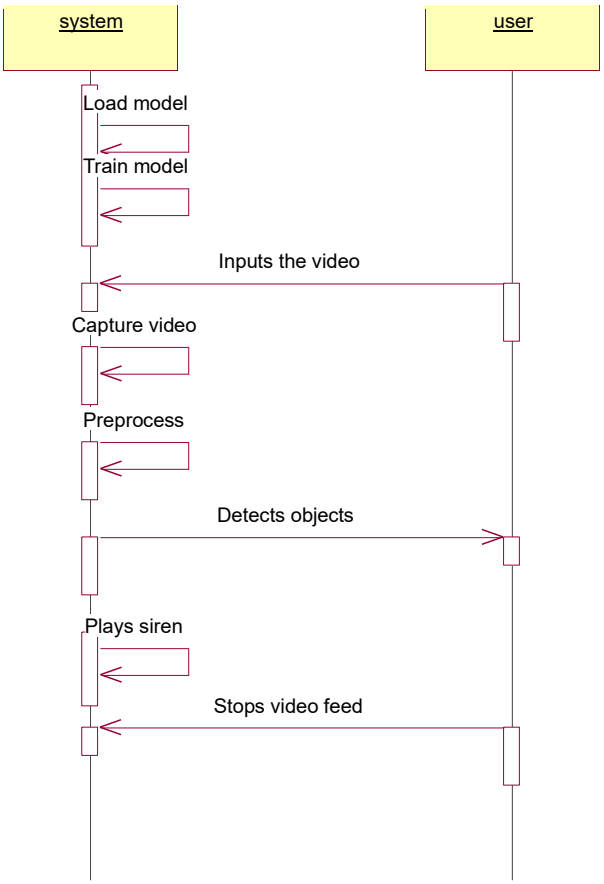


fig 4.2.3 Sequence diagram

4.2.4 COLLABORATION DIAGRAM

In the collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are like that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.

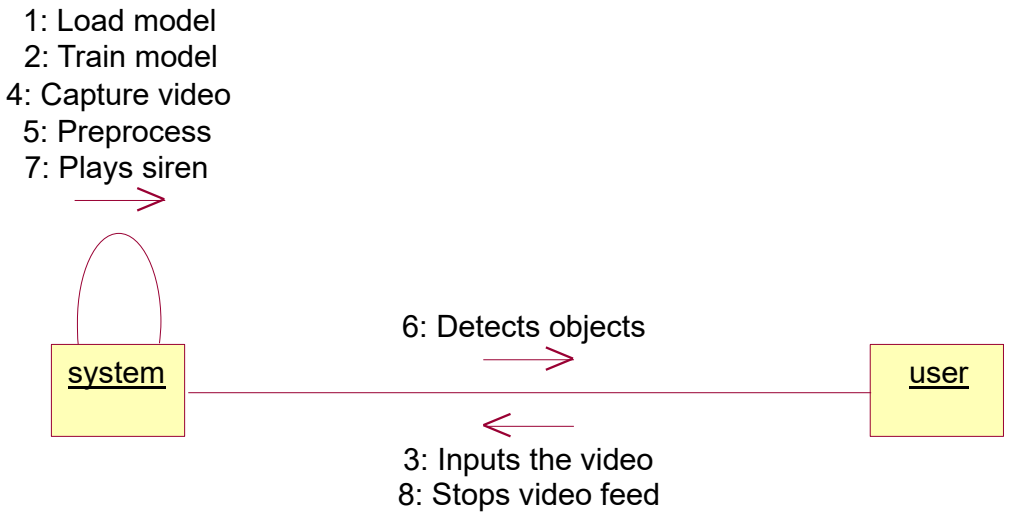


fig 4.2.4 Collaboration diagram

4.2.5 DEPLOYMENT DIAGRAM

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware used to deploy the application.

fig 4.2.5 Deployment diagram

4.2.6 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

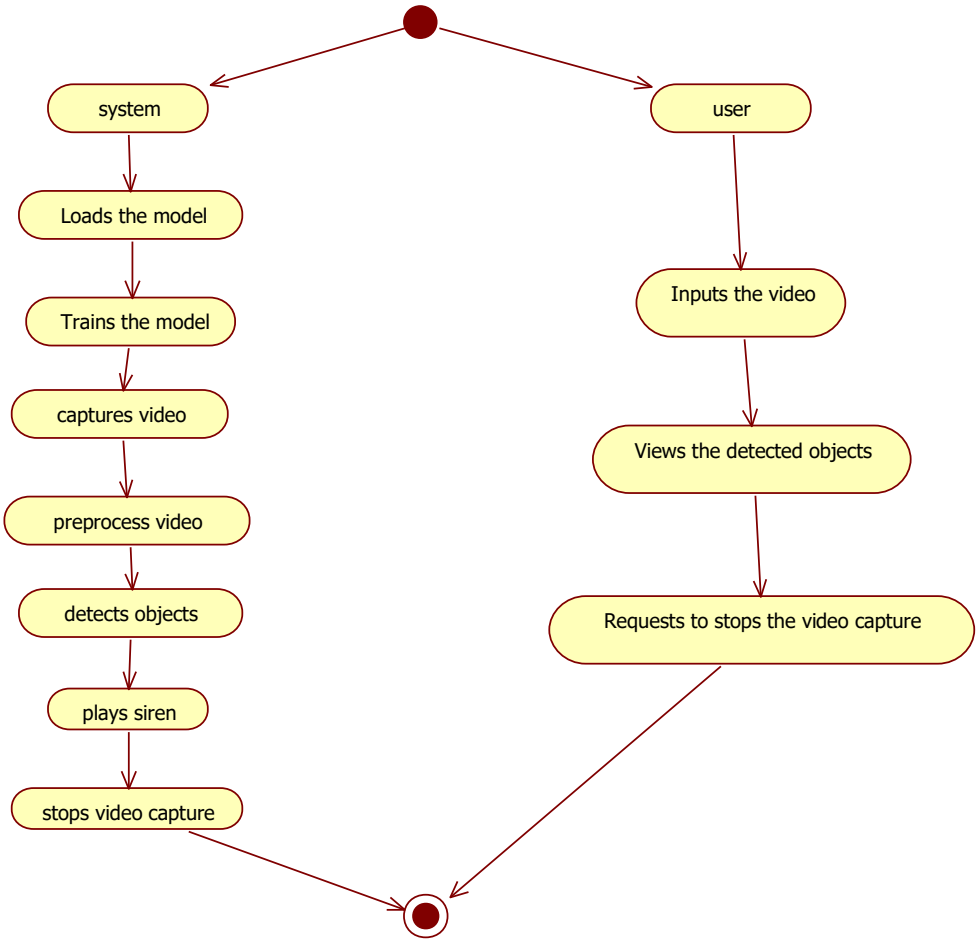


fig 4.2.6 Activity diagram

4.2.7 COMPONENT DIAGRAM

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical **c**omponents in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.

fig 4.2.7 Component diagram

4.2.8 ER DIAGRAM

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let's have a look at a simple ER diagram to understand this concept.

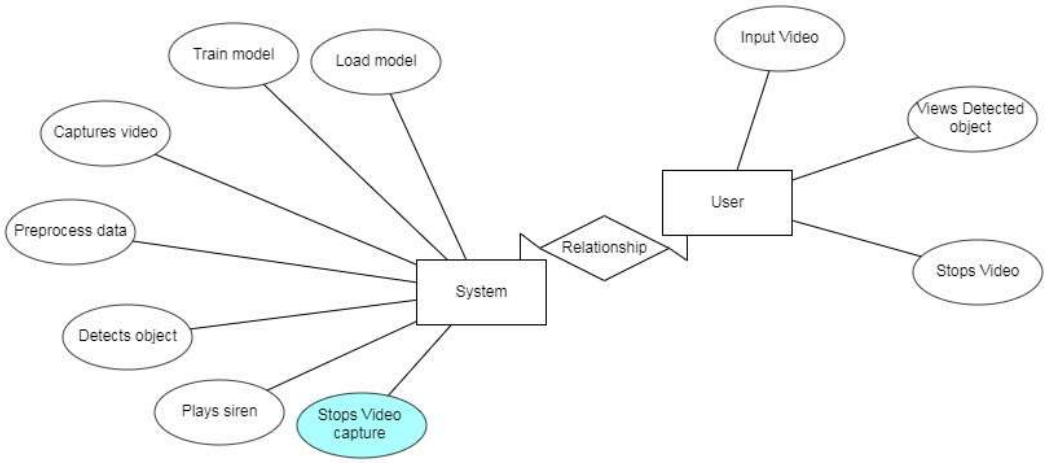


fig 4.2.8 ER-diagram

4.2.9 DFD DIAGRAM

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems

analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

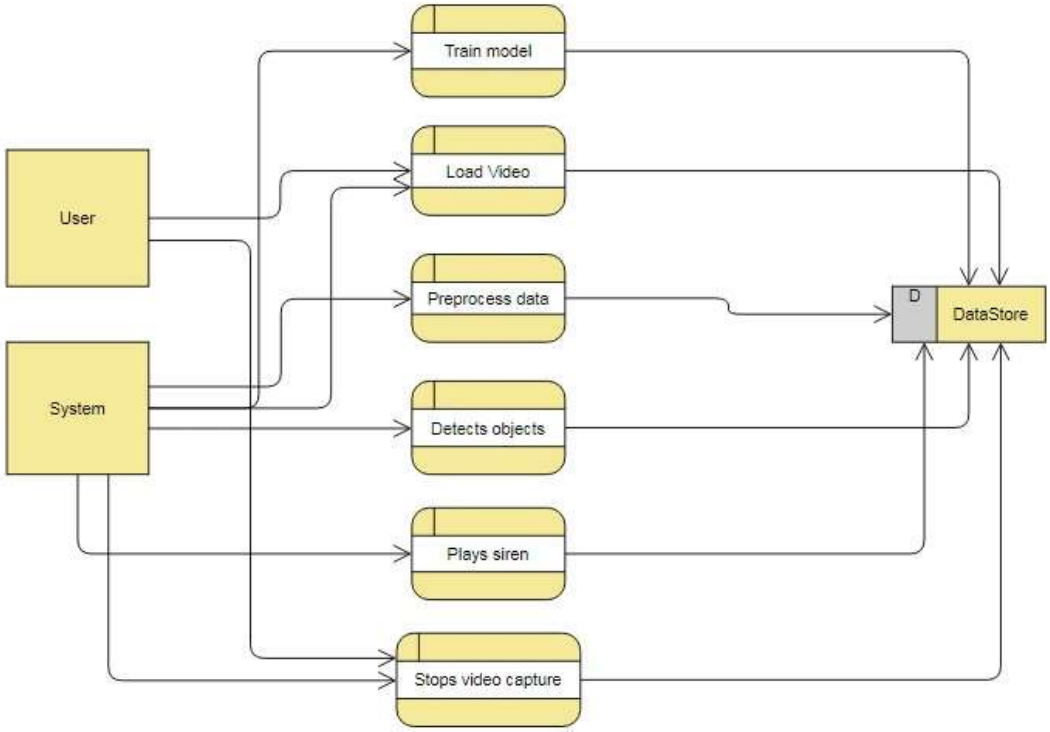


fig 4.2.9 DFD diagram 1

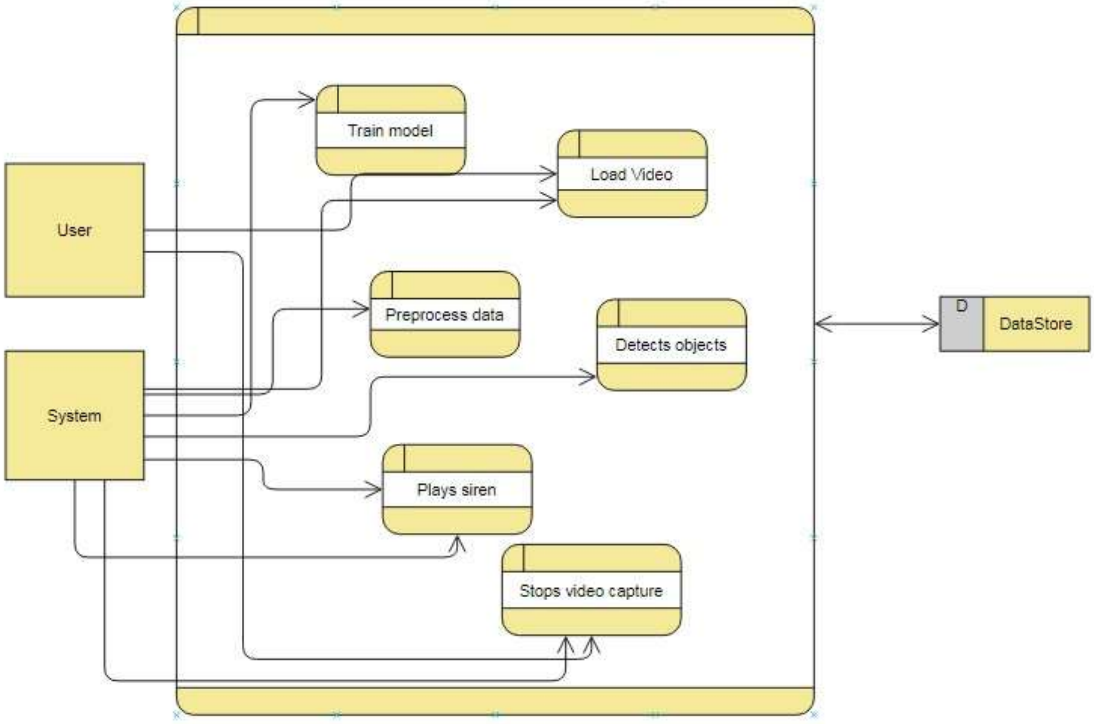


fig 4.2.9 DFD diagram 2

4.3 PROJECT FLOW

The flow for the project is given below:

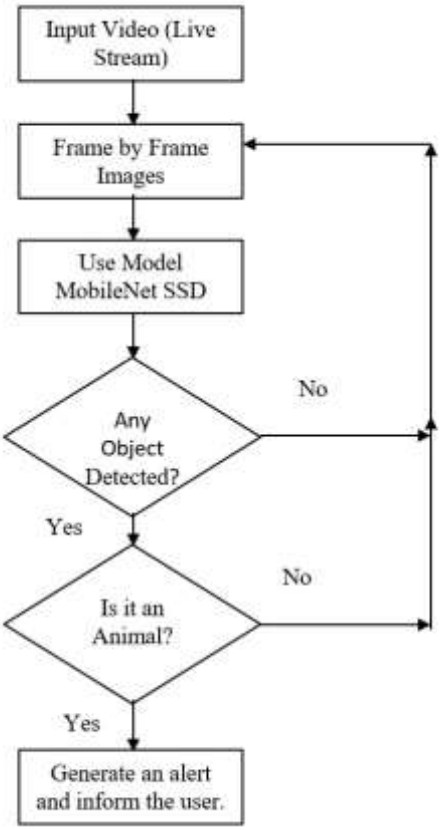


fig 4.3 System Block diagram

CHAPTER 5 SYSTEM IMPLEMENTATION

5.1 INTRODUCTION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

5.2 REQUIREMENTS

HARDWARE Configuration

- RAM - 4GB (min).
- Processor - I3/Intel Processor.
- Hard Disk - 128 GB.
- Key Board - Standard Windows Keyboard.
- Mouse - Two or Three Button Mouse.
- Monitor - Any.

SOFTWARE Configuration

- Operating System : Windows 7+
- Server side Script : Python 3.6+ □ IDE : PyCharm.
- Libraries Used :Numpy,playsound,collections,time,imutils,OpenCV.
- Dataset : MS COCO Image Dataset.

5.3 SOFTWARE INSTALLATION

5.3.1 Installing Python

1. To download and install Python visit the official website of Python and choose your version. https://www.python.org/downloads/ .

fig 5.3.1 Python Home Page

2. Once the download is complete, run the exe for install Python. Now click on Install now.
3. You can see Python installing at this point.
4. When it finishes, you can see a screen that says the Setup was successful. Now click on "Close".

5.3.2 Installing PyCharm

1. To download PyCharm visit the website and Click the "DOWNLOAD" link under the Community Section. https://www.jetbrains.com/pycharm/download/

Fig 5.3.2 PyCharm Download Page

2. Once the download is complete, run the exe for install PyCharm. The setup wizard should have started. Click “Next”.

3. On the next screen, Change the installation path if required. Click “Next”.
4. On the next screen, you can create a desktop shortcut if you want and click on “Next”.
5. Choose the start menu folder. Keep selected JetBrains and click on “Install”.
6. Wait for the installation to finish.
7. Once installation finished, you should receive a message screen that PyCharm is installed. If you want to go ahead and run it, click the “Run PyCharm Community Edition” box first and click “Finish”.
8. After you click on "Finish," the Following screen will appear.

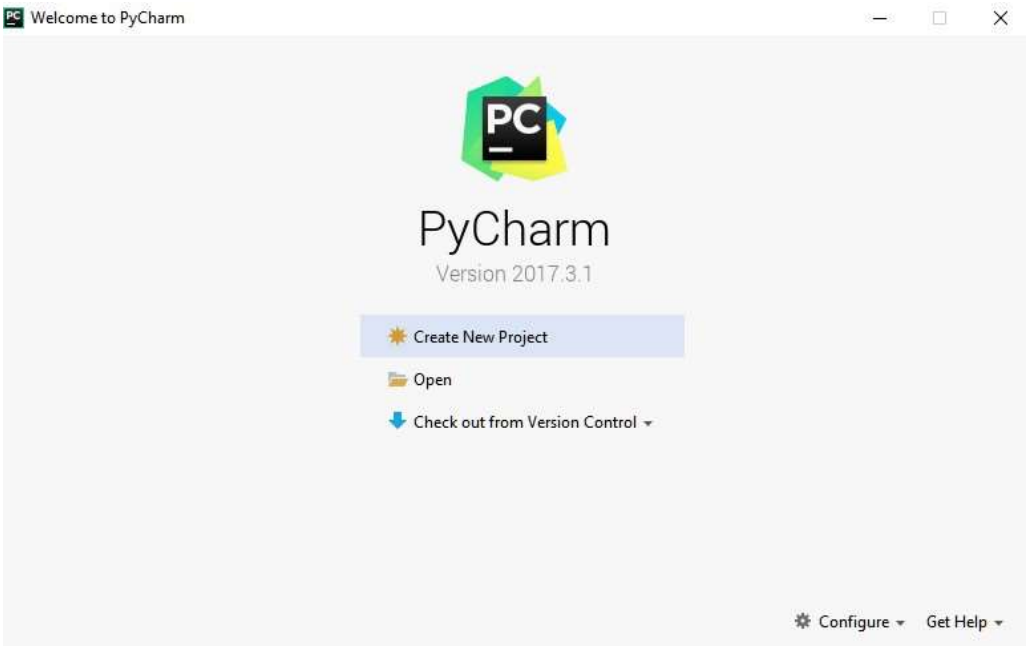


Fig 5.3.2 PyCharm Home Page

9. You need to install some packages to execute your project in a proper way.
10. Open the command prompt/ anaconda prompt or terminal as administrator.
11. The prompt will get open, with specified path, type “pip install package name” which you want to install (like numpy, pandas, seaborn, scikit-learn, matplotlib.pyplot).

Ex: pip install numpy.

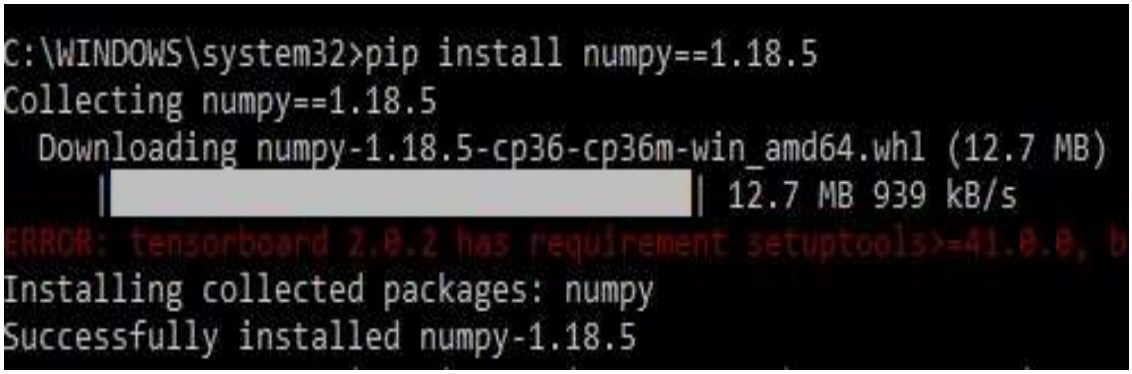


Fig 5.3.2 pip installation

5.4 ALGORITHM USED

5.4.1 Computer Vision

Computer vision is the field of computer science that focuses on replicating parts of the complexity of the human vision system and enabling computers to identify and process objects in images and videos in the same way that humans do.

The problem of computer vision appears simple because it is trivially solved by people, even very young children. Thanks to advances in artificial intelligence and innovations in deep learning and neural networks, the field has been able to take great leaps in recent years and has been able to surpass humans in some tasks related to detecting and labeling objects.

One of the driving factors behind the growth of computer vision is the amount of data we generate today that is then used to train and make computer vision better.

5.4.2 OpenCV

OpenCV is the huge open-source library for computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human.

When it is integrated with various libraries, such as Numpy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its

various features we use vector space and perform mathematical operations on these features.

Using OpenCV library, we can:

- Read and write images.
- Capture and save videos.
- Process images (filter, transform).
- Perform feature detection.
- Detect specific objects such as faces, eyes, cars, in videos or images.
- Analyze the video, i.e., estimate the motion in it, subtract the background, and track objects in it.

5.4.3 Artificial Neural Network:

An artificial neural network is a system of hardware or software that is patterned after the working of neurons in the human brain and nervous system. Artificial neural networks are a variety of deep learning technology which comes under the broad domain of Artificial Intelligence.

Deep learning is a branch of Machine Learning which uses different types of neural networks. These algorithms are inspired by the way our brain functions and therefore many experts believe they are our best shot to moving towards real AI (Artificial Intelligence).

Some types of Neural Networks:

5.4.3.1 Feed forward Neural Network – Artificial Neuron

This is one of the simplest types of artificial neural networks. In a feed forward neural network, the data passes through the different input nodes until it reaches the output node.

In other words, data moves in only one direction from the first tier onwards until it reaches the output node. This is also known as a front propagated wave which is usually achieved by using a classifying activation function.

Unlike in more complex types of neural networks, there is no back propagation and data move in one direction only. A feed forward neural network may have a single layer, or it may have hidden layers.

In a feed forward neural network, the sum of the products of the inputs and their weights are calculated. This is then fed to the output. Here is an example of a single layer feedforward neural network.

5.4.3.2 Radial Basis Function Neural Network

A radial basis function considers the distance of any point relative to the center. Such neural networks have two layers. In the inner layer, the features are combined with the radial basis function.

Then the output of these features is considered when calculating the same output in the next time-step.

The radial basis function neural network is applied extensively in power restoration systems. In recent decades, power systems have become bigger and more complex. This increases the risk of a blackout. This neural network is used in the power restoration systems in order to restore power in the shortest possible time.

5.4.3.3 Multilayer Perceptron (MLP)

A multilayer perceptron has three or more layers. It is used to classify data that cannot be separated linearly. It is a type of artificial neural network that is fully connected. This is because every single node in a layer is connected to each node in the following layer.

A multilayer perceptron uses a nonlinear activation function (mainly hyperbolic tangent or logistic function). This type of neural network is applied extensively in speech recognition and machine translation technologies.

5.4.3.4 Convolutional Neural Network

A convolutional neural network (CNN) uses a variation of the multilayer perceptron. CNN contains one or more than one convolutional layer. These layers can either be completely interconnected or pooled.

Before passing the result to the next layer, the convolutional layer uses a convolutional operation on the input. Due to this convolutional operation, the network can be much deeper but with much fewer parameters.

Due to this ability, convolutional neural networks show very effective results in image and video recognition, natural language processing, and recommender systems.

Convolutional neural networks also show great results in semantic parsing and paraphrase detection. They are also applied in signal processing and image classification.

5.4.3.5 Recurrent Neural Network (RNN) – Long Short-Term Memory

A Recurrent Neural Network is a type of artificial neural network in which the output of a particular layer is saved and fed back to the input. This helps predict the outcome of the layer.

The first layer is formed in the same way as it is in the feed forward network. That is, with the product of the sum of the weights and features. However, in subsequent layers, the recurrent neural network process begins.

From each time-step to the next, each node will remember some information that it had in the previous time-step. In other words, each node acts as a memory cell while computing and carrying out operations. The neural network begins with the front propagation as usual but remembers the information it may need to use later. If the prediction is wrong, the system self-learns and works towards making the right prediction during the back propagation. This type of neural network is very effective in text-to-speech conversion technology.

5.4.3.6 Modular Neural Network

A modular neural network has a few different networks that function independently and perform sub-tasks. The different networks do not really interact with or signal each other during the computation process. They work independently towards achieving the output.

As a result, a large and complex computational process can be done significantly faster by breaking it down into independent components. Computation speed increases because the networks are not interacting with or even connected to each other.

5.4.3.7 Sequence-To-Sequence Models

A sequence-to-sequence model consists of two recurrent neural networks. There's an encoder that processes the input and a decoder that processes the output. The encoder and decoder can either use the same or different parameters. This model is particularly applicable in those cases where the length of the input data is not the same as the length of the output data. Sequence-to-sequence models are applied mainly in chatbots, machine translation, and question answering systems.

Let us consider the basic part of any Neural Network:

5.4.3.7.1 Input Layer

It is the layer where we provide the input for the model. The number of features our input has is equal to the number of neuron in the input layer.

Image Input Layer can create an image input layer using an image input layer. An image input layer inputs images to a network and applies data normalization. Specify the image size using the input Size argument. The size of an image corresponds to the height, width, and the number of color channels of that image. For example, for a grayscale image, the number of channels is 1, and for a color image it is 3.

5.4.3.7.2 Hidden Layer

The input features are transferred to the hidden layer(s) where different processes/activities take place. There can be multiple hidden layers. The layers undergo

mathematical operations like matrix multiplication, convolutions, pooling etc. along with an activation function.

Convolution Layer are the major building blocks used in convolutional neural networks. Convolution is the simple application of a filter to an input that results in an activation. Repeated application of the same filter to an input result in a map of activations called a feature map, indicating the locations and strength of a detected feature in an input, such as an image. The innovation of convolutional neural networks is the ability to automatically learn many filters in parallel specific to a training dataset under the constraints of a specific predictive modeling problem, such as image classification. The result is highly specific features that can be detected anywhere on input images.

Pooling Layer is common to periodically insert a Pooling layer in-between successive Convolution layer in a CNN architecture. Its function is to progressively reduce the spatial size of the representation to reduce the number of parameters and computation in the network, and hence to also control overfitting. The Pooling Layer operates independently on every depth slice of the input and resizes it spatially, using the MAX operation. The most common form is a pooling layer with filters of size 2x2 applied with a stride of 2 down samples every depth slice in the input by 2 along both width and height, discarding 75% of the activations. Every MAX operation would in this case be taking a max over 4 numbers (little 2x2 region in some depth slice). The depth dimension remains unchanged.

Input Sequence Layer is a sequence input layer that inputs sequence data to a network. In the general case, input sequences and output sequences have different lengths (e.g. machine translation) and the entire input sequence is required in order to start predicting the target. This requires a more advanced setup, which is what people commonly refer to when mentioning "sequence to sequence models" with no further context.

Here's how it works: In the general case, input sequences and output sequences have different lengths (e.g. machine translation) and the entire input sequence is required to start predicting the target. This requires a more advanced setup, which is what people commonly refer to when mentioning "sequence to sequence models".

Here's how it works:

- A RNN layer (or stack thereof) acts as "encoder": it processes the input sequence and returns its own internal state. Note that we discard the outputs of the encoder RNN, only recovering the state.

- Another RNN layer (or stack thereof) acts as "decoder": it is trained to predict the next characters of the target sequence, given previous characters of the target sequence.

5.4.3.7.3 OUTPUT LAYER

They layer which is used to generate probability scores using sigmoid or softmax functions which is then converted to the output of our model.

Softmax and Classification Layers converts logits into probabilities by taking the exponents from every output and then norms each of these numbers by the sum of such exponents, such that the entire output vector adds up to one – every probability should be one. Generally, cross-entropy loss is the loss of such a problem in several classes. In the last layer of an image classification network such as CNN (e.g. VGG16) used in ImageNet competitions, softmax is also applied.

A softmax layer applies a softmax function to the input. A classification layer computes the cross-entropy loss for multi-class classification problems with mutually exclusive classes. Create a classification layer using classification Layer. For classification problems, a softmax layer and then a classification layer must follow the final fully connected layer. The softmax function is also known as the normalized exponential and can be considered the multi-class generalization of the logistic sigmoid function. For typical classification networks, the classification layer must follow the softmax layer. In the classification layer, train Network takes the values from the Softmax function.

CHAPTER 6 SYSTEM TESTING

6.1 INTRODUCTION

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

6.2.4 SYSTEM TESTING

System Testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

6.2.5 BLACKBOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, like most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated as a black box you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

6.2.6. WHITEBOX TESTING

White field trying out is a attempting the place wherein the product analyzer is aware of in regards to the internal operations, structure and language of the product, or very likely its motivation. It's cause. It is utilized to test territories that can't be come to from a discovery level.

6.3 TESTING STRATEGY & APPROACH

Testing will be performed manually, and functional tests will be written in detail.

6.3.1 TEST OBJECTIVES

- All field entries must work properly.
- The entry screen, messages and responses must not be delayed.

6.3.2 FEATURES TO BE TESTED

- Verify that the entries are in the correct format.
- No duplicate entries should be allowed.
- Whether the mail is sent to the correct person or not.

6.4 SYSTEM STUDY

6.4.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and the business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are:

6.4.1.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of funds that the company can pour into the research and development of the system is limited. The expenditure must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

6.4.1.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. No system developed must not have a high demand for the available technical resources. This will lead to high demand for the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

6.4.1.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

6.5 TESTCASES

- A Camera will be placed at different locations in the farm (conveniently where animals enter from).
- To process the live video feed the system uses OpenCV (live video - input). The pretrained model, which is trained, is used to detect objects.
- Then the model predicts the object and if the object is an animal, then an alert is sent to the farmer and to divert the animal and siren will be played.

CONDITIONS

- If the pretrained model detects animals, then the result is success.
- If the pretrained model detects humans, then the result is failure.
- If the pretrained model detects any other intrusion, then the result is failure.
-

TESTCASES	INPUT	RESULT	REASON
Case (i)	ANIMAL INTRUSION	SUCCESS	SATISFIED
Case (ii)	HUMAN INTRUSION	FAILURE	UNSATISFIED
Case (iii)	ANY OTHER INTRUSION	FAILURE	UNSATISFIED

Table (6.5) Testcases

6.6 TESTING THE TESTCASES

Case (i) Input

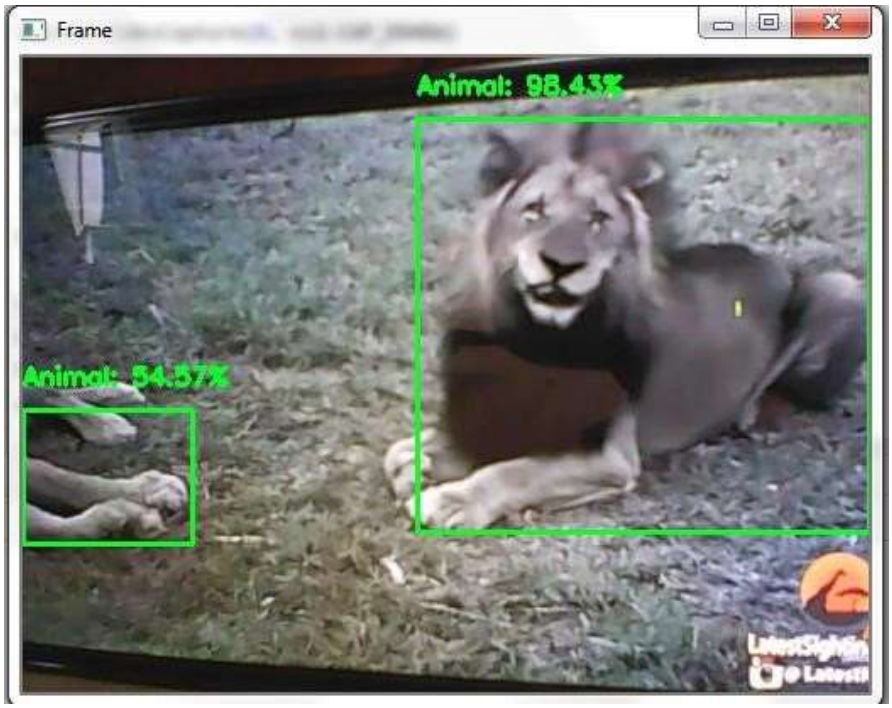


fig 6.6 case (i) Animal Intrusion

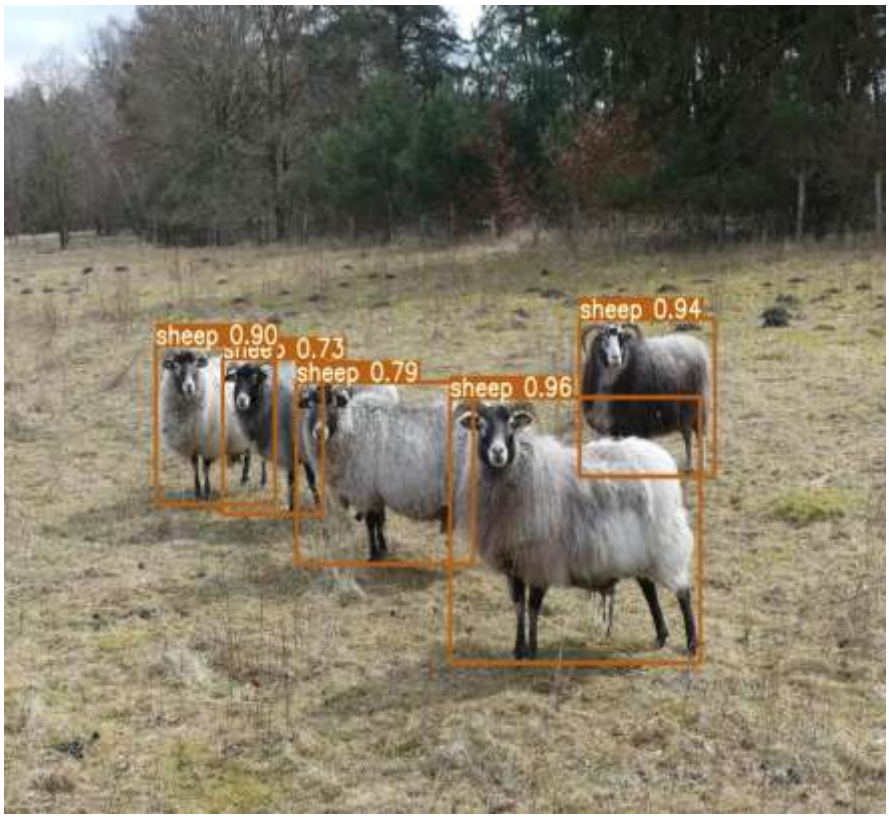


fig 6.6 case (i) Animal Intrusion

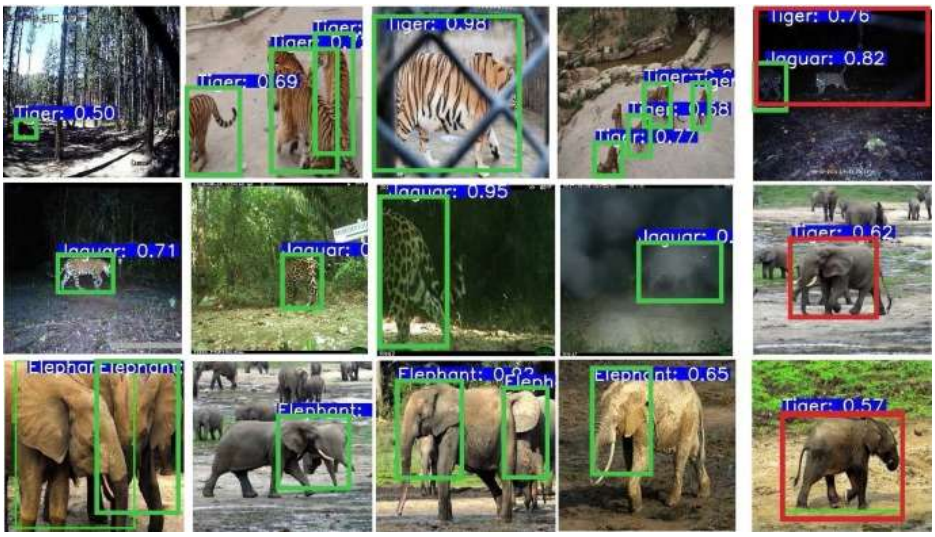


fig 6.6 case (i) Animal Intrusion

fig 6.6 case (i) Animal Intrusion

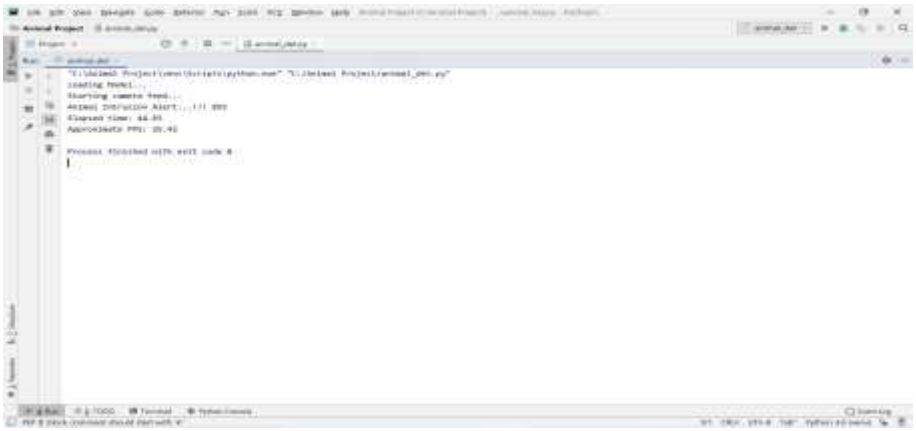
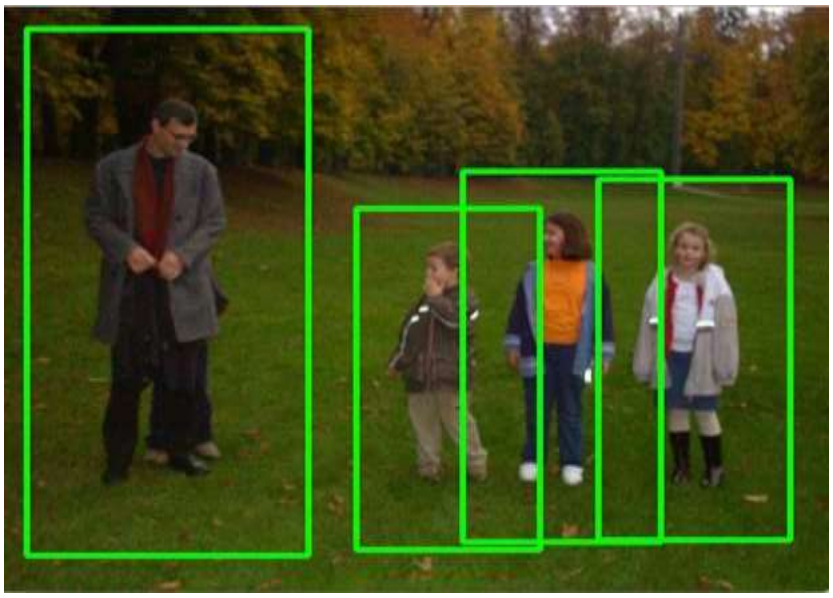


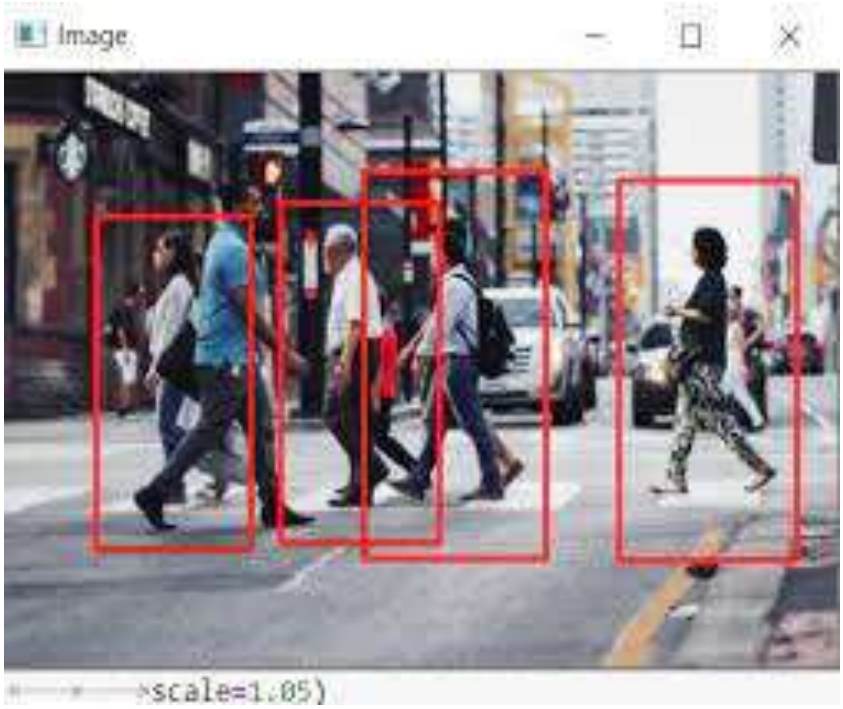
fig 6.6 case (i) Window of Animal Intrusion with Alert

Here, a siren is played in the farm as Animal Intrusion takes place. To distract the Animals before they start the damage. And also, an alert will be generated to the Farmer that "*Animal Intrusion*".

Case(ii) Input



6.6 fig case (ii) Human Intrusion



6.6 fig case (ii) Human Intrusion

Output

Live Stream continues as no animal intrusion takes place. The process continues until the animal is detected.

Case (iii) Input

Fig 6.6 case (iii) Any Other Intrusion

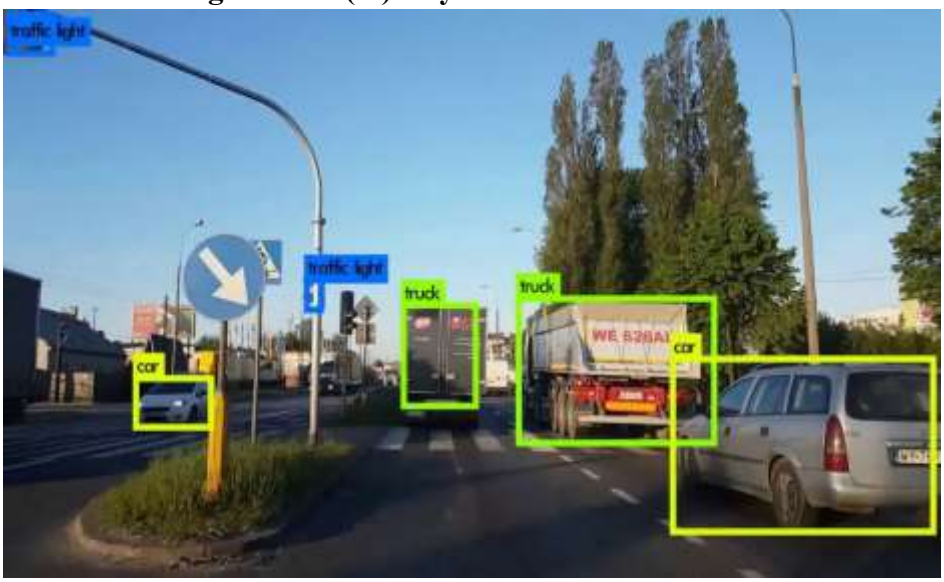


Fig 6.6 case (iii) Any Other Intrusion

Output

Live Stream continues as no animal intrusion takes place. The process continues until the animal is detected.

6.7 EXISTING METHOD PICTURES

The Existing Method is most of the farmers rely on medieval techniques like using:

- Scarecrow.
- Relying on guards to monitor crops.
- Protecting using electric fencing.

Scarecrow

Relying on guards to monitor crops

Protecting using electric fencing

6.8 RESULTS

- The proposed scheme presents a novel approach to detect any intrusion in farms.
- This approach has been implemented by using neural networks.
- We have successfully developed a deep learning model using the deep neural network architecture to detect the presence of any animal in the farm.
- We were able to get a net frame rate of 18 frames per second (approximately).

Comparison on our results with existing and current related work:

In existing farmers is used scarecrow & guards for monitoring crops, but this method won't be helpful in different climatic conditions and moreover time is wasted in this situation. And using electric fencing is a good solution but some government opposes this as illegal, and it may be a threat to wildlife too.

While comparing with our proposed system results it detects animals in any condition like light, weather and so on. And mainly detects the animal and intimates the farmer and if he/she is not in range to farm then it will start a smoke to diver animals.

Comparison on our results with state of art method:

We have selected an advanced algorithm that have been proposed to perform tasks, we will take data from a dialog box that too live data and then that data is cleaned and preprocessed and sent into the model and the model is trained on MobileNet SSD and that will predict whether it is an animal or not. It predicts with good accuracy of 100%. The model can predict animals at any climatic condition that prompts us to the robustness of the system.

CHAPTER 7 CONCLUSION & FUTURE ENHANCEMENTS

7.1 CONCLUSION

The problem of damaging crops by wild animals has become a major social problem in the current time. It requires urgent attention and an effective solution. The proposed method allows us to detect any animal presence or intrusion in farms using video from any camera device placed in the farms. The object detection model worked almost consistently at 18 frames per second.

It is a cheap and robust system. The siren scares the intruders away as well as alerting the farmer to act. Thus, this application can be used to protect crops in the farm. It might be very useful for agricultural purposes instead of traditional methods used today.

In this process while detecting the animals, the surveillance system also detects human as animal if the human behaves like an animal or even acts as any animal behavior. The proposed system achieves an accuracy of 100% regarding animal detection. Thus, it can be concluded from the above discussion that a reliable, secure, fast and efficient system has been developed replacing a manual and unreliable system. Hence this system can be implemented for better results regarding animal detection.

7.2 FUTURE ENHANCEMENTS

The methods to scare away the intruders can be changed in future. We may even employ a laser-based system instead of just a siren to do so. It can also be embedded in an IOT based system for easy utilization.

LEARNING OUTCOMES

- Importance of classification.
- Scope of animal detection.
- Use of computer vision techniques.
- Importance of PyCharm IDE.
- Benefits of Mobile Net model.
- What is OpenCV?
- Need of using pre trained model. Process of debugging a code.
- Input and Output modules.
- How test the project based on user inputs and observe the output.
- Practical exposure to
    - o Hardware and software tools.
    - o Solution providing for real time problems. o Working with team/ individual.
    - o Work on Creative ideas.

BIBLIOGRAPHY

REFERENCES

[1] Parikh, M. et al. “Wild-Animal Recognition in Agriculture Farms Using WCOHOG for Agro-Security.” (2017).

[2] S. Yadahalli, A. Parmar and A. Deshpande, "Smart Intrusion Detection System for Crop Protection by using Arduino," *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, 2020, pp. 405-408, doi: 10.1109/ICIRCA48905.2020.9182868.

[3] Howard, Andrew & Zhu, Menglong & Chen, Bo & Kalenichenko, Dmitry & Wang, Weijun & Weyand, Tobias & Andreetto, Marco & Adam, Hartwig. (2017). "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications."

[4] Deshpande, Abhinav. (2016). Design and Implementation of an Intelligent Security System for Farm Protection from Wild Animals. 5. pp.2319-7064.

- ✓ https://www.ijraset.com/research-paper/animal-detector-system-for-forest-monitoring-using-opencv-and-raspberry-pi
- ✓ https://www.researchgate.net/publication/372165159_ANIMAL_DETECTION_IN_FARMS_USING_OPENCV
- ✓ https://ijsrem.com/download/animal-detection-in-farms-using-opencv/