Animal Detection and Identification using Machine Learning

A major project phase-I report

Submitted To

Chhattisgarh Swami Vivekanand Technical University
Bhilai, India

For

The Award of The Degree

of

Bachelor of Technology

in

Computer Science & Engineering

By

Vipin Chandra Sao	Siddharth V. Pillai	Shubhanshu Tiwari
Roll No.-03302219120	Roll No.-303302219102	Roll No.-303302219099
En. No.-BH3803	En. No.-BH3785	En. No.-BH3782
Semester 7th (CSE)	Semester 7th (CSE)	Semester 7th (CSE)

Under the Guidance of

Mr. Yogesh Kumar Rathore

Asst. Professor CSE

Department of Computer Science & Engineering

S.S.I.P.M.T, Raipur

Department of Computer Science & Engineering
Shri Shankaracharya Institute of Professional Management & Technology Raipur (C.G.)

SESSION 2022-23

DECLARATION BY THE CANDIDATE

We the undersigned solemnly declare that the Major Project report entitled "***Automated animal detection and identification using Machine Learning***", is based on our own work carried out during the course of our study under the supervision of ***Mr. Yogesh Kumar Rathore.***

We assert that the statements made, and conclusions drawn are an outcome of the project work. We further declare that to the best of our knowledge and belief that the report does not contain any part of any work which has been submitted for the award of any other degree/diploma/certificate in this University/deemed University of India or any other country.

(Signature of Candidate 1)	**(Signature of Candidate 2)**	**(Signature of Candidate 3)**
Name: Vipin Chandra Sao	**Name: Siddharth V. Pillai**	**Name: Shubhanshu Tiwari**
Roll No.:303302219120	Roll No.:303302219102	Roll No.:303302219099
En. No.: BH3803	En. No.: BH3785	En. No.: BH3782
Semester:7th	Semester:7th	Semester: 7th

CERTIFICATE BY THE SUPERVISOR

This is to certify that the Major project report entitled **"*Automated animal detection and identification using Machine Learning*"** is a record of project work carried out under my guidance and supervision for the fulfillment of the award of degree of Bachelor of Engineering in the faculty of Computer Science & Engineering of Chhattisgarh Swami Vivekananda Technical University, Bhilai (C.G.) India.

To the best of my knowledge and belief the report

i) Embodies the work of the candidate himself
ii) Has duly been completed
iii) Fulfills the partial requirement of the ordinance relating to the B.E. degree of the University
iv) Is up to the desired standard both in respect of contents and language for being referred to the examiners.

(Signature of the Supervisor)
Mr. Yogesh Kumar Rathore
Assistant Professor, Dept of C.S.E.
S.S.I.P.M.T, Raipur (C.G.)

Forwarded to

Chhattisgarh Swami Vivekanand Technical University

Bhilai

(Signature of HOD)
Dr. J.P.Patra
Dept. of Computer Science & Engineering
S.S.I.P.M.T, Raipur, C.G

(Signature of the Principal)
Dr. Alok Kumar Jain
S.S.I.P.M.T, Raipur, C.G

CERTIFICATE BY THE EXAMINERS

The Project report entitled "**Automated animal detection and identification using Machine Learning**" has been examined by the undersigned as a part of the examination and is hereby recommended for the award of the degree of Bachelor of Technology of Chhattisgarh Swami Vivekanand Technical University, Bhilai.

Internal Examiner
Date:

External Examiner
Date:

ACKNOWLEDGEMENT

Working for this project has been a great experience for us. There were moments of anxiety, when we could not solve a problem for the several days. But we have enjoyed every bit of process and are thankful to all people associated with us during this period we convey our sincere thanks to our project guide **Mr. Yogesh Kumar Rathore** for providing me all sorts of facilities. His support and guidance helped us to carry out the project. We owe a great dept. of his gratitude for his constant advice, support, cooperation & encouragement throughout the project we would also like to express our deep gratitude to respected **Dr. J P Patra** (Head of Department) for his ever helping and support. We also pay special thanks for his helpful solution and comments enriched by his experience, which improved our ideas for betterment of the project. We would also like to express our deep gratitude to respected **Dr. Alok Kumar Jain** (Principal) and college management for providing an educational ambience. It will be our pleasure to acknowledge, utmost cooperation and valuable suggestions from time to time given by our staff members of our department, to whom we owe our entire computer knowledge and also we would like to thank all those persons who have directly or indirectly helped us by providing books and computer peripherals and other necessary amenities which helped us in the development of this project which would otherwise have not been possible.

(Signature of Candidate 1)
Name: Vipin Chandra Sao
Roll No.:303302219120
En. No.: BH3803
Semester:7th

(Signature of Candidate 2)
Name: Siddharth V. Pillai
Roll No.:303302219102
En. No.: BH3785
Semester:7th

(Signature of Candidate 3)
Name: Shubhanshu Tiwari
Roll No.:303302219099
En. No.: BH3782
Semester: 7th

ACKNOWLEDGEMENT –AICTE IDEA Lab

We have taken efforts in this project. However, it would not have been possible without the kind support and help of AICTE-IDEA Lab at SSIPMT, Raipur. We would like to extend our sincere thanks to all the gurus, mentors and support staff of Idea lab.

ABSTRACT

Animal detection is an application of machine learning through which we are able to detect and identify animal in an image or a video. It can provide user to tackle various animal-based problems like cattle counting, monitoring animals, preventing unwanted animal invasion like monkeys etc. in an area. In this paper, machine learning is used to create an animal detection and identification ML model that can be used in a diverse area. To train the ML model we first label the images from our data set. After labelling we provide the labelled data to a ML model for training, here YOLOv5 is used. After 50 epochs the model was able to achieve max F1 value of 0.98 with confidence of 0.89.

Keywords :- Machine Learning, Residual Neural Network, Artificial Neural Network.

TABLE OF CONTENTS

LIST OF FIGURES

+

CHAPTER-1

INTRODUCTION

1.1 INTRODUCTION

Observing wild animals in their natural environments is a central task in ecology. The fast growth of human population and the endless pursuit of economic development are making over-exploitation of natural resources, causing rapid, novel and substantial changes to Earth's ecosystems. An increasing area of land surface has been transformed by human action, altering wildlife population, habitat and behaviour [1]. More seriouslymany wild species on Earth have been driven to extinction, and many species are introduced into new areas where they can disrupt both natural and human systems. Monitoring wild animals, therefore, is essential as it provides researchers evidences to inform conservation and management decisions to maintain diverse, balanced and sustainable ecosystems in the face of those changes.

Various modern technologies have been developed for wild animal monitoring, including radio tracking, wireless sensor network tracking, satellite and global positioning system (GPS) tracking, and monitoring by motion sensitive camera traps. Motion-triggered remote cameras or "camera traps" are an increasingly popular tool for wildlife monitoring, due to their novel features equipped, wider commercial availability, and the ease of deployment and operation [2]. For instance, a typical covert camera model is capable of not only capturing high definition images in both day and night, but also collecting information of time, temperature and moon phase integrated in image data. In addition, generous and flexible camera settings allow tracking animals secretly and continuously. Once being fully charged, a camera can snap thousands of consecutive images, providing a large volume of data. These specifications make camera traps a powerful tool for ecologists as they can document every aspect of wildlife.

Visual data, if can be captured, is a rich source of information that provide scientists evidences to answer ecology-related scientific questions such as: what are the spatial distributions of rare animals, which species are being threatened and need protection such as bandicoot, which cohort of pest species, such as red fox and rabbit, need to be controlled; these are examples of key questions to understand wild animals' populations, ecological relationships and population dynamics [4]. To this end, a recently widely-used approach by ecologists is to set up several camera traps in the wild to collect image data of wild animals in their natural habitats.

In earlier research, People have worked on Identification of animal has aided humanoid being to look atand to require care their animals easily. Their work shows that support vector machines (SVM's) can simplify well on problematic image classification problems where the important features are high dimensional histograms. Their work also marks an experimental method, and

its clarifies having progressively better results gotten over alterations to the SVM architecture and that they have explained that it's likely to extend the classification obtained on image histograms to higher levels with error rates as low as 11% for the classification of nearly 14 Corel groups and 16% for a more general set of substances. Higher level spatial features, histogram features are used. Histograms are wont to bifurcate other kinds of data than images.

Another research have proposed object classification system and claimed it's capable of accurately detecting, localizing, and recovering the configuration of textured animals in real images. They have built a distortion model of shape from a collective videos of animals and an appearance model of texture from a labeled group of animal images, and syndicate these 2 models spontaneously. They developed a simple texture descriptor. They tested their models on a few datasets; images taken by professional photographers from the Corel collection, and images from the net taken from by Google. Instead of employing a histogram of textures, they represented texture with a patch of pixels. They demonstrated that their work is good enough for detecting animals. Broadly speaking, they also discussed on unsupervised algorithm for learning models using both video and images. These learned models appear promising for classification tasks beyond discovery, like localization, recovery, and counting.
To get a better understanding firstly we need to know about Neural Networks.

1.2 Neural Network-

A neural network consists of a number of nodes, called neurons, that performs computations based on the input and the functions used and generates outputs. These outputs are further given as input to other node which further perform computation. Each node is a representation of weight computation, with positive and negative weights denoting. Distinct responses to inputs that produces outputs. While a single node cannot make decisions, a neural network's decentralized structure breaks down complexity into components that can learn behaviour through emergent interaction. This non-linear method offers a more adaptable and responsive way to represent non-linear systems as a result [5].
Some basic components of a neuron are -

1. Inputs – The collection of values, known as inputs, are those for which an output must value must be predicted. They can be thought of as properties of a dataset.
2. Weights – Weights are the actual values with each input or attribute, and they indicate hoe important each one is for predicting or forecasting the outcome. It is multiplied with input values to provide priority to output generated by a layer.

3. Bias -Bias is an additional value to moves the activation function in left or right direction. It is added to make sure the neuron does not become dead.
4. Summation Function – Its purpose is to combine the weights along with input and bias to produce output.

i. $Y = WX + B$

ii. Where Y is the output, W is the weight, X is the input value and B is the bias. The above
iii. equation can also be compared with equation of a straight line.

5. Activation function- An activation function is an integral part of neural network which decides whether the neuron should be active or nor. It generates non linearity in the model.

i. A neural network comprises of vertical arrangements of nodes called layer and each layer's output act as input of the next layer, refer to figure 1. There are three types of layers in a neural network-

6. Input layer – Input data is provided to this layer and after the data is passed onto the rest of network.
7. Hidden Layer – For a neural network there are either one or more than hidden layers possible. Hidden layers are responsible to provide neural network with their exceptional performance and intricacy. They carry out several tasks concurrently, including data transformation and automatic feature detection.
8. Output Layer – This layer provides the result of the problem.

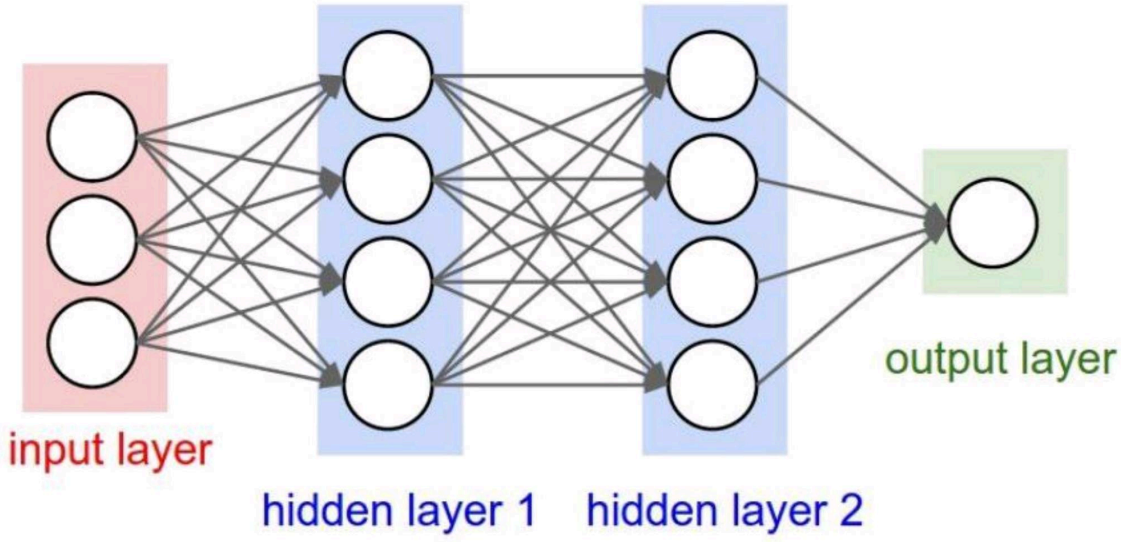

Figure 1.1 – Layers in Neural Network

In above fig 1.2 the different layers of neural network is showing that they are divided into three different layers input layer, hidden layer and output layer.

1.3 Models Used

We have used 2 models for Machine learning (1 for CNN and 1 for RCNN):

CNN (Conventional Neural Network):-

Artificial neural networks are used to classify words, audio, and images among other things. Different forms of neural networks are employed for various tasks. For example, to predict the order of words, recurrent neural networks—more specifically, an LSTM—are used. Similarly, to classify images, convolution neural networks are employed. We're going to create the fundamental building element for CNN in this blog [6].
Let's first review the basic concepts of neural networks before delving into the Convolution Neural Network. There are three different sorts of layers in a typical neural network:

1.**Input Layers**: This is the layer where we supply our model with input. The entire number of characteristics in our data is equal to the number of neurons in this layer (number of pixels in the case of an image).

2.**Hidden Layer**: The hidden layer is then fed the input from the input layer. Depending on our model and the volume of the data, there may be numerous hidden levels. The number of neurons in each hidden layer might vary, but they are typically more than the number of features. Each layer's output is calculated by multiplying the output of the layer below it by its learnable weights, adding learnable biases, and then computing the activation function, which makes the network nonlinear.

3.**Output Layer**: After being input into a logistic function like sigmoid or softmax, the output from the hidden layer is transformed into the probability score for each class.
The model is then given the data, and each layer's output is then obtained. We next calculate the error using an error function, such as cross-entropy, square loss error, etc. This stage is known as feedforward. After that, we calculate the derivatives and backpropagate into the model. Backpropagation is the process that is utilised to reduce loss in general.

RCNN(Region based CNN):-

Due to the fully linked layer of a Convolution Neural Network's (CNN) inability to handle multiple items and frequency of occurrence. In order to choose a region, we could, for example, perform a sliding window brute force search. Then, we could apply the CNN model to that region. However, this approach has the drawback that the same item might be represented in images of various sizes and aspect ratios [7]. We have a lot of region proposals while taking these considerations into account, and if we applied deep learning (CNN) to all of those areas, that would be highly computationally expensive.

To address the issue of object detection, Ross Girshick et al. (2013) suggested an architecture dubbed R-CNN (Region-based CNN). The selective search technique, which generates about 2000 region recommendations, is used in this R-CNN design. After then, the CNN architecture, which computes CNN features, receives these 2000 region proposals. An SVM model is then fed these features in order to classify the object that is present in the region suggestion. Perform a bounding box regressor as an additional step to more precisely pinpoint the things visible in the image. Regional suggestions: Simply defined, region recommendations are the smaller areas of the input image that may contain the things we are looking for. The R-CNN uses the selected search greedy algorithm to decrease the region recommendations.

Generation of Region Proposals Using Selected Search (Image Source: link) Selective Search: A greedy technique known as selective search combines smaller segmented regions to provide region proposals. This algorithm accepts an image as input and produces region suggestions on it as output. Because it restricts the number of proposals to roughly 2000 and gives these region proposals a high recall, this approach has an advantage over random proposal creation.

1. Create the initial input image sub-segmentation.
2. Recursively combine comparable bounding boxes into larger ones.
3. Create region suggestions for object detection using these larger boxes.

ResNet:-

ResNet, short for Residual Network is a classic neural network used as a backbone for many computer vision tasks. This model was the winner of ImageNet challenge in 2015.The fundamental breakthrough with ResNet was it allowed us to train extremely deep neural networks with 150+layers successfully. Prior to ResNet training very deep neural networks was difficult due to the problem of vanishing gradients [8].

Skip Connection – The Strength OfResNet

ResNet first introduced the connect of skip connection. The diagram below illustrates skip connection.The figure on the left is stacking convolution layers together one after the other. On the right we till stack convolution layers as before but we now also add the original input to the output of the convolution block. This is called skip connection.

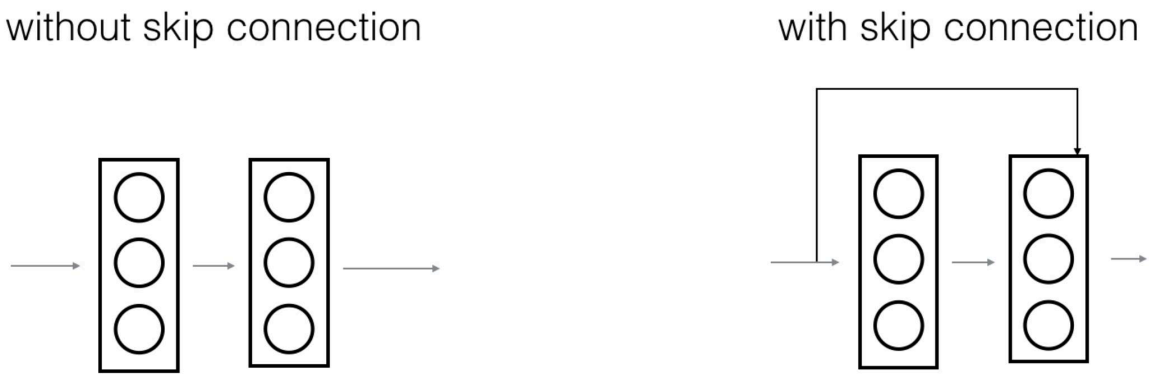

Fig 1.2 RESNet50 model

In above fig 1.2 the different connections of RESNet50 model is shown and how they perform

Architecture Of ResNet50: -

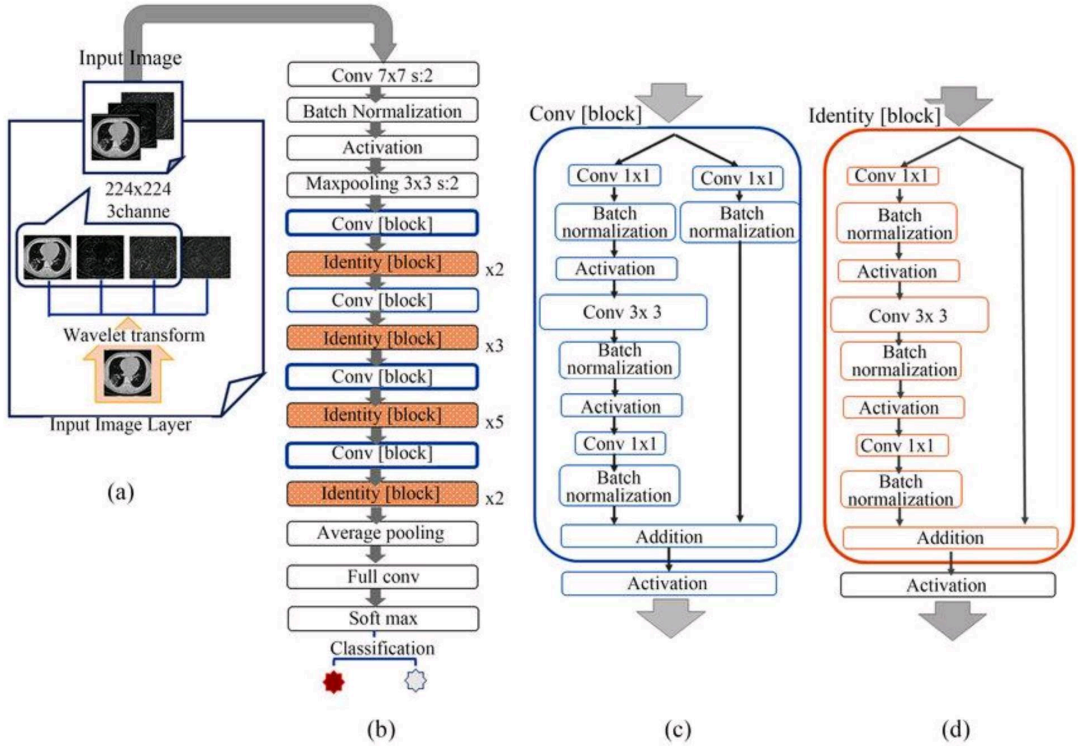

Fig 1.3 Architecture of RESNet50

In above fig 1.3 the architecture is shown that how many different layers are working in the RESNet50 and shown different blocks also they are conv and identity block.

YOLO v5 Model Architecture:-

As YOLO v5 is a single-stage object detector, it has three important parts like any other single-stage object detector.

1.Model Backbone

2.Model Neck

3.Model Head

Model Backbone is mainly used to extract important features from the given input. In yolo v5 the CSP-Cross Stage partial Networks are used as a backbone to extract rich in informative features from an input image [9].

The primary purpose of Model Neck is to produce feature pyramids. Pyramids of features enable models to scale objects successfully in general. The ability to recognise the same thing in various sizes and scales is helpful.

Models that use feature pyramids perform well on unobserved data. Other models, such as FPN, BiFPN, PANet, etc., employ other feature pyramid methodologies.

The final detecting step is primarily carried out using the model Head. It used anchor boxes on the features and produced final output vectors that included bounding boxes, objectless scores, and class probabilities.

Activation Function

The choice of activation function is most crucial in any deep neural network. Recently lots of activation functions have been introduced like leaky ReLU, mish, swish, etc.

YOLO v5 the Leaky ReLU activation function is used in middle/hidden layers and the sigmoid activation function is used in the final detection layer [10].

Optimization Function

For optimization function in YOLO v5, we have two options: -

1. SGD:-

   One of the most widely used algorithms for optimization and by far the most popular method for optimising neural networks is gradient descent. At the same time, each cutting-edge Deep. Implementations of different gradient descent optimization techniques are available in the learning library (e.g., the documentation for lasagne2, caffe3, and keras4). However, because it is difficult to find useful descriptions of these algorithms' advantages and disadvantages, they are frequently employed as "black-box" optimizers.

   The goal of this essay is to give the reader practical intuitions about how various gradient descent optimization techniques behave so that she can apply them. We will first examine the various gradient descent iterations in Section 2. The difficulties encountered during training will next be briefly reviewed in Section 3. The most popular optimization algorithms will then be introduced in Section 4 by demonstrating their drive to overcome these problems and how this results in the development of their update rules. After that, in Section 5, we'll take a quick look at the architectures and techniques that can be used to optimise gradient descent in a distributed and parallel environment.

   By updating the parameters in the opposite direction as the gradient of the objective function J() w.r.t. the parameters, gradient descent is a method for minimising an objective function J() parameterized by a model's parameters R d. The size of the steps we take to reach a (local) minimum is determined by the learning rate. To put it another way, we proceed downward along the slope of the surface produced by the objective function until we come to a valley.

2. Adam:-

   Adam is a method that uses adaptive estimations of lower-order moments to optimise stochastic objective functions using first-order gradients. The method is simple to use, computationally effective, requires little memory, is invariant to diagonal rescaling of the gradients, and works well for issues with a lot of parameters or data. The approach is also suitable for non-stationary goals and issues with extremely noisy and/or sparse gradients. The hyper-parameters may usually be tuned to a reasonable degree and have intuitive interpretations. Adam was inspired by some connections to related algorithms, which are explained. We also examine the algorithm's theoretical convergence characteristics and offer a bound on the convergence rate that is comparable to the most notable outcomes obtained using the online convex optimization framework. Empirical findings show that Adam performs admirably in real-world applications and compares favourably to other stochastic optimization techniques. We conclude by talking about AdaMax, an Adam variant based on the infinity norm.

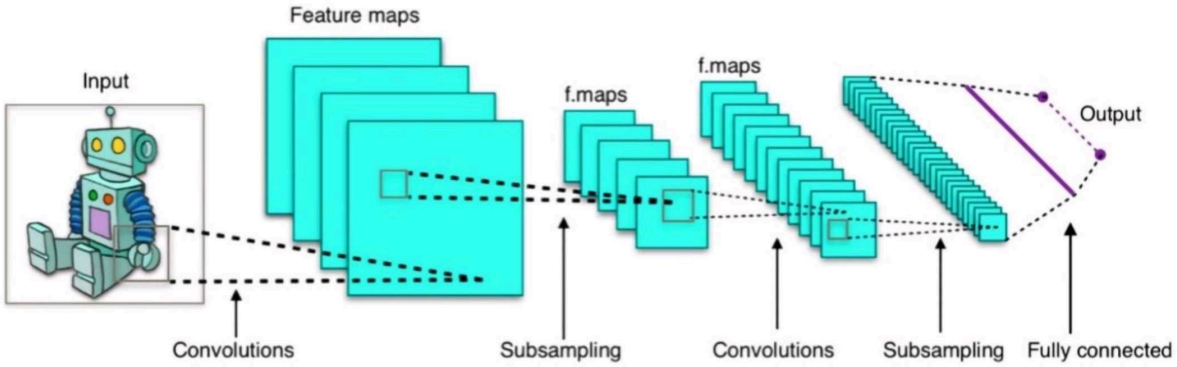

Fig 1.4 Optimization function(ADAM)

In The Above Fig 1.4 The Optimization Function Is Shown, It's Adam Which Is Optimization Function Of YOLOV5.

CHAPTER-2

LITERATURE REVIEW

2.1 LITERATURE SURVEY

[11] The paper talks about the conflict between humans and animals like monkey menace in urban areas and provides an effective solution by deployingintrusion detection applied with the help of wireless sensor based network based on UWB technology. The paper informs about SVM or softmax classifier to classify humans from animals. The intrusion of animals will be found by analyzing the UWB signals of the objects that entered the network , and extracting the features using convolutional neural network. The method proposed by the paper can allow us to detect human and animal invasion effectively with improved accuracy of 16% with respect to traditional manual extraction.

[12] The paper tries to tackle the problem of collision of animals and vehicles on roads which result in ecological imbalance. This paper tries to use animal detection via cameras that are installed on the roadside. The methodology used allows us to classify areas of image into two parts: animal and non-animal. In this paper, KNN and RF were used to process the features and it was found that KNN learning model was more reliable than Random Forest. For future, the paper suggested the use of deep learning to check for best possible methods in several conditions.

[13] The paper stated about the problem regarding the causalities and woundscaused due to road accidents due to collision with animals on mainly highways.The paper tries to implement a low cost animal detection using computer vision. The algorithm used for detection is based on HOG and cascade classifier. Training is done on more than 2000 images which contains positive and negative images. The system was ale to achieve max accuracy of 82.5% for detection. The limitation regarding the system is that it can detect and prevent collision till the range of 30-35km/h after which the system can detect but cant prevent the collision.

[14] The paper gives a sequence of algorithm components for the first stage of animal recognition pipeline called detection. The paper informs about 5-component detection pipeline ,each of which is a separate deep convolutional neural network, that can be used in computer vision based animal recognition system. The paper introduced a new dataset named WILD that can be used to challenge detection scenarios. The method used by the paper was able to poll off a localization mAPof 81.67% along with species and viewpoint annotation classification efficiency of 94.28% and 87.11% respectively along with an AoI accuracy of 72.75% across 6 defined animal species. For future the paper suggested improvements on WILD and performing comprehensive identification performance.

[15] The paper tries to tackle the problem of counting livestock like sheep on a regular basis which is otherwise done on major events like shearing or loading. The paper focuses on detecting sheep in padlock through UAVvideo and countsheep. The paper used 2 methods (i) a Region based Convolutional Neural Network for the detection of sheeps and (ii)an Expert System which took the advantage of uniform white color of sheep and blob analysis was done on the greyscale image. Both methods performed well but the method involving R-CNN require more work.

[16]The paper talks about the problem of human animal conflict which results in enormous amount of resource loss like loss of crops, livestock, property etc. To tackle the stated problem, the paper states a animal monitoring system that monitors a zone to prevent the entry of wild animals. The monitoring system consists of sensor that detects any intrusion and then camera is used to find the image of intruder and the image is classified using image classifier and suitable action is taken based on the class of intruder and a notifies owner using GSM.

[17]The paper proposes a wildlife monitoring and analysis system using deep convolutional neural network. The methodology consists of pre-processing, fine tuning DCNN features and classification through learning algorithms. The classification is done through transfer learning for which VGG-F pretrained model is used. The model used was pre-trained on ILSVRC 2012 dataset. For animal-background verification model training, classification uses machine learning algorithms such as SVMs, KNNs and ensemble algorithms. The end result showed that the accuracy of animal detection was 91% with F1-measure having max value 0.95. An accuracy of 91.4% with weighted KNN and DCNN features was obtained. The proposed system in the paper is able to detect wild animal during both day and night time.

[18] The paper deals with wild animal surveys and proposed a deep learning system to aid UAS (Unmanned Aircraft Systems) in detection of animals. Mainly two neural networks were used namely RetinaNet, a deep CNN that helped in achieving high precision and accurate animal surveys in UAS algorithms, and You Only Look Once v3 (YOLOv3), a single pass deep CNN that was able to develop a reliable model for automated counting of wild animals in UAS imagery. For future, the deep learning model can be made to support multiclass object system.

[19] The paper proposes the use of drones for cattle detection. The images that were captured by drone are analyzedby CNN to identify the object captured in the image. The CNN was

trained to detect cattle The architecture used for cattle detection is 64x64-18C7-MP4-96C5-MP2-4800L-2.The dataset was created from recordings made by a multirotor at different heights.The training dataset consist of samples of target, i.e. cattle, and the background. The average accuracy obtained in training was 97.1% and during testing an accuracy of 95.5% was obtained. For the future work, the CNN can be further trained to count the animals that have been grouped in the cattle.

[20] The paper suggests a machine learning pipeline for underwater animal detection system that allows us to monitor and understand the marine ecosystem. The dataset used was created using satellite images near LoVe observatory as it is rich in biodiversity. For the classification 2 versions of SVM and 4 CNN were used. The first SVM was linear and had C=1 and the second SVM was also linear but it had stochastic gradient descent training. For the CNN, two different structures were used. The first structure (CNN-1 and CNN-3) consisted of 2 blocks of convolution, activation and pooling layer while second (CNN-2 and CNN-4) contained 3 blocks. The first SVM had an accuracy of 0.5137 while the second had 0.4196 and CNN-1 had an accuracy of 0.6191, CNN-2 had an accuracy of 0.6563, CNN-3 had an accuracy of 0.6346 and CNN-4 had an accuracy of 0.6421. Using the various techniques an accuracy of 76.18% and AUC of 87.59% was achieved.

2.2 PROBLEM IDENTIFICATION

An increasing area of land surface has been transformed by human action, altering wildlife population, habitat and behavior. More seriously, many wild species on Earth have been driven to extinction, and many species are introduced into new areas where they can disrupt both natural and human systems. Monitoring wild animals, therefore, is essential as it provides researchers evidences to inform conservation and management decisions to maintain diverse, balanced and sustainable ecosystems in the face of those changes.

The aim of the proposed algorithm is to implement animal classification by using two deep learning neural network frameworks – CNN and Faster RCNN and highlight the importance of deep learning technology in classification problems. The proposed algorithm is consisting of five main steps.

- Image collection
- Data set splitting
- Train CNN and Faster RCNN algorithm
- Classify using CNN and Faster RCNN algorithm

CHAPTER-3

METHODOLOGY

3.1 Developer Requirements

Software Required

- IDE:- VS Code.
- Programming language:- Python(V3.8).
- Libraries:-Kivy, Tensorflow,pytorch, OpenCV.

Hardware Required

- i3 Processor Based Computer or higher
- Memory: 4 GB RAM.
- 5 GB free disk space.
- Internet connection.
- Camera.

3.2 User Requirements

Software Required&Hardware Required

- Windows 7 or 10
- 64-bit CPU (Intel / AMD)
- 4 GB RAM
- 5 GB free disk space

3.3 Type of SDLC Model

In this project, we are using Iterative SDLC Model.

In this Model, you can start with some of the software specifications and develop the first version of the software. After the first version if there is a need to change the software, then a new version of the software is created with a new iteration. Every release of the Iterative Model finishes in an exact and fixed period that is called iteration.

The Iterative Model allows the accessing earlier phases, in which the variations made respectively. The final output of the project renewed at the end of the Software Development Life Cycle (SDLC) process.

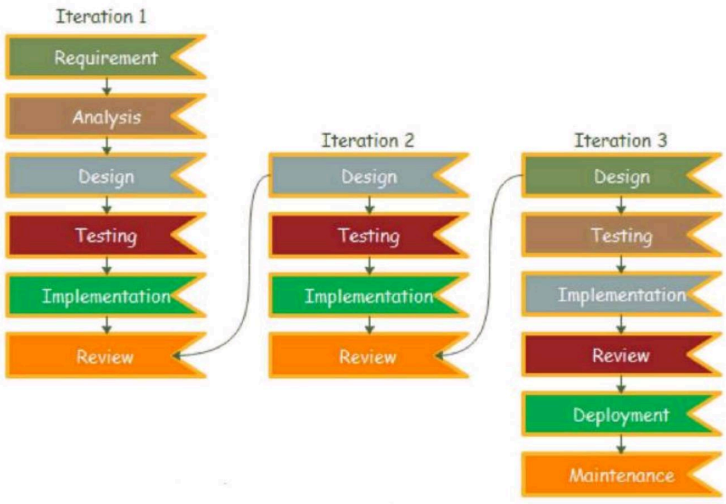

Fig 3.1 Iterative Model of SDLC

In above fig 3.1 the iterative model of SDLC(Software development life cycle) is shown that how they work in steps.

3.4Data Flow Diagram

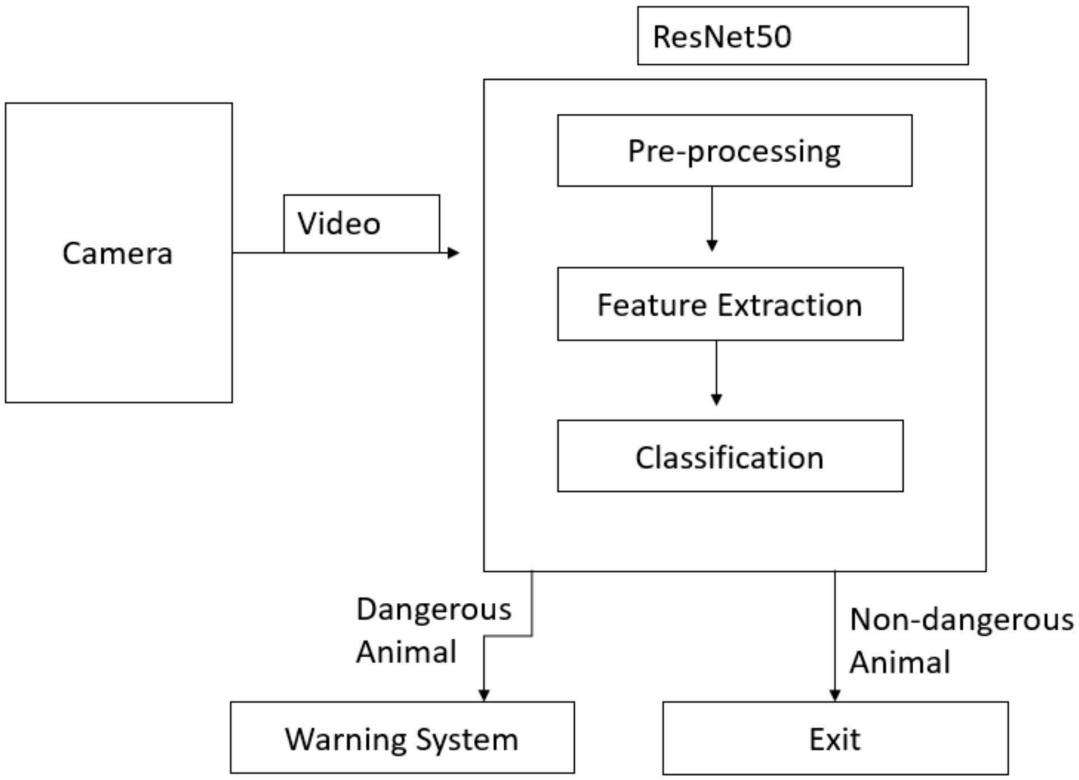

Fig 3.2Data flow Diagram

The above fig (3.2) represents data flow diagram which represents the flow of data at every state, and how the different steps is connected to each other.

3.5Workflow Diagram

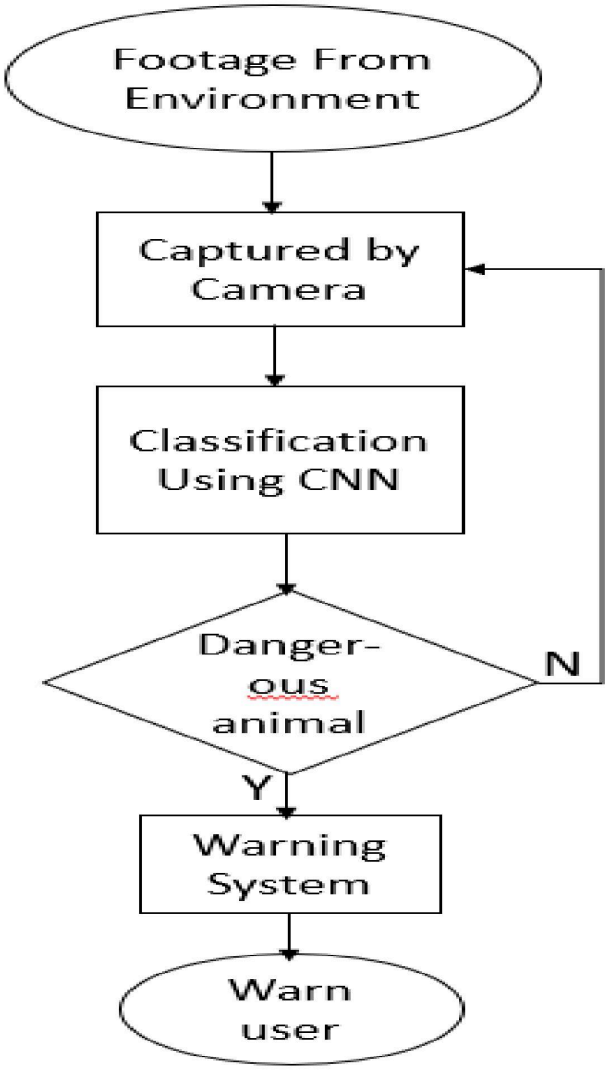

Fig 3.3Workflow Diagram

The above figure represents sequence diagram, the proposed system's sequence of data flow is represented.

3.6USE CASE Diagram

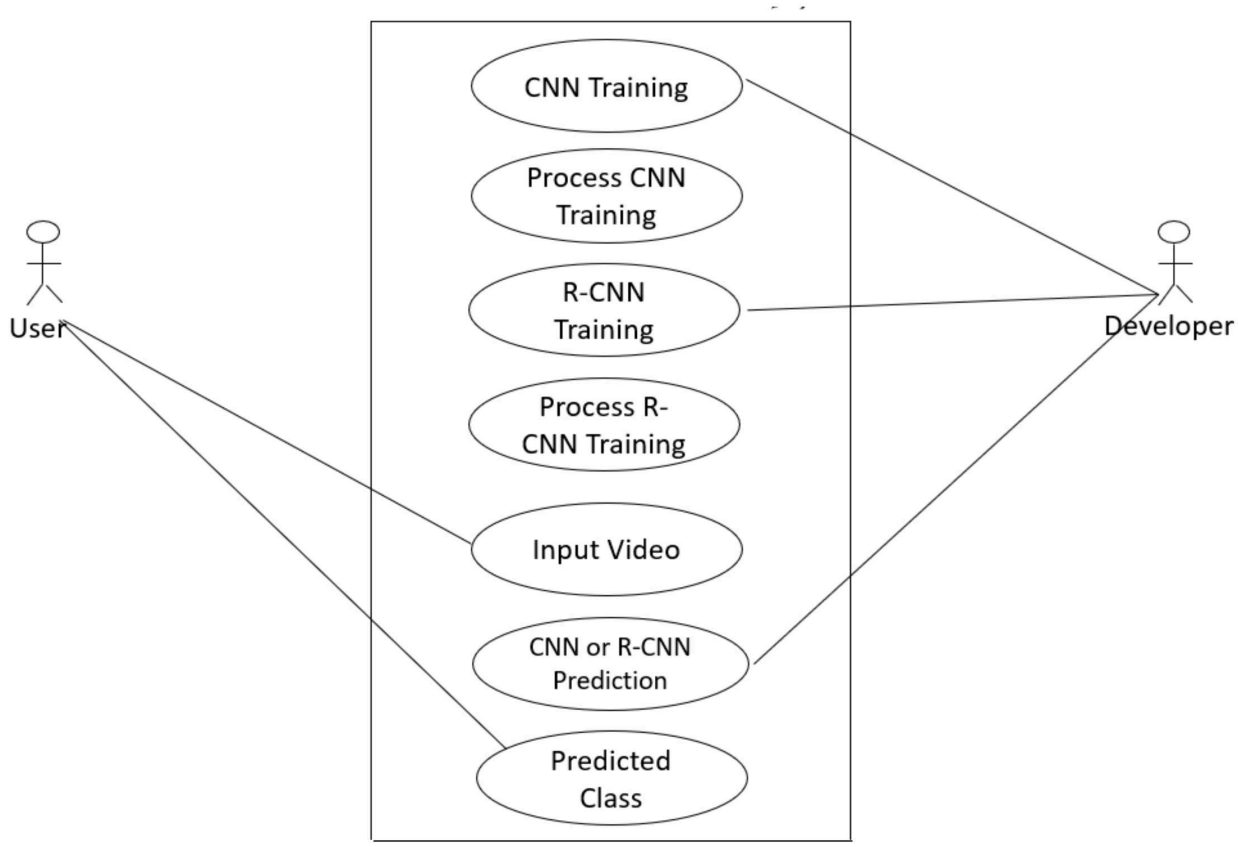

Fig 3.4Use Case Diagram

The above figure represents use case diagram, in which user upload dataset istrained and model is created for classification.

3.7Sequence Diagram

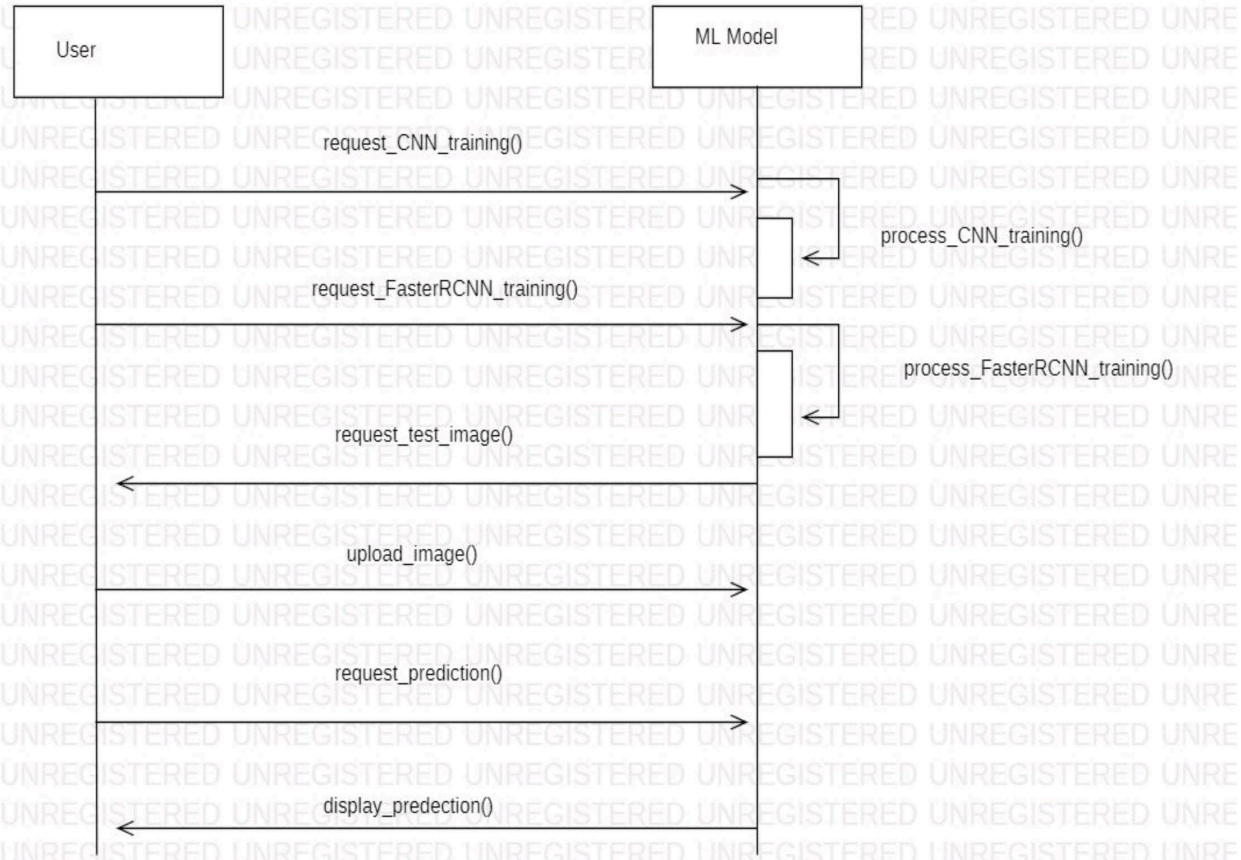

Fig 3.5 Sequence Diagram

The above figure represents sequence diagram, the proposed system's sequence of data flow is represented.

3.8Activity Diagram

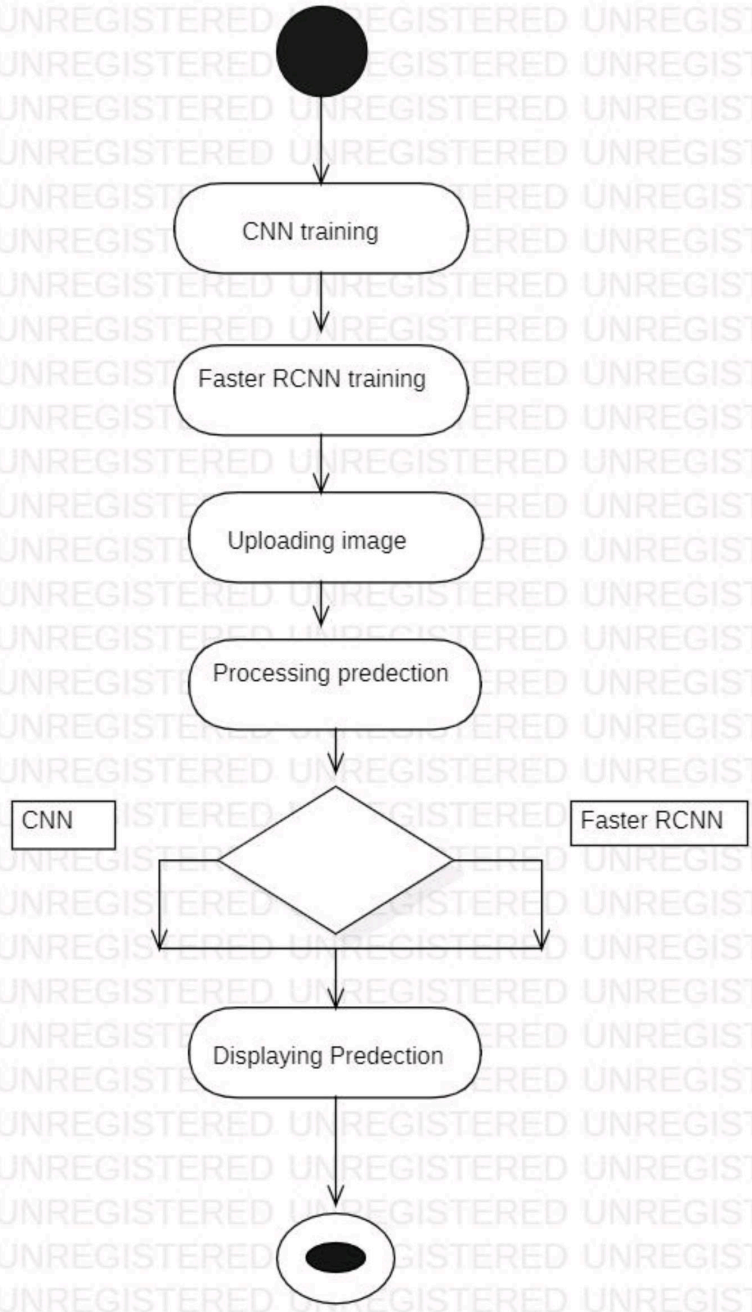

Fig 3.6 Activity Diagram

The above figure represents activity diagram of proposed system. The figure shows complete flow of activity from dataset loading and all sequence of module.

3.9Collaboration Diagram

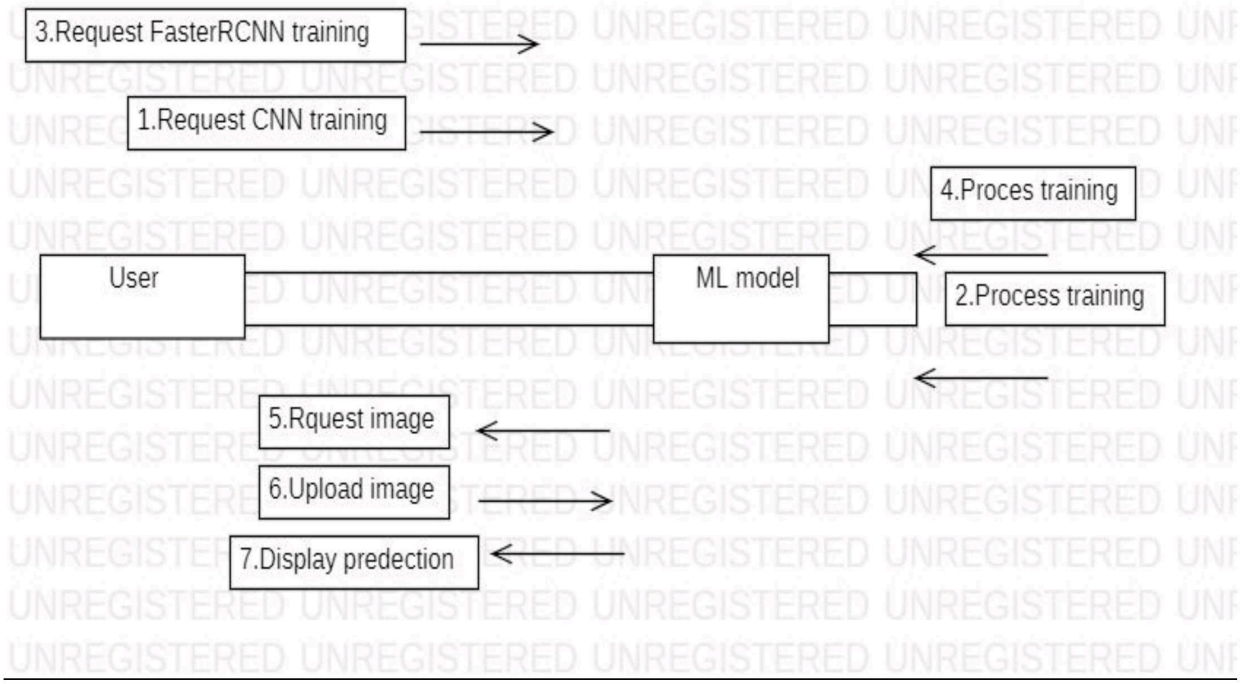

Fig 3.7 Collaboration Diagram

The above figure represents collaboration diagram of proposed system. The figure shows complete flow of collaboration from dataset loading and all sequence of model.

CHAPTER-4

RESULT

4.1 Training RESNet50:

4.1.1 steps for training

```
[ ] from tensorflow.keras.layers import Conv2D,Flatten,Dense,MaxPool2D,BatchNormalization,GlobalAveragePooling2D
    from tensorflow import keras
    from tensorflow.keras.applications.resnet50 import preprocess_input, decode_predictions
    from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_img
    from tensorflow.keras.applications.resnet50 import ResNet50
    from tensorflow.keras.preprocessing import image
    from tensorflow.keras.models import Sequential
    from tensorflow.keras.models import Model
    from google.colab import drive
    import matplotlib.pyplot as plt
    import numpy as np

[ ] drive.mount('/content/drive')

    Mounted at /content/drive

[ ] img_h, img_w = (224,224)
    batch_size = 32

    train_data_dir = r"/content/drive/MyDrive/processed_data/train"
    valid_data_dir = r"/content/drive/MyDrive/processed_data/val"
    test_data_dir = r"/content/drive/MyDrive/processed_data/test"
```

Fig 4.1 Here in this step we importing libraries and splitting the dataset.

```
[ ] train_datagen = ImageDataGenerator(preprocessing_function = preprocess_input,
                                       shear_range=0.2,
                                       zoom_range=0.2,
                                       horizontal_flip=True,
                                       validation_split=0.4)

    train_generator = train_datagen.flow_from_directory(
        train_data_dir,
        target_size = (img_h, img_w),
        batch_size = batch_size,
        class_mode='categorical',
        subset='training'
        )

    valid_generator = train_datagen.flow_from_directory(
        valid_data_dir,
        target_size = (img_h, img_w),
        batch_size = batch_size,
        class_mode='categorical',
        subset='validation'
        )

    Found 1985 images belonging to 90 classes.
    Found 360 images belonging to 90 classes.
```

```
test_generator = train_datagen.flow_from_directory(
    test_data_dir,
    target_size = (img_h, img_w),
    batch_size = 1,
    class_mode='categorical',
    subset='validation'
    )

Found 360 images belonging to 90 classes.
```

Fig 4.2 Creating the training , testing and validation datagen to find the classes in dataset.

```
base_model = ResNet50(include_top = False,weights='imagenet')

x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024,activation='relu')(x)

predictions = Dense(train_generator.num_classes,activation='softmax')(x)
model = Model(inputs = base_model.input, outputs = predictions)

for layer in base_model.layers:
    layer.trainable = False

model.compile(optimizer='adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])

model.fit(train_generator, epochs=10)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5
94773248/94765736 [==============================] - 1s 0us/step
94781440/94765736 [==============================] - 1s 0us/step
Epoch 1/10
63/63 [==============================] - 740s 12s/step - loss: 2.2313 - accuracy: 0.5028
Epoch 2/10
63/63 [==============================] - 361s 6s/step - loss: 0.4172 - accuracy: 0.8811
Epoch 3/10
63/63 [==============================] - 371s 6s/step - loss: 0.3574 - accuracy: 0.9214
Epoch 4/10
63/63 [==============================] - 371s 6s/step - loss: 0.2729 - accuracy: 0.9315
Epoch 5/10
63/63 [==============================] - 371s 6s/step - loss: 0.1471 - accuracy: 0.9602
Epoch 6/10
63/63 [==============================] - 373s 6s/step - loss: 0.0743 - accuracy: 0.9788
Epoch 7/10
63/63 [==============================] - 374s 6s/step - loss: 0.0416 - accuracy: 0.9914
Epoch 8/10
63/63 [==============================] - 374s 6s/step - loss: 0.0236 - accuracy: 0.9950
Epoch 9/10
63/63 [==============================] - 372s 6s/step - loss: 0.0213 - accuracy: 0.9945
Epoch 10/10
63/63 [==============================] - 374s 6s/step - loss: 0.0346 - accuracy: 0.9919
<keras.callbacks.History at 0x7f85049cbc50>
```

Fig 4.3 Training RESNet50 model with 10 epochs

```
[ ] model.save('/content/drive/MyDrive')
```

```
[ ] class_name = [ 'antelope','badger','bat','bear','bee','beetle','bison',
    'boar','butterfly','cat','caterpillar','chimpanzee','cockroach','cow','coyote','crab',
    'crow','deer','dog','dolphin','donkey','dragonfly','duck','eagle','elephant','flamingo','fly','fox','goat','goldfish','goose',
    'gorilla','grasshopper','hamster','hare','hedgehog','hippopotamus','hornbill','horse','hummingbird',
    'hyena','jellyfish','kangaroo','koala','ladybugs','leopard','lion','lizard','lobster',
    'mosquito','moth','mouse','octopus','okapi','orangutan','otter','owl','ox','oyster','panda','parrot','pelecaniformes','penguin',
    'pig','pigeon','porcupine','possum','raccoon','rat','reindeer','rhinoceros','sandpiper',
    'seahorse','seal','shark','sheep','snake','sparrow','squid','squirrel','starfish',
    'swan','tiger','turkey','turtle','whale','wolf','wombat','woodpecker','zebra']
```

```
[ ] test_loss , test_acc = model.evaluate(test_generator,verbose = 2)
    print('\n Accuracy ', test_acc)

    360/360 - 129s - loss: 0.7478 - accuracy: 0.8139 - 129s/epoch - 358ms/step

     Accuracy  0.8138889074325562
```

Fig 4.4and fig 4.5 Saving classes in the list and finding the accuracy of the model.

```
[ ] pred = model.predict(image)
    print(pred)
```

```
[ ] new_model = keras.models.load_model('/content/drive/MyDrive/ResNet50_animal.h5')
```

```
[ ] test_loss , test_acc = new_model.evaluate(test_generator,verbose =2)

    360/360 - 103s - loss: 0.7827 - accuracy: 0.8000 - 103s/epoch - 287ms/step
```

Fig 4.6Saving the model in local storage and finding loss and accuracy.

4.1.2 Testing on input image

Fig 4.7In this we taking images for input images for predicting the class of animal.

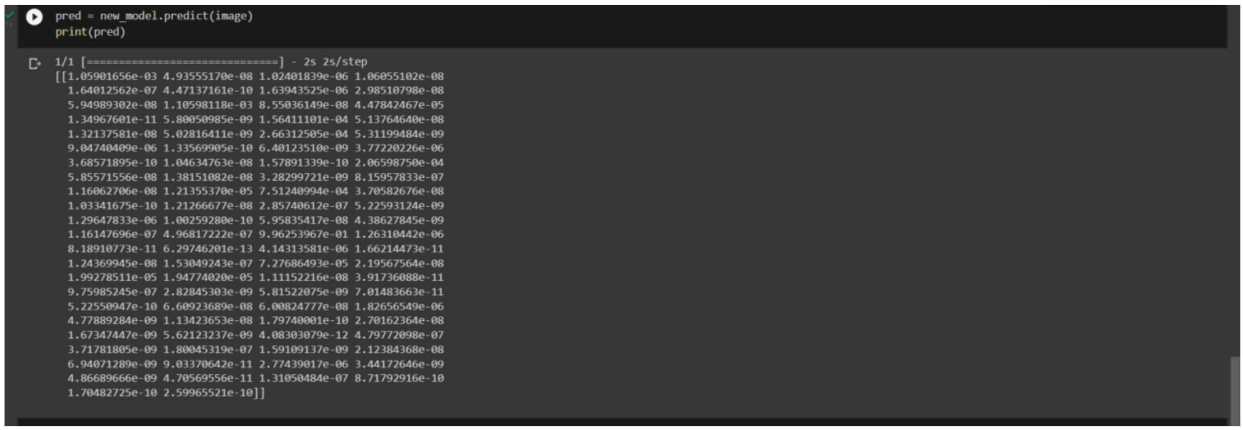

Fig 4.8 Result of the prediction with respect to all classes of listed animals.

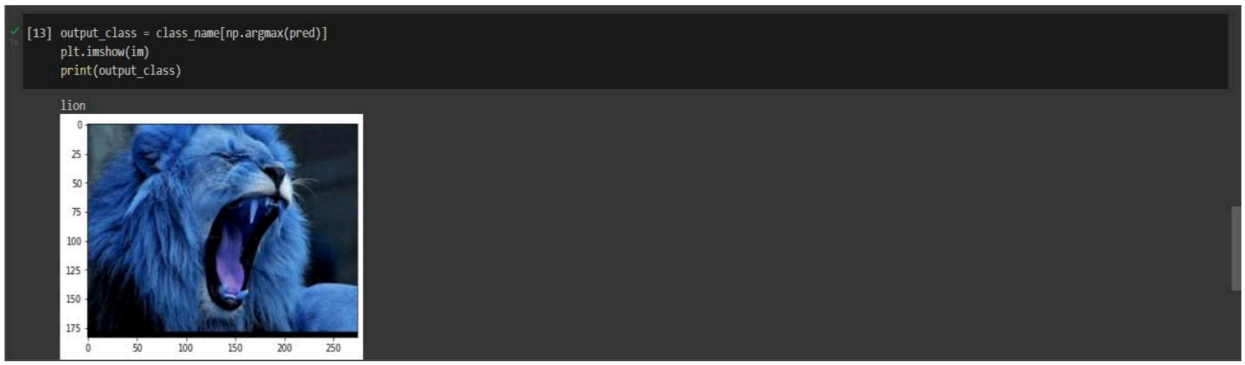

Fig 4.9Here is the output class of the image.

4.2 Training YOLOV5:

4.2.1 Step for training

1. We train the YOLOv5 model with the help of labelled data which has been labelled by the developer.
2. After training the model we get the weights

```
In [1]:
import torch
from matplotlib import pyplot as plt
import numpy as np
import cv2
import os
```

```
In [10]:
!cd yolov5 && python train.py --img 320 --batch 16 --epochs 50 --data dataset.yml --weights yolov5s.pt
```

```
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95:  17%|#6        | 1/6 [00:02<00:12,
2.47s/it]
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95:  33%|###3      | 2/6 [00:04<00:09,
2.46s/it]
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95:  50%|#####     | 3/6 [00:07<00:07,
2.55s/it]
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95:  67%|######6   | 4/6 [00:10<00:05,
2.62s/it]
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95:  83%|########3 | 5/6 [00:13<00:02,
2.84s/it]
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100%|##########| 6/6 [00:15<00:00,
2.55s/it]
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100%|##########| 6/6 [00:15<00:00,
2.59s/it]
                 all        178        233       0.98      0.983      0.993        0.9
            elephant        178         97      0.959      0.971      0.989      0.889
                lion        178         72      0.986      0.993      0.994      0.888
               tiger        178         64      0.995      0.984      0.995      0.923
Results saved to runs\train\exp5
```

Fig 4.10 After training we get an accuracy of 80%

3. We use the weights obtained by YOLOV5 model for transfer learning.

```
In [2]:
model = torch.hub.load('ultralytics/yolov5','custom',path='yolov5/runs/train/exp5/weights/best.pt',force_reload = True)
```

```
Downloading: "https://github.com/ultralytics/yolov5/archive/master.zip" to C:\Users\HP/.cache\torch\hub\master.zip
YOLOv5  2022-8-2 Python-3.9.5 torch-1.8.2+cpu CPU

Fusing layers...
Model summary: 213 layers, 7018216 parameters, 0 gradients, 15.8 GFLOPs
Adding AutoShape...
```

Fig 4.11 Saving the model

4.2 Step for predicting:

1. We input the image file name from the user.

```
In [*]: 1 img_path = input("Enter Image Name/Path: ")
        Enter Image Name/Path: 0a5f5db460.jpg
```

Fig 4.12 Taking Input image

2. We use the path of image for training the model and predict the output.

```
In [14]: 1 results = model(img)
         2 results.print()
image 1/1: 394x700 1 tiger
Speed: 10.2ms pre-process, 217.2ms inference, 15.8ms NMS per image at shape (1, 3, 384, 640)
```

Fig 4.13 Prediction of Input image

3. Display the prediction output.

```
In [17]: 1 %matplotlib inline
         2 plt.imshow(np.squeeze(results.render()))
         3 plt.show()
```

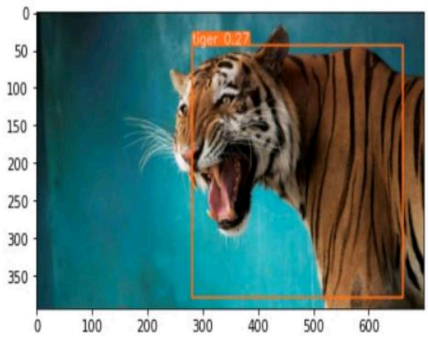

Fig 4.14 Output of prediction on image

4. We take video path/input from camera for detection.

```
In [18]: 1 input_video = input("Enter video name/path: ")
         Enter video name/path: Elephant.mp4
```

Fig 4.15 Taking video input and predicting

5. We use the model to predict class from the video and display the output

```
In [*]:
cap = cv2.VideoCapture(input_video)
while cap.isOpened():
    ret, frame = cap.read()

    # Make detections
    results = model(frame)

    cv2.imshow('YOLO', np.squeeze(results.render()))

    if cv2.waitKey(10) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

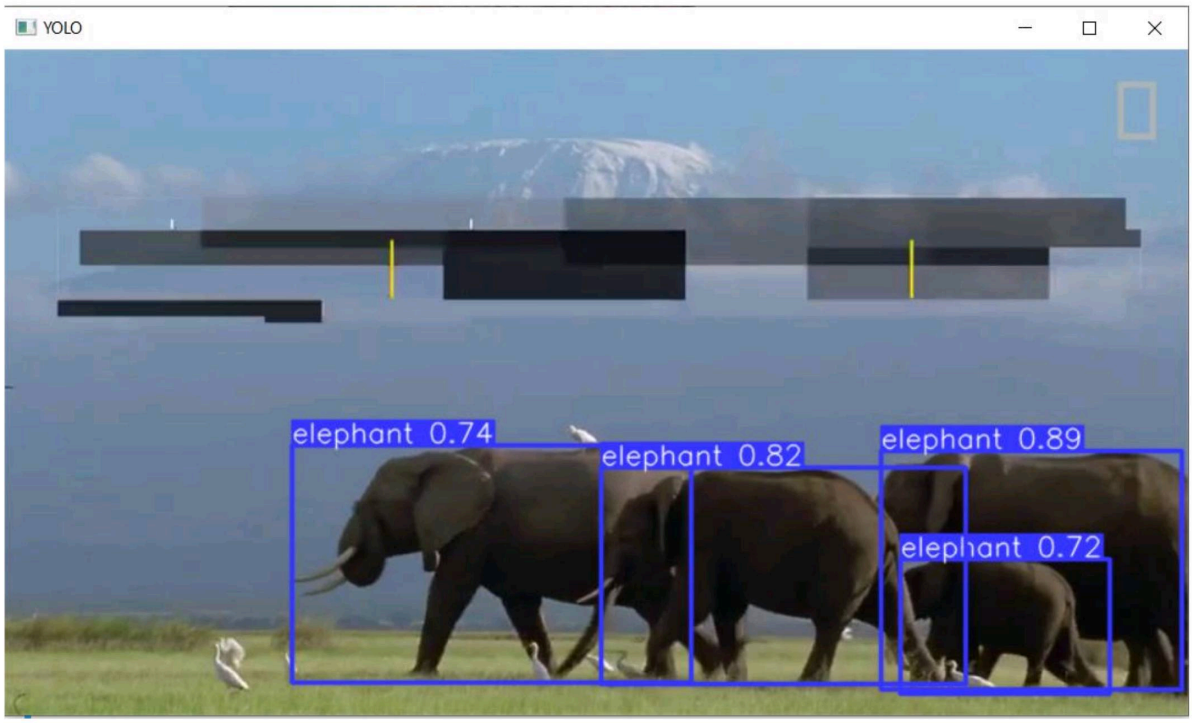

Fig 4.16 Output of prediction on video

RESNet epoch variation

EPOCH	Loss	Accuracy
6	0.8538	0.7924
10	0.7478	0.8139
12	0.7276	0.8022
14	0.7125	0.7824

YOLOv5 epoch variation

EPOCH	train/box_loss	train/object_loss	Train/cls_loss	Metrics/precision
10	0.055346	0.019502	0.014426	0.45647
25	0.039306	0.016005	0.0076101	0.92838
30	0.036894	0.015491	0.0035796	0.97397
50	0.012117	0.0042471	0.026788	0.98355

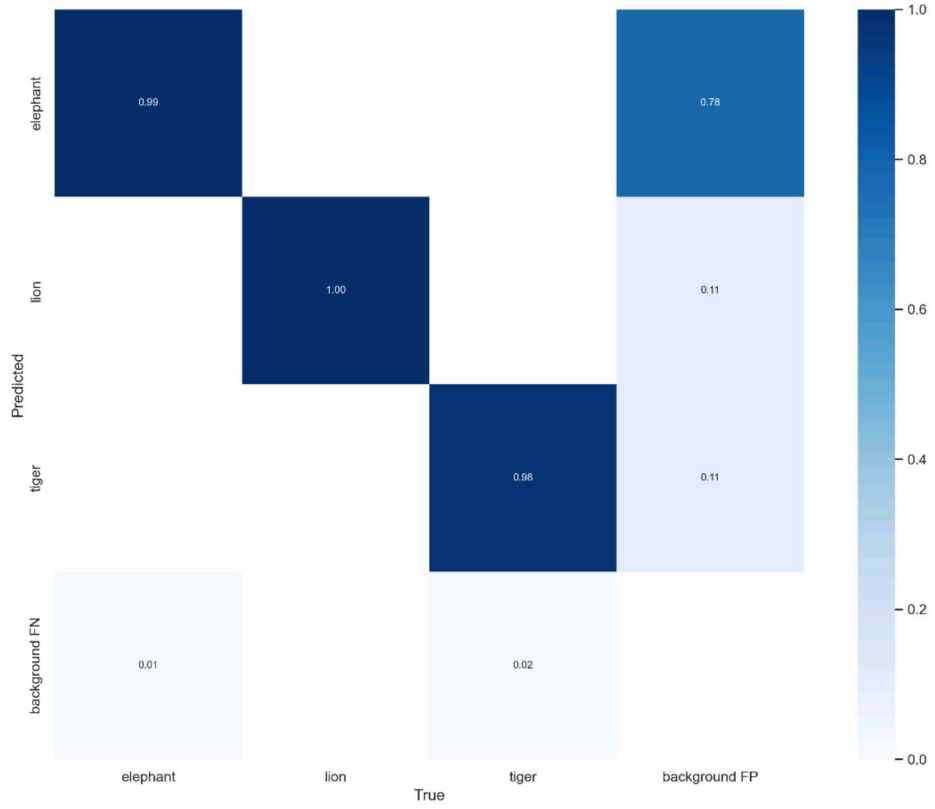

Fig 4.17 Confusion matrix of YOLO

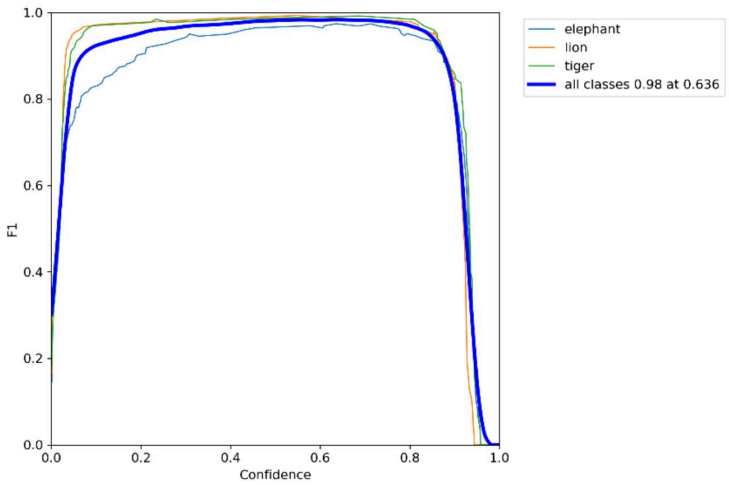

Fig 4.18 F1 curve of YOLO

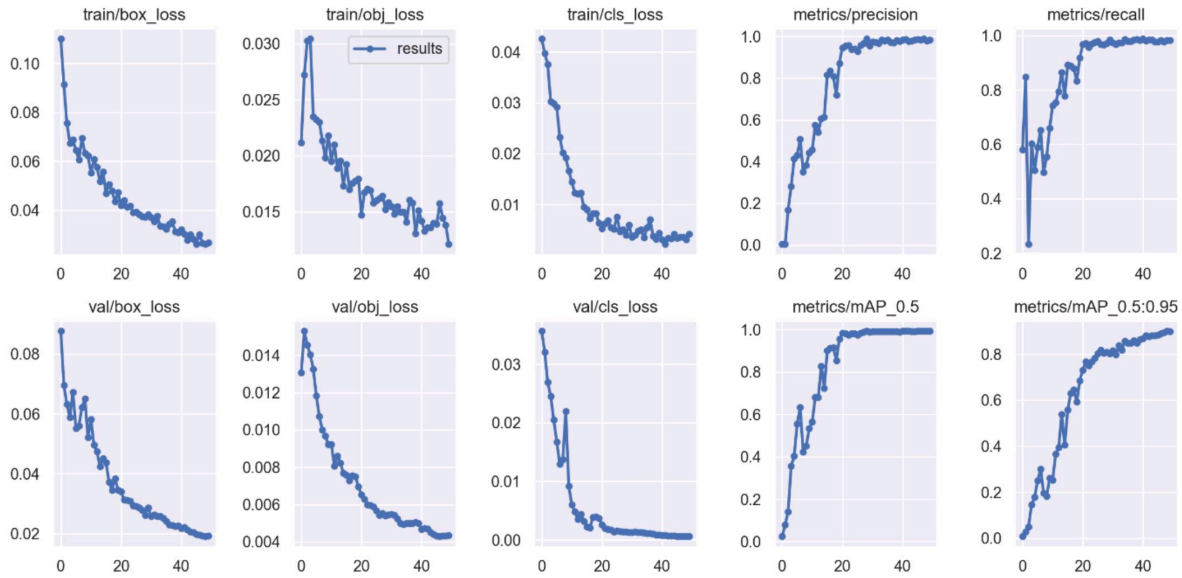

Fig 4.19 Result of YOLOv5

CHAPTER-5

CONCLUSION

Conclusion

Using the most recent technology, we proposed in this paper an automated classification system for animal photos and frameworks for deep learning neural networks to minimize human labour and expense. We assessed the neural networks' capacity to accurately categories photos. The outcomes emphasize the role of deep learning neural networks in image processing classification. Consequently, these network frameworks can be leveraged to decrease labour costs and effort and gather data on wildlife to protect and preserve the wildlife system.

FUTURE SCOPE

Future Scope

In the future, there will be very large scope, this project can be made based on wireless networks. Wireless sensor network, sensors and different types algorithms are used to collect the information of animals and these information is transmitted through network to the end user that initiates giving corrective actions..This work can be further extended by sending an alert in the form of a message when the animal is detected to the nearby forest office. Furthermore it can be used to reduce human wildlife conflict and also animal accidents.

REFRENCES

REFERENCES

[1] PARINITA BADRE, SIDDHANT BANDIWADEKAR, PRACHI CHANDANSHIVE, AAKANKSHA CHAUDHARI, SONALI JADHAV. Automatically Identifying Animals Using Deep Learning. In: 2018 International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE). Vol. 6, Issue 3, March (2018). ISSN(Online).

[2] GULLAL SINGH CHEEMA, SAKET ANAND. Automatic Detection and Recognition of Individuals in Patterned Species. IIIT – Delhi.

[3] Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. In November (2017).

[4] TIBOR TRNOVSZKY, PATRIK KAMENCAY, RICHARD ORJESEK, MIROSLAV BENCO, PETER SYKORA. Animal Recognition System Based on Convolution Neural Network. In September (2017).

[5] SAMER HIJAZI, RISHI KUMAR, CHRIS ROWEN. Using Convolution Neural Networks for Image Recognition. IP Group, Cadence.

[6] SHAOQING REN, KAIMING HE, ROSS GIRSHICK, JIAN SUN. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.

[7] HU W, HUANG Y, WEI L, ZHANG F, LI H (2015) Deep convolutional neural networks for hyperspectral image classification. Journal of Sensors 2015.

[8] REN, S., HE, K., GIRSHICK, R., SUN, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS. pp. 91–99 (2015).

[9] SIMONYAN, K., ZISSERMAN, A.: Very deep convolutional networks for large-scale image recognition.

[10] KRIZHEVSKY, A., I. SUTSKEVER and G. E. HINTON. ImageNet classification with deep convolutional neural networks. Annual Conference on Neural Information Processing Systems (NIPS). Harrah's Lake Tahoe: Curran Associates, 2012, pp

[11] Animal Intrusion Detection Based on Convolutional Neural Network byWenlingXue*, Ting Jiang*, Key Laboratory of Universal Wireless Communication Beijing University of Posts and Telecommunications Beijing, China,Ting Jiang* School of Electronic and Computer Engineering, ZhijiangWanliUniversity,Ningbo, China.

[12] A Proposal of an Animal Detection System Using Machine Learning by William H. S. Antônio, Matheus Da Silva, Rodrigo S. Miani& Jefferson R. Souza

[13] A Practical Animal Detection and Collision Avoidance System Using Computer Vision Technique by SACHIN UMESH SHARMA[1] AND DHARMESH J. SHAH[2]. 1 - Department of

Electronics and Communication, Gujarat Technological University, Ahmedabad 382424, India. 2 - Department of Electronics and Communication, Sankalchand Patel University, Visnagar 384315, India.

[14] An Animal Detection Pipeline for Identification by Jason ParhamCharles Stewart - Rensselaer Polytechnic Institute , Jonathan Crall - Kitware, Inc., Daniel Rubenstein - Princeton University, Jason Holmberg - Wild Me , Tanya Berger-Wolf University of Illinois-Chicago.

[15] Detecting and Counting Sheep with a Convolutional Neural Network by Farah Sarwar[1,2], Anthony Griffin[1,2] ,Priyadharsini Periasamy[2] , Kurt Portas[3] and Jim Law[3], [1] High Performance Computing Research Lab, [2] Electrical and Electronic Engineering Department School of Engineering, Computer and Mathematical Sciences Auckland University of Technology, New Zealand, [3] Palliser Ridge Limited, Pirinoa, New Zealand.

[16] IoT – based Wild Animal Intrusion Detection System by Prajna P, Soujanya B.S – Information Science Engineering, Vivekananda Collge of Engineering and Technology, Puttur.

[17] Wild Animal Detection Using Deep Convolutional Neural Network by Gyanendra K. Verma and Pragya Gupta

[18] Wild animal survey using UAS imagery and deep learning: modified Faster R-CNN for kiang detection in Tibetan Plateau byJinbang Peng, DongliangWang,Xiaohan Liao, QuanqinShao,ZhigangSun,HuanyinYue,Huping Ye.

[19] Detection of Cattle Using Drones and Convolutional Neural Networks by Alberto Rivas, Pablo Chamoso, Alfonso González-Briones and Juan Manuel Corchado.

[20] Video Image Enhancement and Machine Learning Pipeline for Underwater Animal Detection and Classification at Cabled Observatories by Vanesa Lopez-Vazquez, Jose Manuel Lopez-Guede, Simone Marini, EmanuelaFanelli, Espen Johnsen and Jacopo Aguzzi.

Website

- https://www.tensorflow.org/
- https://pytorch.org/

PUBLICATIONS

AUTOMATED ANIMAL IDENTIFICTION AND DETECTION USING MACHINE LEARING

Vipin Chandra Sao
Department of Computer Science & Engineering
Shri Shankaracharya Institute of Professional Management & Technology
Raipur, India

Shubhanshu Tiwari
Department Of Computer Science & Engineering
Shri Shankaracharya Institute Of Professional Management & Technology
Raipur, India

Siddharth V. Pillai
Department of Computer Science & Engineering
Shri Shankaracharya Institute of Professional Management & Technology
Raipur, India

Mr. Yogesh Kumar Rathore
Assistant Professor
Department of Computer Science & Engineering
Shri Shankaracharya Institute of Professional Management & Technology
Raipur, India

ABSTRACT – ANIMAL DETECTION IS AN APPLICATION OF MACHINE LEARNING THROUGH WHICH WE ARE ABLE TO DETECT AND IDENTIFY ANIMAL IN AN IMAGE OR A VIDEO. IT CAN PROVIDE USER TO TACKLE VARIOUS ANIMAL-BASED PROBLEMS LIKE CATTLE COUNTING, MONITORING ANIMALS, PREVENTING UNWANTED ANIMAL INVASION LIKE MONKEYS ETC. IN AN AREA. IN THIS PAPER, MACHINE LEARNING IS USED TO CREATE AN ANIMAL DETECTION AND IDENTIFICATION ML MODEL THAT CAN BE USED IN A DIVERSE AREA. TO TRAIN THE ML MODEL WE FIRST LABEL THE IMAGES FROM OUR DATA SET. AFTER LABELLING WE PROVIDE THE LABELLED DATA TO A ML MODEL FOR TRAINING, HERE YOLOV5 IS USED. AFTER 50 EPOCHS THE MODEL WAS ABLE TO ACHIEVE MAX F1 VALUE OF 0.98 WITH CONFIDENCE OF 0.89.

1. INTRODUCTION

Machine learning and neural network are being widely used for the purpose of detecting any objects or creature through a video or real time footage. Various day today activities and task are being automated and performed with the help of combination of these two technologies. Using the same method an animal detection system can be made which can be used specifically for animal detection and identification and its applications. For performing animal detection first, we need to label our training data after which we feed it to our CNN model which produces the output prediction.

Animal Detection–

Animal detection is the process through which a computer is able to detect an animal in a frame of video or image with the help of computer vision and machine learning. Animal detection has a lot of application areas like in monitoring cattle, alarm system in village area near forest to alert users from wild animal invasion, to prevent animal-vehicle collision near forest roads, in monitoring wild animals and their behaviors to understand their ecology and biome in depth.

Labelling Data –

Labelling is the process of marking the objects through outline that we want our CNN model to learn about. In order to perform labelling, an open soft software named labeling (https://github.com/heartexlabs/labelImg) has been used with the help of which we can label the data with ease.

Machine Learning-

It is the science of building intelligent machines that can perform a job strategically better ways than humans. Large training datasets are the backbone of machine learning, allowing it to quickly learn patterns in data and use those patterns in most effective way to solve a problem. Following are the types of machine learning:

1. Supervised learning – In this type of learning we provide data along with expected output to the model so that it can directly map the labels to the data and devise most significant strategy to obtain that output. Classification and regression comes under this learning.
2. Unsupervised Learning- In this learning we only provide dataset to the model and the model is made to find the pattern and derive output on its own. Clustering and association comes under this learning.
3. Reinforced learning – This learning is based on reward system. In this, we provide a