WILDLIFE WARNING SYSTEM USING DEEP LEARNING

PROJECT REPORT

submitted by

ARTHANA RAVEENDRAN (LTVE16CS066)

GOKULKRISHNA A S (LTVE16CS067)

SARATH KRISHNAN V K (TVE16CS053)

SETHULEKSHMI M (TVE16CS054)

to

the APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of the Degree

of

Bachelor of Technology

in

Computer Science and Engineering

Department of Computer Science and Engineering

College of Engineering, Trivandrum

Kerala

June 20, 2020

DECLARATION

We undersigned hereby declare that the project report **WILDLIFE WARNING SYSTEM USING DEEP LEARNING**, submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under supervision of **Dr. Dhanya S Pankaj (Assistant Professor)**. This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Trivandrum

Date: June 20, 2020

Arthana Raveendran

Gokulkrishna AS

Sarath Krishnan VK

Sethulekshmi M

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COLLEGE OF ENGINEERING, TRIVANDRUM

CERTIFICATE

This is to certify that the report entitled **"WILDLIFE WARNING SYSTEM USING DEEP LEARNING",** submitted by **Arthana Raveendran**, **Gokulkrishna AS**, **Sarath krishnan VK**, **Sethulekshmi M** to the **APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY** in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a bonafide record of the project presented by them under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Dr. Dhanya S Pankaj	**Dr. Piyoosh P**	**Dr. Salim A**
Assistant Professor	Assistant Professor	Professor
Department of CSE	Department of CSE	Department of CSE
(Guide)	(Coordinator)	(Head of Department)

ACKNOWLEDGEMENT

It is with extreme respect that we remember the names of all who had been a great help and guidance throughout our sessions. Firstly, we would like to thank the Almighty for giving us the wisdom and grace for presenting our project. With a profound sense of gratitude, we would like to express our heartfelt thanks to our guide, **Dr. Dhanya S Pankaj**, Assistant Professor, Department of Computer Science and Engineering, for her expert guidance, co-operation and immense encouragement in pursuing this project. We extending our gratitude to our seminar coordinator, **Dr. Saritha R**, Assistant Professor, Department of Computer Science and Engineering and **Dr. Piyoosh P**, Assistant Professor, Department of Computer Science and Engineering, for their co-operation and support. We are very much thankful to **Dr. Salim A**, Head of the Department of Computer Science and Engineering, for providing necessary facilities and his sincere co-operation. Our sincere thanks is extended to all the teachers of the Department of CSE and to all our friends for their help and support.

Arthana Raveendran

Gokulkrishna A S

Sarath Krishnan V K

Sethulekshmi M

ABSTRACT

Nowadays, thousands of animals are killed by road accidents which makes wildlife-vehicle collision a relevant problem in our society. Even though prevention remedies are taken by Government, wildlife accidents are increasing annually. Efficient and reliable monitoring of wild animals is essential for wildlife conservation. By leveraging on recent improvements in machine learning, we aim to implement an automated system for vehicle speed warning and wildlife detection to avoid wildlife-vehicle collision in this project. Night vision cameras installed on roadways and jungle borders are used to capture the animal images. A convolutional neural network architecture is employed to identify the animal. Depending on the type of the animal detected, different warning signals are generated to roadside units, which can help the drivers to take appropriate action. Snapshot Serengeti dataset is used for training the convolutional layers. A database from Kaggle consist of four animals such as tiger, gaur, deer and elephant which are commonly present in the forests of Kerala. Based on size of the animals different output signals are forwarded to the road side units (RSU) which are kept at a fixed distance from the surveillance camera. Road side unit gives warning to the drivers.

Contents

List of Figures

List of Tables

ABBREVIATIONS

CNN Convolutional Neural Network.

ANN Artificial Neural Network.

WVC Wildlife Vehicle Collision.

NH National Highways.

ResNet Residual neural Network.

RSU Road Side Units.

PIR Pyroelectric InfraRed Sensors.

WSN Wireless Sensor Network.

Chapter 1

Introduction

There are many areas such as Bandipur National Park, Sathyamangalam wildlife sanctuary and tiger reserve and Gavi where wild animals cross the roads regularly, especially at night. In some cases, the driver is not able to avoid the wild animal and ends up hitting it. In other cases, the driver is able to avoid the animal, but their evasive maneuvering causes a car accident with another vehicle or with a stationary object like a tree or light pole. Problems get a bit more complex if the crash did not just involve a driver hitting a wild animal. If a driver swerves to avoid the animal and either crashes their car in a single-vehicle accident or into another car to create a multi-vehicle crash, the driver is likely to be found responsible for the accident, and therefore liable for its costs.

Human wildlife conflict is one of the major threat to Indian wildlife, human activities such as deforestation, habitat loss, lack of prey and illegal roads cut through forest are threatening the safety and survival of wildlife in India. Vehicles with high speed in the roads through forest are killing many animals annually mostly chital deer, mouse deer, fox, birds, snakes and nocturnal animals such as Indian civets, black-naped hare and also tiger and leopard. Many wild animals have been killed due to road accidents and speeding vehicles passes through the wildlife protected area. Big animals like sloth bears, striped hyena, blue bull and small creatures such as snakes, monitor lizards and Jackal are getting endangered due to roadkill. Bandipur National Park located in the state of Karnataka, known for its wildlife and has many types of biomes. The largest protected area in southern India has two national highways (NH-67 and NH-212) passing through the park and the road has been a major concern

for wild animals as speeding vehicles have killed many wild animals including tiger, leopard, elephant calf, Indian civet and deer. Muthangha wildlife sanctuary and tiger reserve in Kerala-Karnataka border is a protected area and serves as one of the important wildlife corridor passing through Nilgiri Biosphere Reserve. A night life ban is introduced here a few years back due to concern over the causalities occurring to animals there.

Various modern technologies have been developed for wild animal monitoring,wireless sensor network tracking, satellite including radio tracking, and global positioning system (GPS) tracking, and monitoring by motion sensitive camera traps. Cameras can be used for capturing the images of animals in day and night. So cameras traps are mainly used for monitoring wildlife. A camera can snap thousands of images in large volume which can be used for various purposes. Nowadays machine learning is used to solve many real-world problems. As a prevailing approach to AI, machine learning, in particular deep learning, has drawn considerable attention in recent years due to its astonishing progress. Convolutional neural network (CNN) has been successfully used. In this project, CNN is employed for classification of wildlife. ResNet is used animal recognition classification. ResNet makes it possible to train up to hundreds or even thousands of layer and still achieves compelling performance. ResNet contains more convolution layers than AlexNet, VGG and GoogleNet, therefore it gives more accuracy in image classification. So, ResNet can be used for the image recognition than other CNNs.

In this project, high definition camera traps and CNN are used for giving warning about presence of animals. High definition camera traps capture presence of animals at a particular location if they are found, and CNN based model will classify animals based on their size. The model is able to identify common animals found in Indian forests such as tiger, elephant, gaur and deer. After identifying the type of animal, a proper warning issued through a road side unit fixed at a safe distance from the location of camera.

Chapter 2

Existing Methods

The idea is to prevent wildlife-vehicle collision(WVC) to avoid accidents. Vehicle collisions with animals are dangerous, expensive and a threat to wildlife and people's survival. There are many traditional methods for avoiding the wildlife-vehicle collision. Some of traditional methods are fences, underpass and warning signs. Fences is a traditional method usually used in the forest areas to prevent animals from accessing roads, thereby reducing the rate of WVC. Another existing methods in this area makes use of lamp set and PIR sensors, radars, processors, actuators etc.

2.1 WILDLIFE-VEHICLE COLLISION AVOIDANCE SYSTEM USING LAMP SET AND PYROELECTRIC INFRARED SENSORS

The wildlife-vehicle collision system consists of a lamp set on road sides which has solar panels for gaining energy and PIR sensors to detect the wildlife. Once the wildlife species is detected by the sensor the LED lights attached to the sensor gets ON. The sensed data is transmitted using a RF transmitter to a RF receiver placed inside the vehicle. The PIR sensor can sense approximately from 5 meters to 12 meters distance. No external supply is required as the sensors work on solar energy. This develops a smart and highly sensitive detection system to prevent unnecessary wildlife-vehicle collisions.

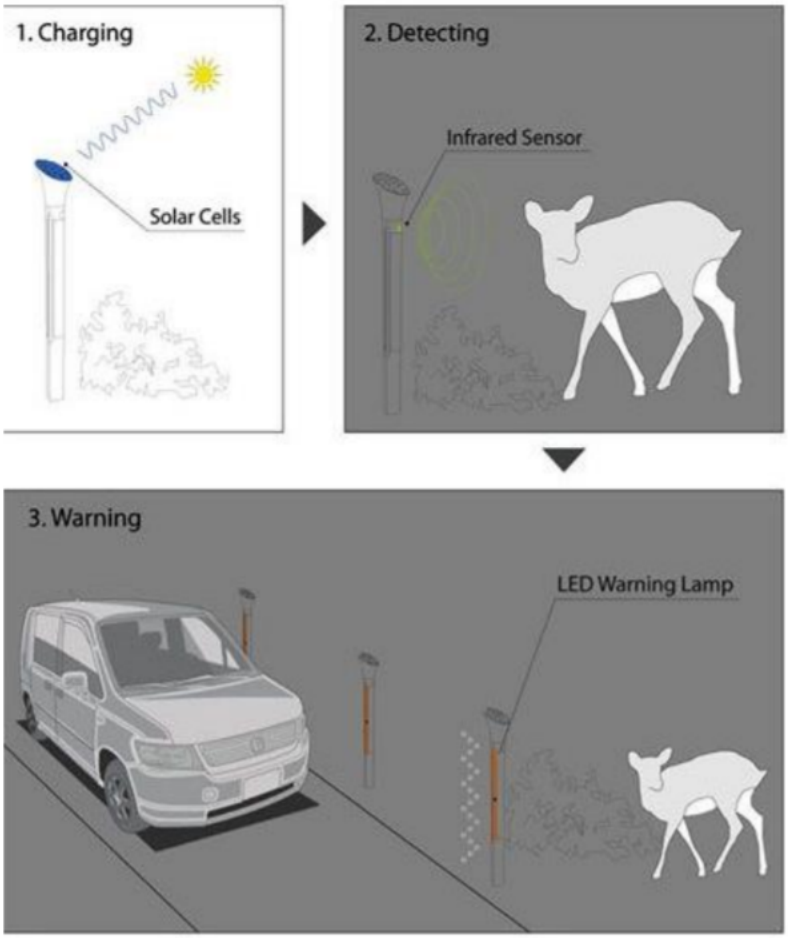


Figure 2.1: LED warning system [3]

2.2 LARGE ANIMAL DETECTION SYSTEM

LADS consist of two parts. First part is to track the large animals approaching and travelling towards the road in the monitored region. This tracking part consist of radar and processors. Radar senses when an animal enter the monitored region. Processors associated will examine the signals by the radar and determine whether the animal is large enough. Second part consist of warning system for the drivers. Driver warning system is achieved using flashing beacons with wireless links when the processor determines whether the animal is large enough. LADS gives high reliability by reducing the probability of producing false alarm. This is achieved by the ability of LADS to distinguish animals and vehicles.

2.3 WSN-BASED ALERT SYSTEM FOR PREVENTING WILDLIFE-VEHICLE COLLISIONS IN ALPS REGIONS

Wireless sensor network(WSN)-based system is composed by network of sensors and actuators for detecting animals approaching the road and light signal devices for warning the drivers. The nodes of the network are installed on the road sides and wirelessly interconnected in order to share the sensed data. The last ones provide real-time alert to the drivers by means of light alarms controlled by actuators. The data acquired by the sensors are sent to the remote control unit that triggers the signaling procedure according with the implemented strategies.

The WSN nodes are interconnected by means of multihop wireless links in order to increase the network coverage and the vigorousness of the system. Each node is connected with a solar panel for energy management and lifetime optimization. The adopted radar modules recognize the movements in the direction of the sensor. The electromagnetic waves in the microwave range are reflected from the moving object and the frequency of the received signal is proportional to the speed. more than that, information about the size and the distance of the target from the sensor can be extracted from the amplitude of the signal. The sensitivity of the sensors can be online adjusted according with the characteristics of the environment.

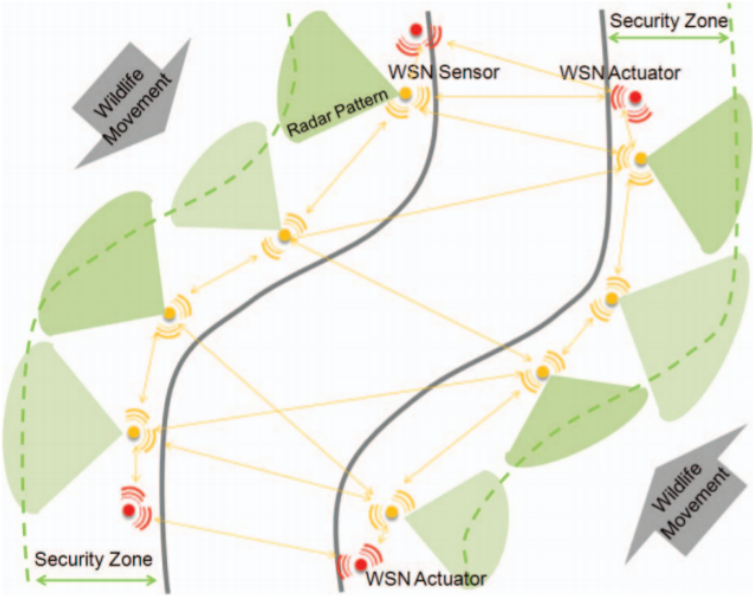


Figure 2.2: WSN of sensors and actuators for wildlife detection [2]

2.4 FENCES

Fences is the most common traditional method used in the forest areas to prevent animals from accessing roads, thereby reducing the rate of WVC.

Figure 2.3: Fences [6]

Chapter 3

Objective

The objective of the project is to implement an automated wildlife detection system using camera traps and provide proper warning to drivers for reducing the rate of dangerous wildlife-vehicle collision.

3.1 WARNING SYSTEM FOR DRIVERS

The images taken by the camera traps at fixed intervals will be fed to CNN based classification software. CNN model identifies the class of animal and a signal will be forwarded to a software present in a road side unit/fixed hardware. This hardware system will be fixed at a safe distance from the camera trap. If dangerous animals such as tiger or elephant is identified by the CNN model, hardware system will display red signal implying driver to stop the vehicle. If the animal is a less dangerous one such as a deer or gaur, yellow signal will be displayed, alerting driver to slow down the vehicle. In the absence of animals, a green light will be displayed.

3.2 WARNING FOR VILLAGERS

Apart from roads passing through forests, the project can be implemented in villages near the forest where presence of animals is often found. The cameras can be placed at border locations through which usually the animal comes to the village, and the warning system can be implemented at a prime spot in village. Instead of light based warning system, output can be in the form of a alerting sound such as that of a siren or horn which will be more effective

in alerting a large number of people even during night. We can also activate some animal repelling systems such as sounds which the animals hate at the location of camera, once the presence of animals is found.

Chapter 4

Design and Implementation

In this section, design and implementation of the proposed system is described. A machine learning system is implemented for animal identification and a hardware system is developed to provide proper warning signal. The system consists of three parts, a camera trap system for taking the picture of wildlife at regular intervals, a Convolutional Neural Network based animal identification software, and a hardware system for giving signals. The system design is detailed in the below section.

4.1 Design

Proposed system consist of three parts, a camera trap system, an Animal identification system and a Warning system.

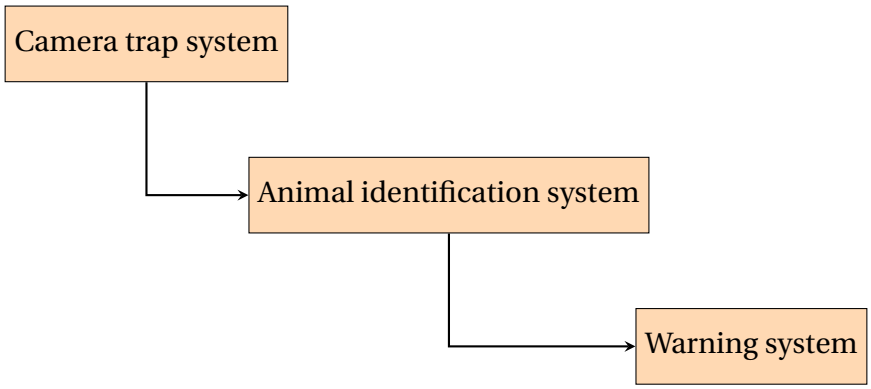


Figure 4.1: Block diagram

4.1.1 Camera trap system

The first part of proposed system consists of a camera trap system, which is used for capturing the images of animals in the road. A camera trap is a remotely activated camera that is equipped with a motion sensor or an infrared sensor, or uses a light beam as a trigger. Camera traps can be left in the field to continuously watch an area of habitat for weeks or even months. Infrared night vision camera is used for continuous image capturing. The cameras used for camera traps are high Resolution 2 Megapixel, 1080P, HD ONVIF Day Night IP IR camera. Many of these cameras are available with optional image stabilization.

At present, images are taken from a wildlife monitoring database named snapshot Serengeti and various databases available in kaggle. Snapshot Serengeti data set includes approximately 2.65 million sequences of camera trap images, totaling 7.1M images, from seasons one through eleven of the Snapshot Serengeti project, the flagship project of the Snapshot Safari network.

Figure 4.2: Camera Trap [6]

4.1.2 Animal identification system

The second part consists of a animal identification software using CNN, which is installed on a raspberry pi. The software takes images as input from the camera and will check the

presence of animals in the images. If the software detects any animal then it will classify the animal in the images and will produce corresponding information and send it to the roadside unit. We use Residual Network (ResNet) as CNN model. ResNet model has more convolutional layers which gives more accuracy than CNNs such as AlexNet had only 5 convolutional layers, the VGG network and GoogleNet has 19 and 22 layers respectively.

4.1.2.1 CNN

In deep learning, a CNN, or ConvNet is a class of deep neural networks, most commonly applied for analyzing image data. They have applications in image and video recognition, recommender systems, image classification, medical image analysis, natural language processing etc.

CNNs are regularized versions of multi layer perceptrons. Multi layer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "fully-connectedness" of these networks makes them prone to over fitting data. Typical ways of regularization include adding some form of magnitude measurement of weights to the loss function. CNNs take a different approach towards regularization, they take advantage of the hierarchical pattern in data and assemble more complex patterns using smaller and simpler patterns. Therefore,on the scale of connectedness and complexity, CNNs are on the lower extreme.

Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they over the entire visual field.

CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage.

4.1.2.2 Architecture of CNN

A convolutional neural network consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of a series of convolutional layers that convolve with a multiplication or other dot product. The activation function is commonly a RELU layer, and is subsequently followed by additional convolutions such as pooling layers, fully connected layers and normalization layers, referred to as hidden layers because their inputs and outputs are masked by the activation function and final convolution.

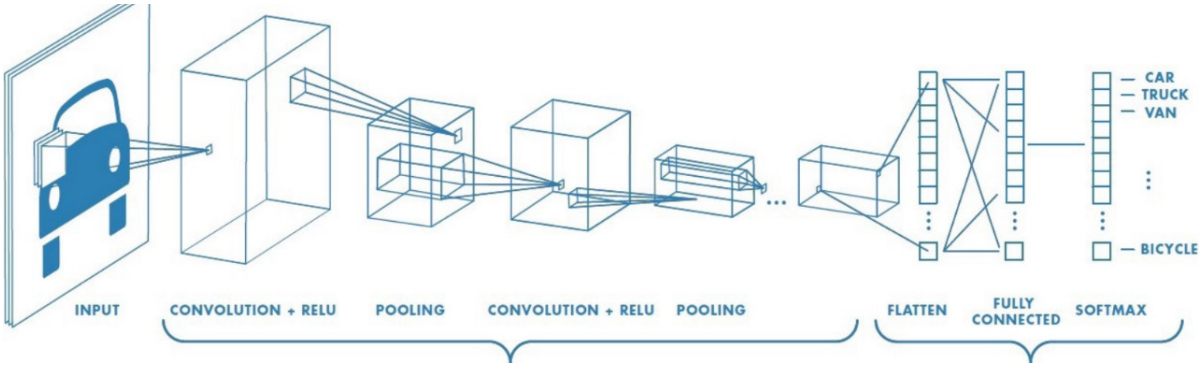


Figure 4.3: Architecture of CNN [6]

Though the layers are colloquially referred to as convolutions, this is only by convention. Mathematically, it is technically a sliding dot product or cross-correlation. This has significance for the indices in the matrix, in that it affects how weight is determined at a specific index point.

Convolutional layers

When programming a CNN, the input is a tensor with shape (number of images) x (image height) x (image width) x (image depth). Then after passing through convolutional layer, the image becomes abstracted to a feature map, with shape (number of images) x (feature map height) x (feature map width) x (feature map channels). A convolutional layer within a neural network should have the following attributes:

1) Convolutional kernels defined by a width and height (hyper-parameters), 2) The number of input channels and output channels (hyper-parameter), 3) The depth of the Convolution filter (the input channels) must be equal to the number channels (depth) of the input

feature map. Convolutional layers convolve the input and pass its result to the next layer. This is similar to the response of a neuron in the visual cortex to a specific stimulus. Each convolutional neuron processes data only for its receptive field. Although fully connected feed-forward neural networks can be used to learn features as well as classify data, it is not practical to apply this architecture t images. A very high number of neurons would be necessary, even in a shallow architecture, due to the very large input sizes associated with images, where each pixel is a relevant variable. For instance, a fully connected layer for a (small) image of size 100 x 100 has 10,000 weights for each neuron in the second layer. The convolution operation brings a solution to this problem as it reduces the number of free parameters, allowing the network to be deeper with fewer parameters.[13] For instance, regardless of image size, tiling regions of size 5 x 5, each with the same shared weights, requires only 25 learnable parameters. By using regularized weights over fewer parameters, the vanishing gradient and exploding gradient problems seen during back propagation in traditional neural networks are avoided.

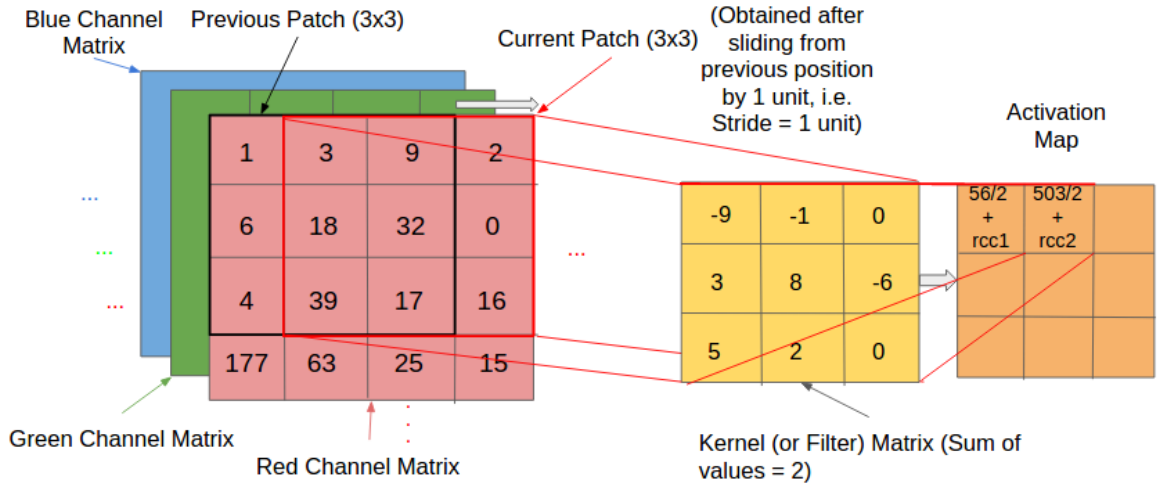


Figure 4.4: Convolutional layers [6]

Pooling

Convolutional networks may include local or global pooling layers to streamline the underlying computation. Pooling layers reduce the dimensions of the data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. Local pooling

combines small clusters, typically 2 x 2. Global pooling acts on all the neurons of the convolutional layer. In addition, pooling may compute a max or an average. Max pooling uses the maximum value from each of a cluster of neurons at the prior layer. Average pooling uses the average value from each of a cluster of neurons at the prior layer.

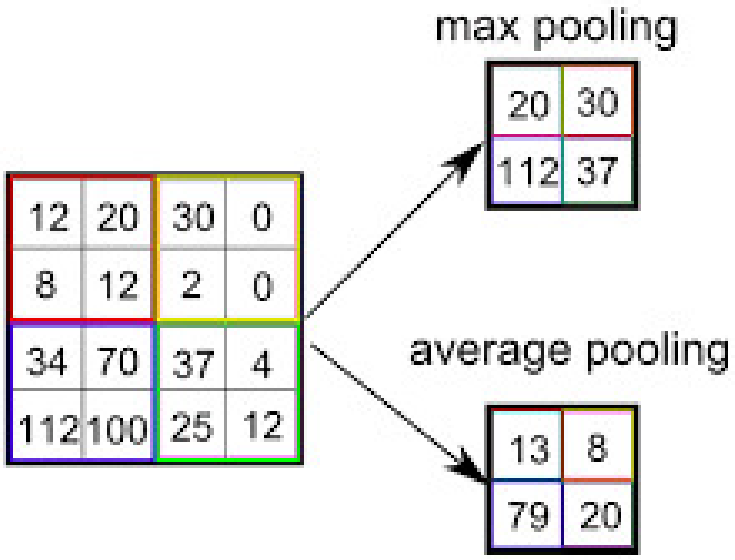


Figure 4.5: Pooling [6]

Fully connected layers

Fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the traditional multi-layer perceptron neural network (MLP). The flattened matrix goes through a fully connected layer to classify the images.

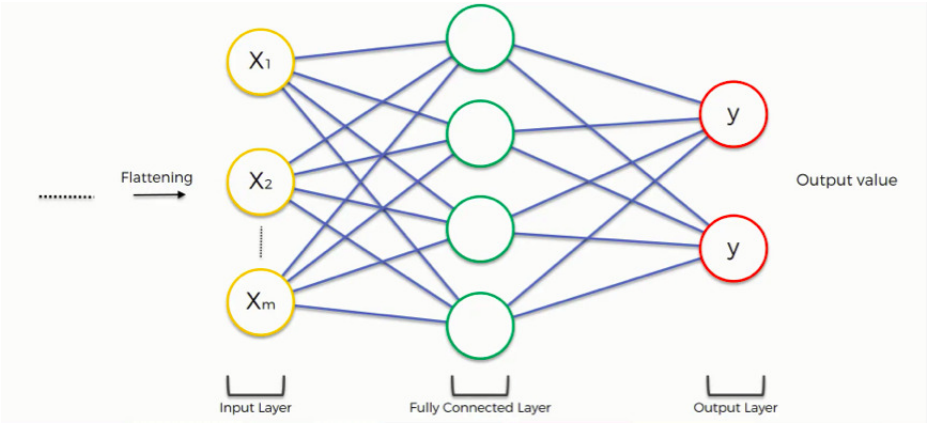


Figure 4.6: Fully connected layers [6]

Receptive field

In neural networks, each neuron receives input from some number of locations in the previous layer. In a fully connected layer, each neuron receives input from every element of the previous layer. In a convolutional layer, neurons receive input from only a restricted subarea of the previous layer. Typically the subarea is of a square shape. The input area of a neuron is called its receptive field. So, in a fully connected layer, the receptive field is the entire previous layer. In a convolutional layer, the receptive area is smaller than the entire previous layer. The subarea of the original input image in the receptive field is increasingly growing as igetting deeper in the network architecture. This is due to applying over and over again a convolution which takes into account the value of a specific pixel, but also some surrounding pixels.

Weights

Each neuron in a neural network computes an output value by applying a specific function to the input values coming from the receptive field in the previous layer. The function that is applied to the input values is determined by a vector of weights and a bias. Learning, in a neural network, progresses by making iterative adjustments to these biases and weights.

The vector of weights and the bias are called filters an represent particular features of the input. A distinguishing feature of CNNs is that many neurons can share the same filter.

This reduces memory footprint because a single bias and a single vector of weights are used across all receptive fields sharing that filter, as opposed to each receptive field having its own bias and vector weighting.

4.1.2.3 ResNet

A ResNet is an artificial neural network (ANN). Resnet is a convnet which contains millions of parameters and many hidden layers. Resnet implement neural networks by utilizing skip connections, or shortcuts to jump over some layers. Typical ResNet models are implemented with double or triple layer skips that contain nonlinearities (ReLU) and batch normalization in between. An additional weight matrix may be used to learn the skip weights. These models are known as HighwayNets. Models with several parallel skips are referred to as DenseNets. In the context of residual neural networks, a non-residual network may be described as a plain network.

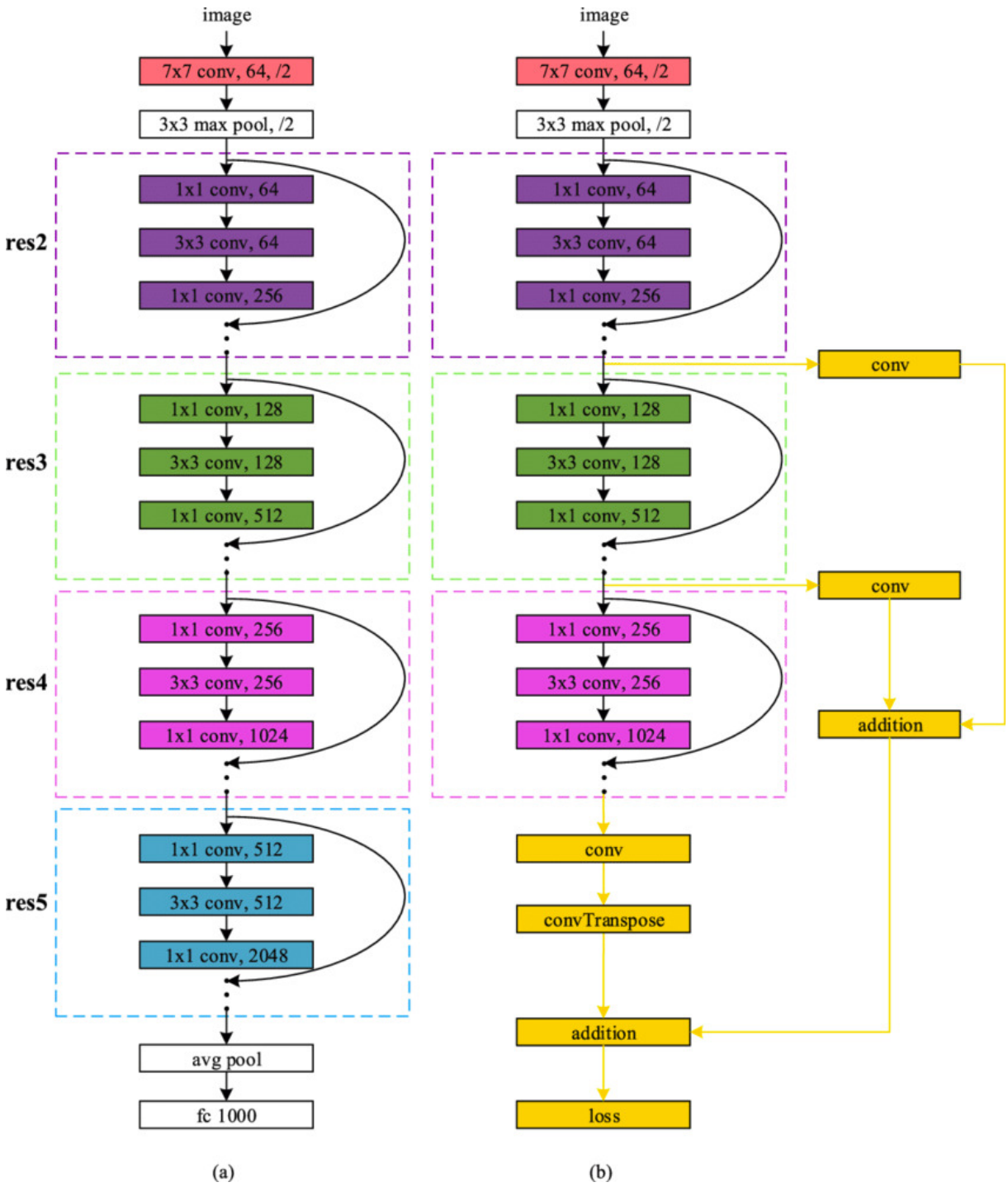


Figure 4.7: Resnet [6]

One motivation for skipping over layers is to avoid the problem of vanishing gradients, by reusing activations from a previous layer until the adjacent layer learns its weights. During training, the weights adapt to mute the upstream layer[clarification needed], and amplify the previously-skipped layer. In the simplest case, only the weights for the adjacent layer's connection are adapted, with no explicit weights for the upstream layer. This works best when a single nonlinear layer is stepped over, or when the intermediate layers are all linear. If not, then an explicit weight matrix should be learned for the skipped connection.

Skipping effectively simplifies the network, using fewer layers in the initial training stages. This speeds learning by reducing the impact of vanishing gradients, as there are fewer layers to propagate through. The network then gradually restores the skipped layers as it learns the feature space. Towards the end of training, when all layers are expanded, it stays closer to the manifold and thus learns faster. A neural network without residual parts explores more of the feature space. This makes it more vulnerable to perturbations that cause it to leave the manifold, and necessitates extra training data to recover.

4.1.3 Warning system

The third part consists of a Road Side Unit (RSU) which receives the signal from the software and which is placed at a fixed distance from the camera. Then according to the size of animals, images can be considered as three classes. Tiger and elephant are considered as first class. Gaur and deer are considered as second class. Small animals and images without any animals considered as third class images. First class animals are considered as harmful animal and second class is considered as less harmful animal. Third class is considered as harmless animal/no presence of animal. The third class images will be incorporated into model at implementation stage based on location of camera, so that weights does not get corrupted. Just like familiar traffic signal system, it consists three colors for giving warning for the drivers. if first class and second class animals are recognised in the input. Warning system displays yellow signal for the second class animals and green signal for the first class animals. Green signal shows safe driving. Yellow signal warns the driver to go slow and the green signal warns the driver to stop the vehicle. This system can be implemented using a simple on board program and a road side unit.

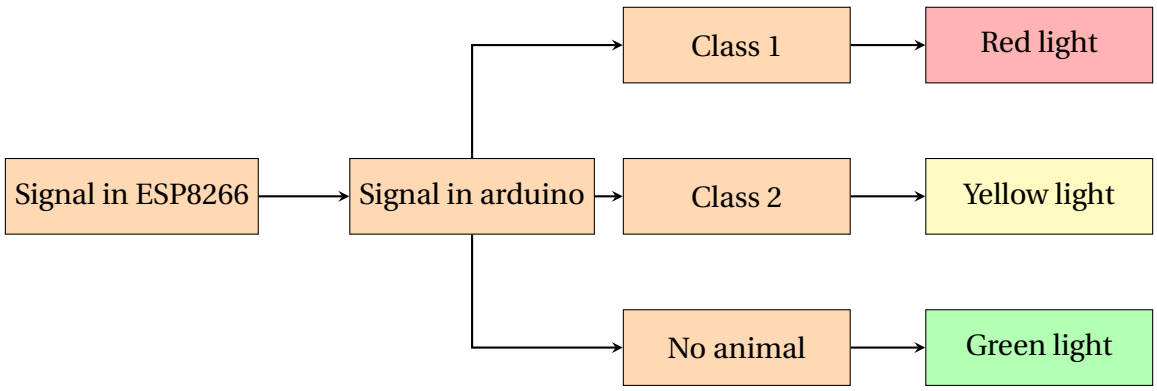


Figure 4.8: Architecture of Warning system

4.2 IMPLEMENTATION

4.2.1 Input dataset

The convolutional layers are trained using images from wildlife conservation project named Snapshot Serengeti which consist of 2.65 million sequences of camera trap images.The dataset for training fully-connected layers are taken from kaggle dataset which consist of four classes that are tiger, elephant, deer and gaur. In the kaggle dataset, class tiger contains 1200 images, class elephant contains 1050 images, class deer contains 1100 images, class gaur contains 1000 image each.

(a) Tiger

(b) Elephant

(a) Deer

(b) Gaur

Figure 4.10: Sample of Dataset [7]

4.2.2 Classifier implementation

4.2.2.1 General implementation details

Anaconda Jupyter Notebook

Anaconda is a free and open-source distribution of the Python and R programming languages, that aims to simplify deployment and package management. The distribution includes data-science packages suitable for various operating systems such as Windows, Linux, and macOS. The Jupyter Notebook is an open-source web application that allows the user to create and share documents that contain code, text visulaiztions etc. Jupyter notebook can be used for tasks such as modeling, data cleaning and transformation task, data visualization etc. Jupyter Notebook is available in Anaconda Prompt and this is the Integrated development environment used.

Figure 4.11: Jupyter Notebook [6]

Python

Python Language is one of the most used language for coding deep learning networks other than R and is used in this project for adding layers(Keras library in python) and general code implementation.

Figure 4.12: Python [6]

Keras

Keras is a deep learning library for python and is compatible with other deep learning libraries such as tensorflow. Layers act as the basic building blocks of neural networks in keras, which contains Convolutional layer, pooling layers, fully connected layers, dense layers. In this project, this library is used for building convolutional, pooling and fully connected layers of Resnet model used for classification.

Figure 4.13: Keras [6]

4.2.2.2 Transfer Learning approach

Transfer learning approach is used for implementing the model. In transfer learning approach, knowledge obtained during one problem will be applied for different but a similar problem. In deep learning, this refers to predictive modeling where weights of some of the layers of a pre trained network is reused in another similar problem to accelerate training by keeping the weights remaining the same , fine tuning them or adapting new weights.In this project, a resnet model is created using using layers in keras and weights of a pretrained resnet model based on a wildlife conservation project named Snapshot Seregetti which is trained over 2.63 million images is added to it.

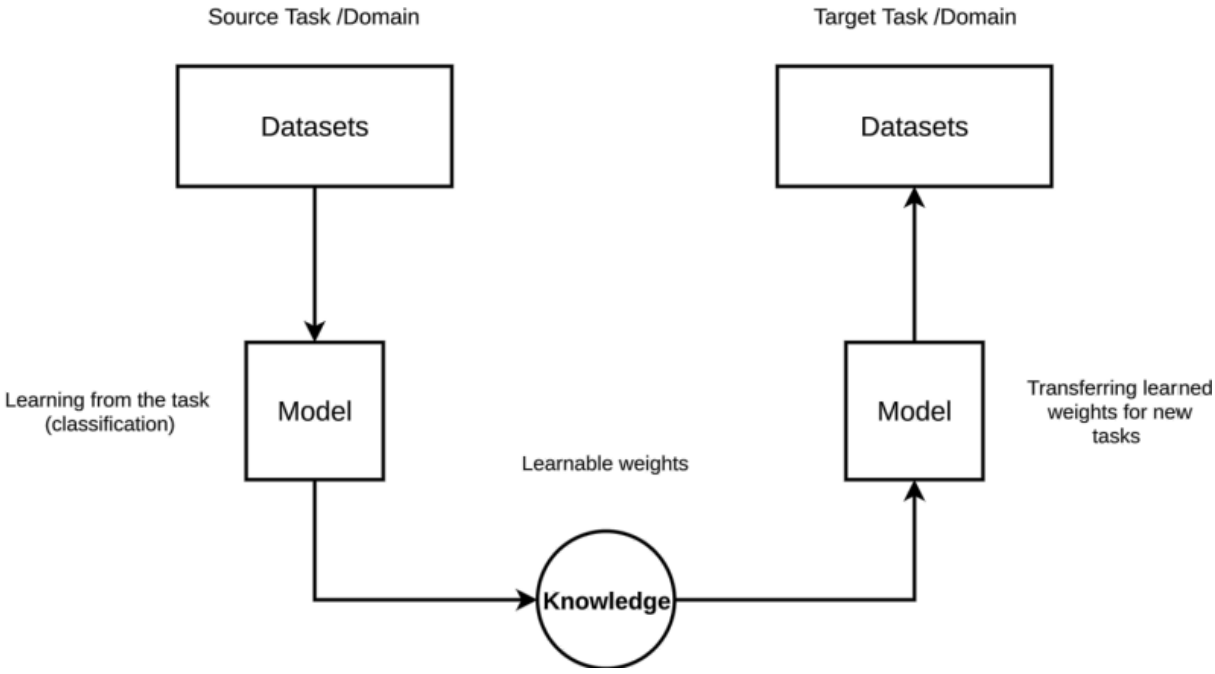


Figure 4.14: Transfer Learning [1]

4.2.2.3 Training fully connected layers

Keeping convolutional layers in frozen state(thus not altering their weights, fully connected layers are trained using the kaggle dataset (In the kaggle dataset, class tiger contains 1200 images, class elephant contains 1050 images, class deer contains 1100 images, class gaur contains 1000 image each) and 4 new output classes ,which include tiger,guar,elephant and deer are added.

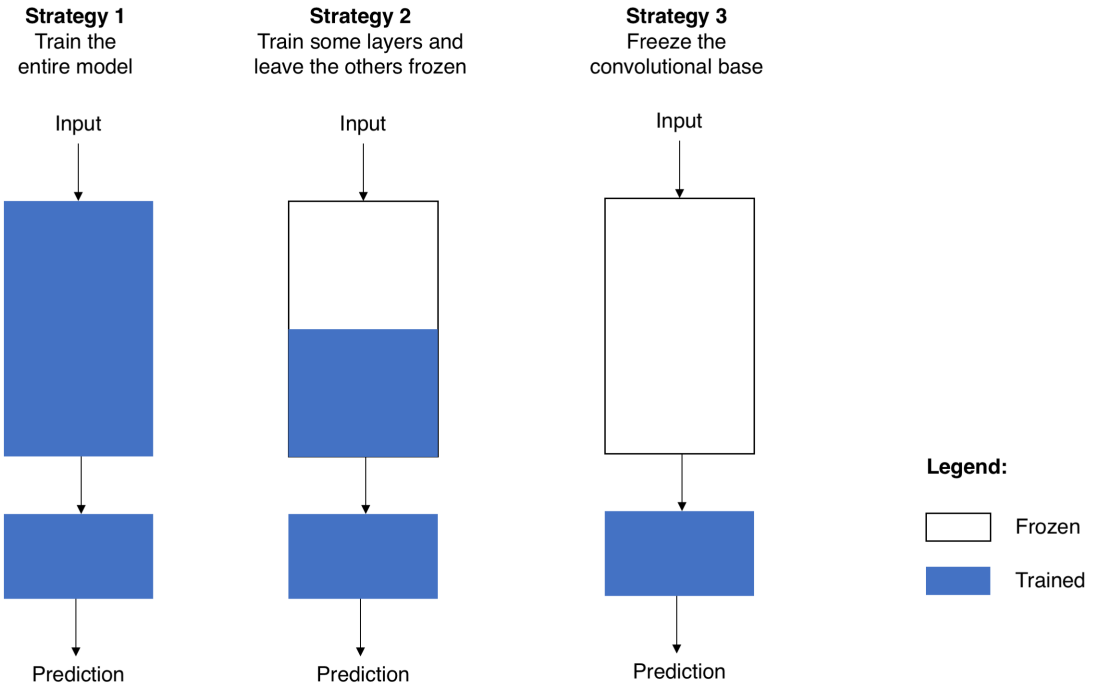


Figure 4.15: Training model using transfer learning approach [1]

4.2.2.4 Fine Tuning Convolutional layers

Since a general to specific layer transition occurs from left to right in the case of convolutional layers, weights of a few more convolutional layers should be altered in order to get accurate output. Decreasing the number of convolutional layers to be kept frozen each time and training the remaining ones, fine tuning is performed until satisfactory predictions are achieved.Once the fine tuning was completed, the model is able to predict the class of animal in an accurate manner for clear images.

4.2.3 Warning system

Once the type of animal is identified, a warning signal will be sent to road side hardware unit which gives warning based on the type of animal. A program running in the on board computer system present there will provide red light for class 1 animals, yellow light for class

2 animals and a green light in no animal case. A working sample of the same is made using an aurdino and LED lights.

Figure 4.16: Warning System [6]

Chapter 5

Results and Discussion

5.1 CLASSIFICATION

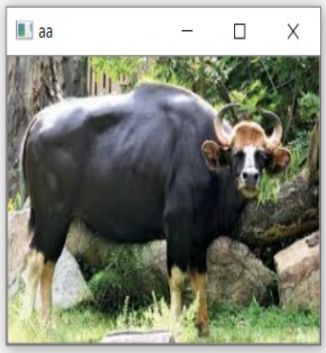
The model is trained using images from a kaggle dataset. In kaggle dataset, class tiger contains 1200 images, class elephant contains 1050 images, class deer contains 1100 images, class gaur contains 1000 image each. A 10 fold cross validation is done using a test set of 100 images in order to optimize the hyper parameters and the software model is completed. Each class contains 25 images for cross validation.

After the completion of software, when we input image of an animal is fed to the software, corresponding class name is obtained in the manner shown below. The CNN model showed 100 percent accuracy in this cases when the image is much clear.

In [*]:
```
img = cv2.imread('./data/test/2.jpg')
orig_img = img
img = cv2.resize(img, (299, 299))
img = img * (1. / 255)
img = img.reshape(1, 299, 299, 3)

pred = model.predict(img)
predicted_class_indices = np.argmax(pred, axis = 1)
labels = { 'deer': 0, 'elephant': 1, 'gaur': 2, 'tiger': 3 }
labels = dict((v, k) for k, v in labels.items())
predictions = [labels[k] for k in predicted_class_indices]
print(predictions)

cv2.imshow('aa', orig_img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```




```
['gaur']
```

Figure 5.1: Sample Output1

```
In [*]: img = cv2.imread('./data/test/1.jpg')
        orig_img = img
        img = cv2.resize(img, (299, 299))
        img = img * (1. / 255)
        img = img.reshape(1, 299, 299, 3)

        pred = model.predict(img)
        predicted_class_indices = np.argmax(pred, axis = 1)
        labels = { 'deer': 0, 'elephant': 1, 'gaur': 2, 'tiger': 3 }
        labels = dict((v, k) for k, v in labels.items())
        predictions = [labels[k] for k in predicted_class_indices]
        print(predictions)

        cv2.imshow('aa', orig_img)
        cv2.waitKey(0)
        cv2.destroyAllWindows()

['tiger']
```

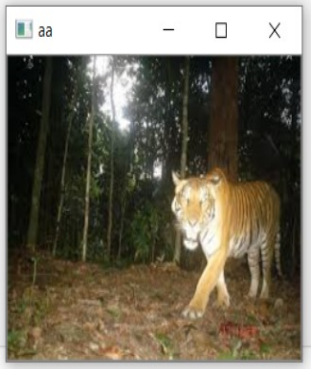


Figure 5.2: Sample Output2

Test Set

Since all images taken from kaggle dataset resulted in accurate prediction when used as test data, a more challenging dataset is created for testing, which contains camera trap images of animal taken from google. It consists of 9 images of class tiger, 8 images of class gaur, 7 images of class deer and 8 images of class elephant, which were either shot at dim light or were blur.

(a) Deer

(b) Elephant

(a) Gaur

(b) Tiger

Figure 5.4: Sample of Dataset [7]

Using this database, testing is done and the following results are obtained.

	Precision	Recall	f1-score	Support
0	1.00	0.43	0.60	7
1	0.57	1.00	0.73	8
2	0.88	0.88	0.88	8
3	1.00	0.78	0.88	9
Accuracy			0.78	32
macro average	0.86	0.77	0.77	32
weighted average	0.86	0.78	0.78	32

Table 5.1: Classification results

- class 0: deer
- class 1: elephant
- class 2: guar
- class 3: tiger

There are four types of observations, true positives (TP): correctly predicted positive values, true negative (TN): correctly predicted negative values, false positive (FP): actual class is no and predicted class is yes and false negative (FN): actual class is yes but predicted

class in no. Accuracy can be calculated based on these parameters. Precision, recall and f1-score shows the performance of the system. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Precision can be calculated as:

$$\text{Precision} = TP/TP+FP$$

Precision of class 0 (deer) is obtained as 1.00. That implies all the challenging images in the validation set is correctly identified. For class 1 (elephant) precision is 0.57, class 2 (guar) is 0.88 and class 3 (tiger) is 1. Recall is the ratio of correctly predicted positive observations to the all observations in actual class. Recall can be calculated as:

$$\text{Recall} = TP/TP+FN$$

Recall of class 0, class 1, class 2 and class 3 is 0.43, 1.00, 0.88 and 0.78 respectively. F1 Score is the weighted average of Precision and Recall. F1-score takes both false positives and false negatives into account. F1-score can be obtained as follow:

$$\text{F1 Score} = 2*(\text{Recall} * \text{Precision}) \ / \ (\text{Recall} + \text{Precision})$$

F1-score for class 0 is 0.60, class 1 is 0.73, class 2 is 0.88 and class 3 is 0.88. Accuracy is the given as sum of true positive and true positive by total number of observation. It can be calculated as:

$$\text{Accuracy} = TP+TN/TP+FP+FN+TN$$

Accuracy of this model is 0.78 for the given challenging validation set. Support is the number of samples of true response that lie in that class. Macro-average: the function to compute f1 for each label, and returns the average without considering the proportion for each label in the dataset. Weighted average: the function to compute f1 for each label, and returns the average considering the proportion for each label in the dataset.

(a) Elephant

(b) Bison

Figure 5.5: Images wrongly identified [7]

These are some of the following images which are wrongily classified. They are either shot at dim light or where distinguishing features of animals (such as tusk of an elephant) were absent. An overall accuracy of 78 percent is currently present even in the case of challenging ones. This can only be improved over time by incoporating more and more challenging data images taken from real cameratraps, since the number of such images availble in public domain are either clear or contains distinguishing features of animals.

5.2 WARNING SYSTEM

A Warning System model which recieves signals from backend CNN based classification software and displays warning signals was made using Arduino, ESP8266 Wi-Fi module and LED lights. ESP8266 is used to receive signals from CNN based classification software then this signal send to Arduino. The Arduino runs a program which glows red led for class 1 animals, yellow one for class 2 and green LED when no animal belonging to class 1 or 2 is present.

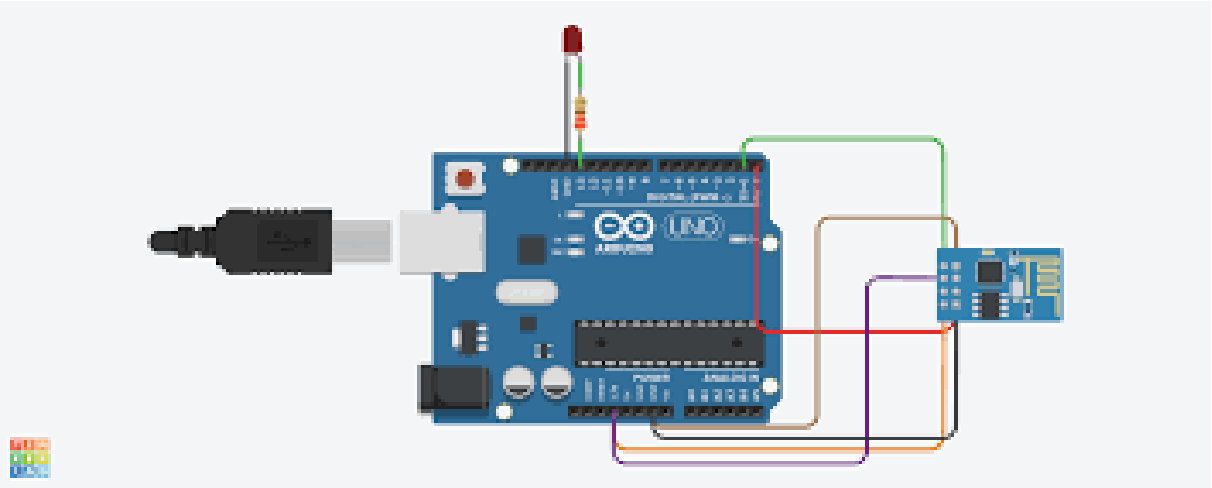


Figure 5.6: Representative image [6]

Chapter 6

Conclusion and Future Scope

The proposed system consists of a high definition infrared night vision camera trap for capturing animal movements, a CNN based classifier for classifying the type of animal and a road side unit hardware for giving proper warning. The model is giving accurate results for images which are clear and is giving a decent accuracy of 78 percentage in case of images which are blur. The ability of software to identify the animals from images which are not clear is due to absence of distinguishing features (such as tusk of an elephant or lines of tiger) due to dim light or being hided behind something. This can only be improved over time by incorporating more and more challenging data images taken from real cameratraps, since the number of such images available in public domain are either clear or contains distinguishing features of animals. By implementing the model near roads and villages near forest, dangerous animal human face off can be reduced by giving proper warning to the people. In collaboration with govt. agencies such as CDIT and forest department, the project can be implemented at forests of Kerala, where animal human interaction is serious cause of concern.

References

[1] Hung Nguyen, Sarah J. Maclagan, Tu Dinh Nguyen, Thin Nguyen, Paul Flemons, Kylie Andrews, Euan G. Ritchie and Dinh Phung, "Animal Recognition and Identification with Deep Convolutional Neural Networks for Automated Wildlife Monitoring", *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA).*

[2] F. Viani, P. Rocca, L. Lizzi, M. Rocca, G.Benedetti, and A. Massa, "WSN-based Early Alert System for Preventing Wildlife-Vehicle Collisions in Alps Regions", *Antennas and Propagations in Wireless Communications(APWC), 2011-APS IEEE Topical Conference.*

[3] Nami Susan Kurain, Poojasree s, S.Priyadharrshini, "Wildlife Vechile Collision Avoidance System", *SSRG International Journal of Electronics and communication Engineering(SSRG -IJECE)- Volume 5 Issue 3 - March 2018.*

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition " *SSRG International Journal of Electronics and communication Engineering(SSRG -IJECE)- Volume 5 Issue 3 - March 2018.*

[5] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger, "Densely Connected Convolutional Networks" *SSRG International Journal of Electronics and communication Engineering(SSRG -IJECE)- Volume 5 Issue 3 - March 2018.*

[6] Wikipedia *https://en.wikipedia.org/wiki/Convolutional*$_n$*eural*$_n$*etwork*