International Journal of Progressive Research in Science and Engineering
Volume-1, Issue-4, July-2020
www.ijprse.com

Neural Network Based Image Classification for Animal Intrusion Detection System

Priya Sharma [1], Sirisha C K[1], Soumya Gururaj[1], Padmavathi C[2]

[1]Student, Department of Electronics and Communication Engineering, Visvesvaraya Technological University, Bangalore, India

[2]Assistant Professor, Department of Electronics and Communication Engineering, Visvesvaraya Technological University, Bangalore, India.

Corresponding Author: priyaagrp@gmail.com

Abstract: - Every year a large number of resources are lost and human life gets in danger due to human-animal clashing in the forest zone and agriculture fields. People end up losing their livestock, crop, property and sometimes their lives too. To prevent the entry of wild animals in the farm field, it has to be monitored all the time. Here we come with a system which will monitor the field for us. For the first round of surveillance, we have a PIR sensor for motion detection and a video camera to take pictures of the animals. In the next stage, we send these pictures of the animals to the Raspberry Pi for processing. These pictures are processed and classified using deep neural network. Even animal counting is done. Finally, the picture is analyzed to check intrusion of wild animal so that appropriate action can be taken based on the type of animal.

Key Words—Deep learning, Intrusion detection, PIR Sensor, Raspberry Pi.

I. INTRODUCTION

Farm animals or livestock are very important for farmers. Not just for food production but also for functions like making manure or fuel for domestic usage. Hence, a great alternative for non-renewable resources. Livestock is the source of income. The demand for livestock products has increased tremendously in urban areas and it is known as the food revolution. In comparison, the livestock products are costlier than the staple food items in these areas. This problem is also added up by increasing population and hence more food consumption. The food import is increasing to meet the demand for the rising population in the country. But what if we could increase production and become self-sufficient? We will be able to save a lot of foreign exchange. And also the farmers in our country will have improved livelihood and poverty relief. One important work of a herdsman is to count the livestock and it is a redundant job to do. It consumes a lot of time and hence making this task automatic will be a relief and will save a lot of time. The system designed here will be affordable and simple to use for herdsman with small to medium herd size. This will be a viable tool for a herdsman. The automated herd counting tool is not cost-effective solution for rancher or herdsman with a small herd. Every day, the animals are counted as they come in proximity with the counter. The herdsman can put the counter on the higher ground like a water tank and check if the count is correct. If not, they can count it manually or come back again later in the day. This is so simple then to manually drive to the location and interactively count them. Also, the count of the animal can be sent to the herdsman via a message to his/her phone. In recent years, the protection of the human life and crops from the wildlife are very challenging factor. Animals such as wild boars, moles, monkeys and many others which may often -damage the crops by self-feeding or simply by running over the fields. Also, there are instances where human lives are lost due to wild animals' attack. The existing methods like electric fences, creating unpleasant noise and detecting the animal's RFID and LF tags which are injected into animal's skin are proving to be less efficient, as few methods compromise on the safety of wildlife and few on the human lives. To better understand animal behavior, humans have come up with many algorithms. From these studies, we can make a warning system which can detect intrusion of wild animals in the field for early precaution measures. We can make three applications; animal detection, animal identification and alerting system. In this paper, the motion detection and tracking of animals can be controlled by Raspberry Pi system. Using image processing techniques, the animal is detected and the information will be passed on to the nearest forest officers, villagers thereby reducing the attacks on the villagers, crop damage and minimizing the probable loss to the farmers.

II. BASICS AND BACKGROUND

A. Learning through neural network

A neural network works on data like how the human brain works on experience in human life. Hence the accuracy of a neural network depends on the amount of data we provide. The repetition makes the memory stronger. Over time, the result gradually becomes accurate. To train a neural network, we will divide the data into three sets. In the first set, we use the data as training for the network. This will help the network establish various weights between the nodes. We use the next data set as validation data set to fine-tune. The third data set is a test to check how accurate our output is. The training time of a neural network is an important aspect because complex tasks will consume more computer power. One more issue is that the neural networks are like a black box to us. The user feeds in the data and receives answer on the other side. We can fine tune the result, but actual decision making is out of our control. Several researches are working on this problem, but it will become more serious or pressing only if the usage of artificial intelligence increase in our daily life.

B. Convolutional Neural Network

The A convolutional neural network (CNN) comprises one or more convolutional layer and then followed by fully connected layers like a standard multilayer neural network. The architecture of the neural network is suitable for 2D input images (even speech signals). This is achieved with local connections and tied weights followed by some form of pooling which results in translation invariant features. Compared to fully connected networks, training CNN is easier and they have fewer parameters with the same number of hidden units.

A CNN consists of a number of convolutional and subsampling layers optionally followed by fully connected layers. The input to a convolutional layer is a m x m x r image where mm is the height and width of the image and r is the number of channels, e.g. an RGB image has r=3. The convolutional layer will have kk filters (or kernels) of size n x n x q where n is smaller than the dimension of the image and q can either be the same as the number of channels r or smaller and may vary for each kernel. The size of the filters gives rise to the locally connected structure which are each convolved with the image to produce k feature maps of size m−n+1. Each map is then subsampled typically with mean or max pooling over p x p contiguous regions where p ranges between 2 for small images (e.g. MNIST) and is usually not more than 5 for larger inputs. Either before or after the subsampling layer an additive bias and sigmoidal nonlinearity is applied to each feature map. The figure below illustrates a full layer in a CNN consisting of convolutional and subsampling sublayers. Units of the same color have tied weights

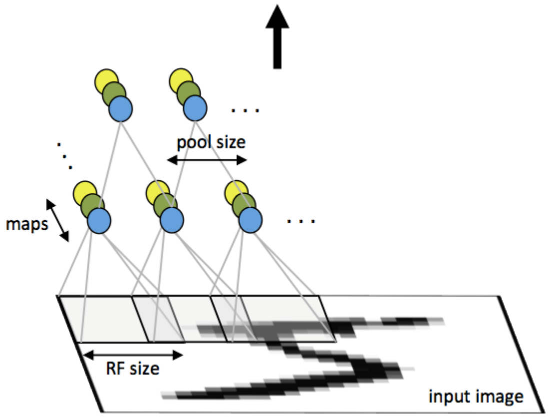


Fig. 1. First layer of convolutional neural network

C. Back Propagation

Let $\delta^{(l+1)}$ be the error for the $(l+1)$ st layer in the network with cost function as $J(W,b;x,y)$ where (W,b) are the parameters and (x,y) are the training data and label pairs. If the l-th layer is densely connected to the $(l+1)$ st layer, then the error for the l-th layer is

$$\delta^{(l)} = \left(\left(W^{(l)}\right)^T \delta^{(l+1)}\right) \cdot f'(z^{(l)})$$

and the gradients will be

$$\nabla W^{(l)} J(W,b;x,y) = \delta^{(l+1)} (a^{(l)})^T$$

$$\nabla b^{(l)} J(W,b;x,y) = \delta^{(l+1)}$$

If the l-th layer is a convolutional and subsampling layer, then the error is propagated through as

$$\delta_k^{(l)} = upsample\left(\left(W_k^{(l)}\right)^T \delta_k^{(l+1)}\right) \cdot f'(z_k^{(l)})$$

Where k is the index of the filter number and $f'(z_k^{(l)})$ is the derivative of the activation function. The up-sample operation passes on the error through the pooling layer. This is successfully bringing about by calculating the error w.r.t to each unit incoming to the pooling layer. For example, if we have mean pooling then up sample will uniformly distributes the error for a single pooling unit among the units which is fed into it in the previous layer. In max pooling, the unit which was chosen as the max will receives all the error because a very small changes in input would change the result only through that unit.

At last, to calculate the gradient w.r.t to the filter maps, we use border handling convolution operation and flip the error matrix $\delta_k^{(l)}$ the same way we flip the filters in the convolutional layer.

$$\nabla W_k^{(l)} J(W,b;x,y) = \sum_{i=1}^{m} (a_i^{(l)}) * rot90(\delta_k^{(l+1)},2)$$

$$\nabla b_k^{(l)} J(W,b;x,y) = \sum_{a,b} (\delta_k^{(l+1)})_{a,b}$$

Where a(l) is the input to the *l*-th layer, and $a^{(1)}$ is the input image. The operation $(a_i^{(l)})*\delta_k^{(l+1)}$ is the “valid” convolution between *i*-th input in the *l*-th layer and the error with respect to the *k*-th filter.

III. Computer Vision

Computer vision is a multidisciplinary discipline of study whose prime aim is helping computers to see. On the whole, it can be called a subpart of machine learning and artificial intelligence. It makes use of general learning algorithms and specialization methods. Since it is a multidisciplinary field of study, it is complex to read because of the borrowed and reused techniques from a variety of contrasting computer science and engineering fields. Each problem in computer vision has a unique approach to solve. One problem may easily be confronted with a hand-crafted statistical method, in contrast another may demand a substantial and complex ensemble of generalized machine learning algorithms. What computer vision needs to do is understand the content of digital images. This involves making or developing methods similar to human brain manner of understanding and solving a problem. By extracting description from a digital image like an object description, a text description, a 3D model etc., the computers can understand the content of an image.

A. OpenCV

OpenCV is the vast open-source library focused on computer vision, machine learning, and image processing. It is known for real-time operation, which is very significant in current system. With openCV, one can analyze images and videos to identify faces, objects and even handwriting belonging to unique humans. When it is integrated with various libraries, such as Numpy, python, it is capable of processing the OpenCV array structure for analysis. We perform mathematical operations and use vector space to identify image pattern and its various attributes.

Released under BSD License, the first version of OpenCV was 1.0. It is free for both commercial and academic use. It supports operating systems like Linux, Mac OS, Windows, Ios and Android and interfaces with languages Python, C++, C and Java. The prime focus for the design of OpenCV was a real-time application for computational efficiency. Everything is written in optimized C/C++ to take advantage of multi-core processing.

IV. Raspberry Pi

A. Raspberry Pi 3 models B+

This new Raspberry Pi 3 Model B+ will benefit designers, developers, and engineers because of its wide range of features and improvements. Engineers can also integrate Pi systems into their products. This raspberry Pi is Proficient in any category, no less than the other Pi computers. Thanks to the ever-improving technological potentiality of electronics. The new models have to either match with old models speed or notably improve upon it. For instance, the CPU has been clocked at 1.4GHz, which is 200MHz faster than the Pi 3 B, and the ethernet speed has been boosted from 100Mbps to 300Mbps. Raspberry Pi 3 B+ model has a Dual-band LAN with modular compliance certification. This is important because electronic products have to go through some tests (CE and FCC) before constructing and releasing them in the market. Mostly these tests look for interference. EMC testing also known as interference testing is costly and difficult to isolate. But thanks to WLAN’s modular compliance certification, the EMCs issue are lower when integrating Pi in the product.

B. Raspbian

Released in 2015, Raspberry Pi OS is a Debian-based Operating system. Raspberry Pi foundation administer this OS as a primary operating system for all the Raspberry Pi single-board computers. Mike Thompson and Peter Green came up with this system first. This was an independent project. The first build was completed in June 2012. Previously, Pi OS was based on Raspbian core, hence got the name Raspbian. They came in 32 bit size. Recent version of Pi OS of size 64 bit does not have Raspbian core, the name has been changed to Raspberry Pi OS. This is for both 64 bit and 32 bit versions.

The Raspberry OS has the firm foundation from Raspbian since it is the official Raspberry OS. This leads to development in features and mobility softwares. With assistance from the raspberry community, it becomes easy to set up this distribution and get going. For this reason, Raspbian also comes pre-installed with office programs, a web-browser, Minecraft and some programming languages (scratch, python, c/c++).

Unlike in heavyweight systems, Processes in Raspbian share come resource during execution. They don’t need to create process specific resources. Hence Raspbians are light and fast.

This increases efficiency and speed of the operating system. The OS speed has increased after acquiring an epiphany-based software (compared to a previous version which runs on Midori). The programs in Raspbians should be written in such a way that they increase efficiency. The OXM command line is a perfect example here.

To find out which is the great software in the pool of good softwares, we should see the ability of the software to be conscious of the end-user. The PI is made to be educational oriented. The Raspbians lets its user to choose the best learning program and tools. In this way the Raspbians becomes powerful than other pi users. Also, it has range from novice style programming languages like python and ruby to simple teachers softwares (scratches). Raspbians always impress with its unique features.

Just as Raspbian OS is easy to use, they are easy to maintain as well. All the items from the repository are updated when we download them. The repository comes with many pre-installed libraries.

V. The Proposed System

First, a database for animal classification is created by capturing pictures of different animals (both wild and domestic animals). This data base can also be enhanced by adding sample pictures from the internet. Now, a program is written with functions such as image set, retrieve image and index image. Image set function is a storage function which contains image collection. The retrieve image function is a search function to look for the image. This is done with the help of index function, which makes an image search index. We give the result of capture image function as a query to the processor. In retrieve image function we have two arguments. One for query image and the other for image stored in the database. This gives the index of the images in the database which looks similar to the query image. These indices are in descending order from the highest matched to the least matched input image with the query image. The matching range is between 0-1. 0 represents they do not match at all. 1 represents both the images are same. The range 0-1 shows query image is in the category of stored images. If the name of the image and normal expression of the image are same, then the animal is a dear otherwise it is a tiger. The value between 0.1-0.9 we understand it matches the image with the stored database image. After the correct identification of a wild animal, the system switches on suitable repellant system. A bright light emitted for elephant, irritating loud noise generator for leopard. Next stage is informing about the intruder to the field owner and also to the forest official through SMS. Also, false alarm is avoided by replying ‘no’ to the message. In case, the animal is not a threat.

A. Methodology

To get correct predictions from deep neural networks, we first need to pre-process our data. These pre-processing tasks normally involve: 1. Mean subtraction 2. Scaling by some factor

OpenCV’s new deep neural network (dnn) module contains two functions that can be used for pre-processing images and preparing them for classification via pre-trained deep learning models.

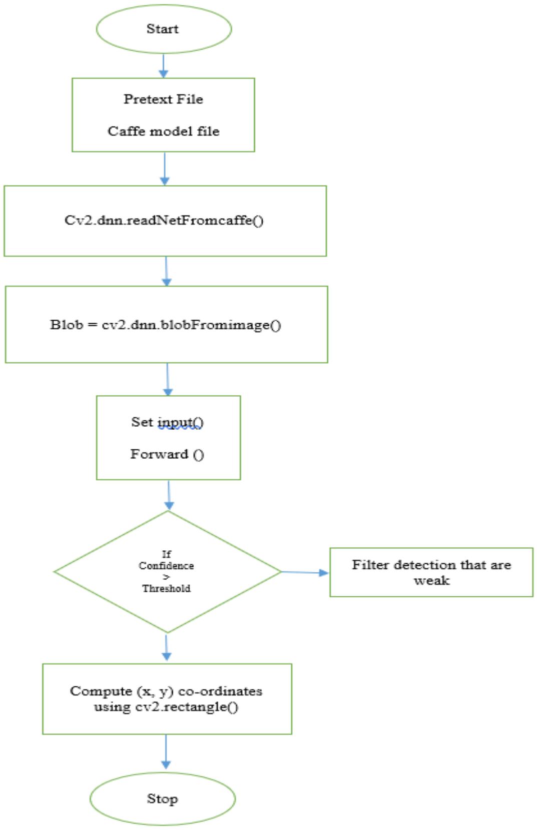

Start
Pretext File
Caffe model file
Cv2.dnn.readNetFromcaffe()
Blob = cv2.dnn.blobFromimage()
Set input()
Forward ()
If
Confidence
>
Threshold
Filter detection that are weak
Compute (x, y) co-ordinates using cv2.rectangle()
Stop

Fig. 2. Flow chart for the proposed system

Algorithm:

1. Image/video acquisition from the camera.
2. Convert video to frames.
3. Store images of each animal as database which is used as training set for our program.
4. Compare camera captured frames with the database.
5. Use ‘imread’ function to read the image and pre-processing is done on that image.
6. Perform Blob detection on the frame and blobs are matched with images from training database images and check if it is matching or not

7. To find out identification of that animal is desired or not, an array is created and program is written for each animal to be identified.
8. To obtain the count- we use if statements to increment count when identified.
9. We obtain the results of Identification and Reckoning of livestock

B. Capturing phase

To detect motion, we first have to capture live images of the area to be monitored and kept under surveillance. This is done by using camera and some motion sensors.

C. Comparing Phase

In comparing phase, it checks two consecutive frames to find out any motion is present or not. We compare the live images captured by the camera to see any changes in the frame. In this way, we can predict occurrence of motion. We place even motion sensors in the field.

D. Pre-Processing

In pre-processing, we apply some methods to extract the required features of an image. It depends on input image type as well. We have denoising which is removing noise in the image. Simple box or Gaussian filter are some denoising techniques. Then we have contrast enhancement in which we correct gray level of the image. Down sampling is for increasing the speed. We have other methods like Morphological operation for binary images, scaling by some factor etc.

E. Image Segmentation

For most application, we are only interested in the certain region of an image. We call the regions as a region of interest or foreground or goals (remaining part is background). We need to separate this region of interest from the image to spot and examine the region of the image. Image segmentation is dividing the image into regions with each region having characteristics and then draw out the target of interest.

Here we are using threshold technique for our image segmentation. In this technique, we choose the threshold value such that we can separate foreground and background based on this grey scale value.

Threshold segmentation has two key steps: 1. Determine the threshold T 2. Compare the pixel value with the threshold value T. Finding out this threshold value is the most important step. When choosing threshold, we can get the best suitable threshold based on original goals of image segmentation. Once we find the appropriate threshold, we can correct the image for segmentation.

F. Hardware Implementation

A camera is used to take pictures of livestock and wild animals to make a database for network training. Also, this camera captures and send images to the Raspberry Pi module for animal identification in real time.

G. Working of the Code

About the libraries used in the code: ‘Numphy’ is a library in python programming language adding support for large multi-dimensional arrays and matrices along with a large collection of high-level mathematical functions to operate on the arrays.

‘Imultis’ is a series of convenience functions to make basic image processing functions such as translation, rotations, resizing, skeletonization, displaying ‘matplotslib’ images, sorting contours, detecting edges and much easier with OpenCV.

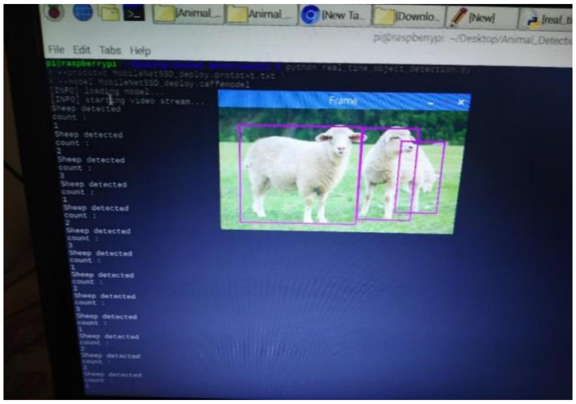
‘Argparse’ module makes it easy to write user friendly command line interfaces. The program defines what arguments it requires and ‘argparse’ will figure out how to parse these out of system arguments. The ‘argparse’ module also automatically generates help and usage messages and issues errors when user gives the program in valid arguments. OpenCV abbreviated as open source computer version is a library with functions that mainly aim real-time computer vision.

Working with openCV: With OpenCV one can perform face detection using pre-trained deep learning face detection model which is shipped with the library. ‘Protext file’ which defines model architecture. ‘Caffemodel’ file which contains the weights for the actual layer. The model is loaded into the net variable using ‘cv2,dnn.readNetfromcaffe()’ function for reading a network model stored in caffe frame work with arguments for “protext” and “model” file path. With cv2.dnn.blobFromImage() function we resize image.

BLOB: Blob stands for binary large object and is used to store information in the database. Blob is a data type that can store binary data. Whereas data types like arrays strings and similar data types are used to store normal data like integers characters, Blob can store multimedia files like images. With cv2.dnn.blobFromImage() function resizing the image to a required resolution can be done. 1.0 is scale factor and here we use default value so there is no scaling after that is spatial size that convolution neural network expects last values are mean subtraction value in tuple and they are RGB means and at the end function returns a “blob” which our input image after resizing, mean subtraction and normalizing. This requires more memory space compared to other data types. Detection: With set Input () we are setting the new input image value for the network. Using forward () function we are running forward pass to compute output of the layer. Then the images are looped through the detections and secondary frames of varying height and width are drawn over the resized image for detection. We are extracting confidence and then comparing the confidence threshold. If the confidence is a minimum threshold, we proceed to draw secondary frames with the probability of detection. Then the images inside the frame are converted to grey scale for comparison. After the detection is confirmed the result is displayed along with the count.

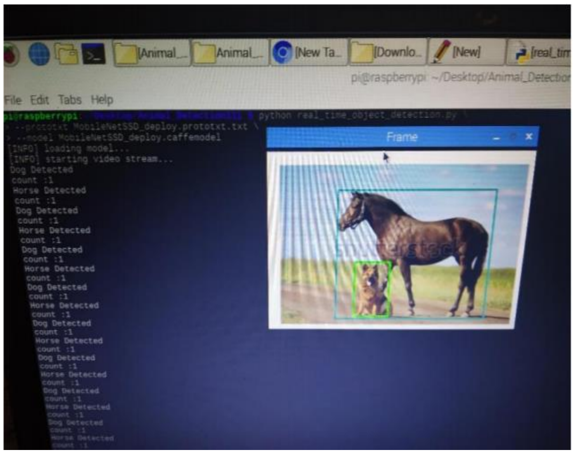
VI. Result

The system designed performs the detection and counting of the livestock.



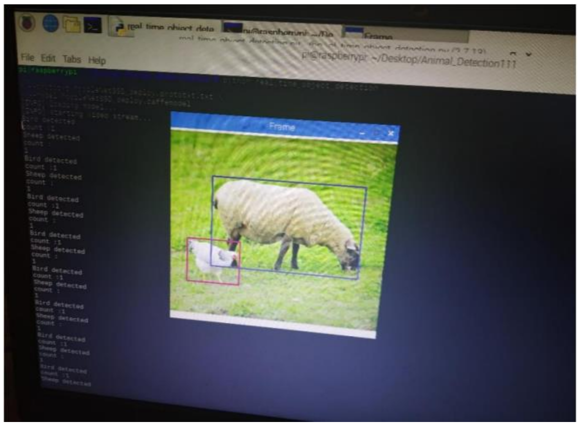
```
pi@raspberrypi:                          $ python real_time_object_detection.py \
> --prototxt MobileNetSSD_deploy.prototxt.txt \
> --model MobileNetSSD_deploy.caffemodel
[INFO] loading model...
[INFO] starting video stream...
Sheep detected
count :
1
Sheep detected
count :
2
Sheep detected
count :
3
Sheep detected
count :
1
Sheep detected
count :
2
Sheep detected
count :
3
Sheep detected
count :
1
Sheep detected
count :
2
Sheep detected
count :
3
Sheep detected
count :
1
Sheep detected
count :
2
Sheep detected
count :
3
Sheep detected
count :
1
```

Fig. 3. Sheep counting and detection



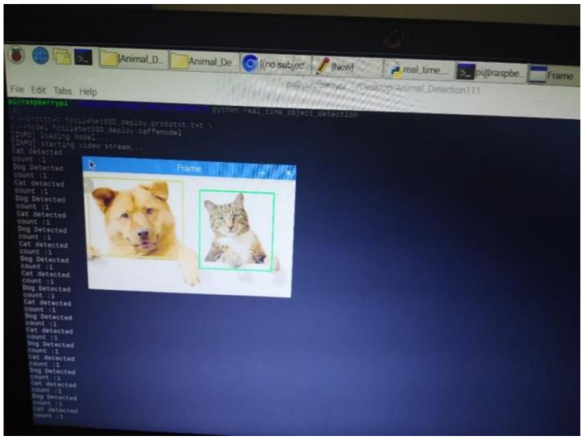
```
pi@raspberrypi:                          $ python real_time_object_detection.py \
> --prototxt MobileNetSSD_deploy.prototxt.txt \
> --model MobileNetSSD_deploy.caffemodel
[INFO] loading model...
[INFO] starting video stream...
Dog Detected
count :1
Horse Detected
count :1
Dog Detected
count :1
Horse Detected
count :1
Dog Detected
count :1
Horse Detected
count :1
Dog Detected
count :1
Horse Detected
count :1
Dog Detected
count :1
Horse Detected
count :1
Dog Detected
count :1
Horse Detected
count :1
Dog Detected
count :1
Horse Detected
count :1
Dog Detected
count :1
Horse Detected
count :1
```

Fig. 4. Horse counting and detection



```
pi@raspberrypi:                          $ python real_time_object_detection
> --prototxt MobileNetSSD_deploy.prototxt.txt
> --model MobileNetSSD_deploy.caffemodel
[INFO] loading model...
[INFO] starting video stream...
Bird detected
count :1
Sheep detected
count :
1
Bird detected
count :1
Sheep detected
count :
1
Bird detected
count :1
Sheep detected
count :
1
Bird detected
count :1
Sheep detected
count :
1
Bird detected
count :1
Sheep detected
count :
1
Bird detected
count :1
Sheep detected
count :
1
Bird detected
count :1
Sheep detected
count :
1
Bird detected
count :1
Sheep detected
```

Fig. 5. Sheep and hen counting and detection



```
pi@raspberrypi:                          $ python real_time_object_detection
> --prototxt MobileNetSSD_deploy.prototxt.txt \
> --model MobileNetSSD_deploy.caffemodel
[INFO] loading model...
[INFO] starting video stream...
Cat detected
count :1
Dog Detected
count :1
Cat detected
count :1
Dog Detected
count :1
Cat detected
count :1
Dog Detected
count :1
Cat detected
count :1
Dog Detected
count :1
Cat detected
count :1
Dog Detected
count :1
Cat detected
count :1
Dog Detected
count :1
Cat detected
count :1
Dog Detected
count :1
Cat detected
count :1
Dog Detected
count :1
Cat detected
count :1
Dog Detected
count :1
Cat detected
count :1
```

Fig. 6. Cat and dog counting and detection

VII. Conclusion

Life of the endangered animals is already threatened and still they are killed in counter-attack or to stop the upcoming disputes. So, such areas should be checked properly to inhibit the accession of wild animals. The proposed system is developed regarding this problem, which will always keep an eye on the field with the help of sensor and camera and later captured an image of the trespasser. Later, appropriate actions can be taken on the basis of the categorization using processed images.

VIII. Future Scope

As a result of the betterment in the portable computers like raspberry pi with respect to the memory, processing speed and networking capabilities, the reliability and performance of this system can be upgraded crucially including the current details of the raspberry pi 3 b+ the intensive usage of resources with the help of a deep learning algorithm which obstruct the work and because of reducing the fidelity to a specific extent and the temperature of the processor extends to a temperature which is unhealthy for the processor itself so accompanied by a preferable cooling system like heatsinks connected to the processor can escalate the performance tremendously because when the processor extends to a threshold temperature it will go through a thermal throttling which lays shuts down the system instantly to save the system from harming itself. By interrelating such systems with a central computer will help in assembling the data and thus the presentation of the entire network is profited by this swapping of data since it will support instruct the algorithm better.

REFERENCES

[1] R.Shanmugasundaram, S.Pavithra, V.Sangeetha, S.Tamilselvan, A.H.Thanveer Ahmed ,"IoT based animal tracking and monitoring system in zoo", South Asian Journal of Engineering and Technology Vol.3, No.2 (2017) 162–168.

[2] Pampapathi B S, Manjunath P C, "Intrusion Detection Using Passive Infrared Sensor (PIR) ", Asian Journal of Engineering and Technology Innovation, Volume 4, Issue 7.

[3] So-Hyeon Kim, Do-Hyeun Kim, Hee-Dong Park, "Animal Situation Tracking Service Using RFID, GPS, and Sensors ", 2010 Second International Conference on Computer and Network Technology, 153 - 156, 2010

[4] Shobhit Kumar Nagpal; P. Manojkumar, &quot;Hardware implementation of intruder recognition in a farm through Wireless Sensor Network & quot;, 2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS), 2016, 1 – 5.

[5] Dr. P. Uma Maheswari and Anjali Rose Rajan, "Animal intrusion detection system using wireless sensor networks", International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST), Vol. 2, Special Issue 10, March 2016.