Detecting Changes in Global Biodiversity Using Deep Learning Techniques

<u>Executive Summary</u>

The thesis comprises of all the steps and techniques we have used to successfully complete our capstone project. The most insightful tool was the application of Deep Learning in Convolutional Neural Networks, which enabled us to classify images of certain animals with a decent accuracy on the test dataset. This was achieved by increasing the number of epochs and having a larger dataset. There are alternatives to this approach such as the usage of Support Vector Machines from Machine Learning to classify the animals. Furthermore, after detecting the animal from the test image given, the model further processes the output using Python libraries such as NumPy, Pandas, Seaborn and Matplotlib to get the required outcomes and displays the information of the detected animal in a graphical form, such that it becomes easier to interpret. The populations of the animals along with the years the census has been taken in is stored in a csv file. Our project aims to visualize the change in population of some animals present in Kaziranga National Park over a span of approximately thirty years, provide techniques for diminishing the possibility of their extinction and also show how the protection of certain animals by keeping them in their natural habitat can help protect global biodiversity for decades to come.

CONTENTS

List of Figures

List of Tables

List of Equations

List of Abbreviations

CNN	Convolutional Neural Networks
DL	Deep Learning
ANN	Artificial Neural Network
2D	Two Dimensional

INTRODUCTION

1.1. OBJECTIVE

To identify animal species from Kaziranga National Park through Deep Learning Techniques and predict their chances of survival.

1.2 MOTIVATION

- Our planet is in the midst of its sixth mass extinction, the worst one since dinosaurs roamed the earth, 65 million years ago. Every 14 minutes, an elephant is killed in Africa, while a Rhino is slaughtered every 11 hours.
- The revolutionary advances in Artificial Intelligence (AI) have unlocked the ability to rapidly process a variety of signals, identify risks from them accurately, and provide real-time alerts to the conservationists.
- In this fight against human greed, AI models have turned into an unlikely ally, helping save our biodiversity, paradoxically from our own hands. With the help of this project, we will find the changes in the biodiversity and also the species which have a chance of getting endangered in the near future so that they can be saved by obtaining accurate predictions based on deep learning techniques.

1.3 BACKGROUND

- Kaziranga National Park is a UNESCO World Heritage Place located in Assam spread around a massive area of 430 sq kilometers, and is the home to two thirds of the world’s one horned rhinoceroses.

- The park was established in the year 1908 and currently contains populations of more than 35 mammalian species and 96 species of wetland birds.

- The national park has been granted maximum protection under the Indian law in order to protect the wildlife that lives in it. One of the major concerns of authorities are poaching activities. Preventive measures like anti-poaching camps, patrolling and strict rules for using weapons and firearms have contributed in reducing the number of casualties.

1.4 LITERATURE REVIEW

Serial No.	**Title**	**Author(s)**	**Paper Description**
1.	Deep Learning for Plant Species Classification Survey	Mary Sobha P.G & Princy Ann Thomas	• Deep learning Convolution Neural Networks are utilized to extricate highlights from the pictures of the leaves. • Possible through increased computing power, pretrained models (AlexNet) • Pre trained models could also be used for transfer learning
2.	Deep Image Representations for Coral Image Classification	Ammar Mahmood, Mohammed Bennaumoun, Farid Boussaid, Renae Hovey	• Automation of analysis of large accessible images by deep learning methods. • For optimalization of training time and accuracy, pretrained CNN can be replaced by a faster and more efficient deep network.
3.	Automatic Annotation of Coral Reefs using Deep Learning	Senjian An, Ferdous A Sohel, Gary A Kendrick, Robert B Fisher	• Provides analysis of AUV images by developing advanced deep learning methods for quick automatic labelling of marine animals or species. • Common highlights from VGGent and ResNet were given to order corals and non-corals.

4.	Where is my Deer? – Wildlife Tracking and Counting via Edge Computing and Deep Learning	Bilal Arshad, Johan Barthelemy, Elliott Pilton	• The techniques of Deep CNN, edge computing and online tracking in a field trial will be applied to determine the population density. • Edge computer, infra-red floodlight, motion detection sensor and camera are used.
5.	Towards Automated Visual Monitoring of Individual Gorillas in the Wild	Clemens Alexander Brust, Tilo Burghardt, Milou Greonenberg, Joachin Denzler	• Computer Vision helps to extract gorilla identities by performing automated species detection followed by individual face detection. • Standard deep Learning models combined with SVM classifier are used for this task.
6.	Multipath Deep Shallow Convolutional Networks for Large Scale Plant Species Identification in Wild Image	Syeda Allena Riaz, Imran Razzak and Saaeda Naz	• Identifies different types of plants with MULTI-PATH MULTI DEEP Convolution network. • This model feeds different versions of plant images thus have better image presentation than traditional CNN. • Even the shallow network showed considerably better performance as compared to pretrained deep learning models of AlexNet, GoogLeNet, and VGGNet.

7.	Improving Transfer Learning & Squeeze and Excitation Networks for Small-Scale Fine-Grained Fish Image Classification	Chenchen Qui, Shaoyong Zhang, Chao Wang, Zhibin Yu, Haiyong Zheng	• Proposes an improved transfer learning method with refined squeeze-and-excitation networks, which is the flip and rotate technique for fine-grained fish image classification on small-scale datasets. • While contrasting the presentation of this technique and regularly utilized CNNs on limited fine-grained datasets, to be specific, Croatian and QUT fish datasets, their outcomes were better.
8.	Investigation of Different CNN Based Models for Improved Bird Sound Classification	Jie Xie, Kai Hu, Mingying Zhu, Jinghu Yu	• Two sorts of techniques are thought about, in the first, a similar profound learning architecture with various input sources is utilized, and in the second, two diverse deep learning structures for developing the interfused models are utilized. • Three TFR or Time frequency representations are used. These are Mel-spectrogram, harmonic-component based spectrogram, and percussive-component based spectrogram. The second deep learning architecture SubSpectralNet is fused to TFR to give a more accurate result.

9.	Fish Recognition from a vessel camera using deep convolutional neural network and data augmentation	Ziqiang Zheng, Chunfeng Guo, Xueer Zheng, Zhibin Yu, Weiwei Wang	• A method is presented where anarea of the fish is masked and then only the masked part is pushed to the next step. • A rotation-based data augmentation method is used to avoid over-fitting which may be caused by the imbalanced training dataset.Finally, the AlexNet, GoogLeNet, Caffenet and VGGNet neural network is employed for the classification; the results successfully enhanced the classification performance.
10.	Convolution Neural Networks for Rhino's Dolphins Identification	Rosalia Maglietta, Vito Reno, Rocco Caccioppoli, Emanuelle Seller	• Novelty of this paper is the improvement of another strategy dependent on deep learning, called as the Neural Network Pool (NNPool). This new strategy additionally incorporates the special capacity of perceiving unrevealed versus known dolphins in tremendous datasets with no communication by the client. The results were faster as compared to traditional Convolution Neural Networks.
11.	Deep Learning for Large Scale Biodiversity Monitoring	David J Klien, Matthew W McKown, Bernie R Tershy	• Focused on processing and analyzing large and high-rate datasets such as audio and image streams. • Custom user interfaces are used to greatly speed up exploration

			along with Machine Learning techniques.
12.	Using Deep Convolutional Networks for Species Identification of Xylotheque Samples	Geovanni Figueoroa Mata, Erick Mata Montero, Juan Carlos Valverde Otarola	• A Deep Convolution Neural Network for automated woods species identification dependent on macroscopic pictures is introduced. • At first, an implementation and study of a modified version of the LeNet convolutional network is performed, which is trained from scratch with a database of macroscopic images of 41 forest species of Brazil. • With this network an accuracy of 93.6% was achieved. Additionally, after fine-tuning the Resnet50 model with pre-trained weights on IMAGENET an accuracy of 98.03% was achieved.
13.	An Automated Vertebrate Animals Classification Using Deep Convolution Neural Networks	Nidhal K. El Abbadi, Elham Mohammed Thabit	• The paper suggests using Deep Convolutional Neural Network (CNN) to detect and classify the animals (vertebrate classes) in digital images. • The dataset comprises of 12000 unique pictures, 9600 pictures for the testing step, and the rest

			pictures (2400) for the training step. • After applying the proposed of deep CNN, the best number of epochs was 100. The absolute performance of the outcomes came to 97.5%.
14.	Animal Recognition and Identification with Deep Convolutional Neural Networks For Automated Wildlife Monitoring	Hung Nguyen, Sarah J. Maclagan, Tu Dinh Nguyen, Paul Flemons	• A single-labeled dataset from Wildlife Spotter project is used along with the state-of-the-art deep convolutional neural network architectures, to train a computational system capable of filtering animal images and identifying species automatically. • The experimental results achieved an accuracy at 96.6% for the task of detecting images containing animal, and 90.4% for identifying the three most common species among the set of images of wild animals(bird, rat and bandicoot) taken in South-central Victoria, Australia. • On comparison to ResNet-50 which had an accuracy of 87.97% on the same dataset the paper's accuracy is 90.4%.

15.	Wildlife Detection and Recognition in Digital Images Using YOLOv3	Mina Gabriel, Sangwhan Cha, Nushwan Yousif B , Daqing Yun	• Deep Learning methods are being used to detect/ recognize wildlife from digital images. • Test dataset consists of 1065 images and 4 classes of animals. • YOLOv3 and YOLOv3-Tiny detect and classify several animals with 752% and 68.4% mean average precision.
16.	Deep Convolutional Neural Networks for Shark Behavior Analysis	Wenlu Zhang, Anthony Martinez, Emily Nicole Meese, Christopher G. Lowe, Yu Yang	• Accelerometer data loggers have been used to measure activities of sharks over a long period of time. • VGGNet, the extension of Alexnet is being used for a deep neural network comprising of 16 to 19 layers. • Random Forest classifier is used for comparison.
17.	Deep Learning for Inexpensive Image Classification of Wildlife on the Raspberry Pi	Brian H Curtin, Suzanne J Matthews	• A Raspberry Pi based camera system is being used for detecting wildlife, in which localized image recognition enables pictures of desired animals to be transferred to the user. • The efficacy is tested and the model is focused on snow leapords • CNN model is built using TensorFlow and Keras.

18.	Feature Extraction using Convolution Neural Networks (CNN) and Deep Learning	Manjunath Jogin, Mohana, Meghana RK, Apoorva S	• Classifiers are being used for image classification and counting, some examples like kNN Classifier, Linear Classifier and SoftMax Classifier. • The results obtained show that the accuracy is 85.97% for image classification, and can be used for video surveillance and security related applications.

2. PROJECT DESCRIPTION AND GOALS

2.1 DATASET

There are 2 different datasets in our project, the first one consists of images of different animals and the second is the census details which we obtained from the Kaziranga National Park website (https://www.kaziranga-national-park.com/).

The image dataset consists of 2617 images belonging to 3 classes in the training set and 658 images belonging to 3 classes in the testing set. These were obtained from two different websites: https://www.kaggle.com/ and https://unsplash.com/.

Examples of datasets:

Figure 1: Example of datasets

2.2 METHODOLOGY

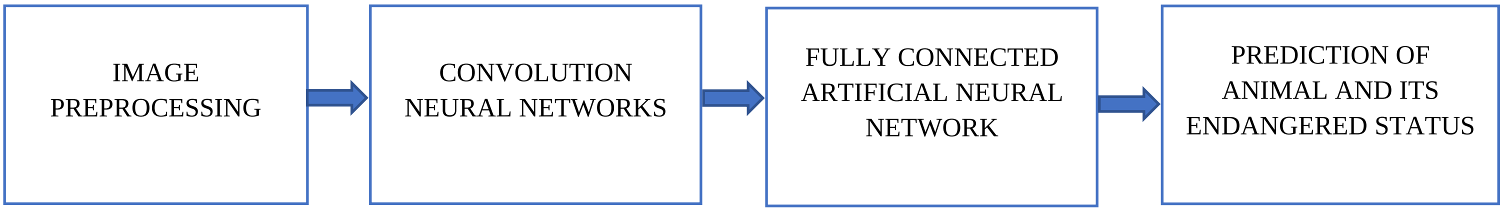


Figure 2: Methodology

- IMAGE PREPROCESSING:

  The training dataset has been rescaled to “1./255” and resized to 128*128. Since our dataset is less, we have 2 techniques to increase the number of images. The first one is to zoom in by scale 0.2 and the second is to flip the images horizontally in a batch of 32 images.

  For example:

Figure 3: Original Image

Figure 4: Preprocessed images

- CONVOLUTION NEURAL NETWORKS

  There are a total of 4 convolution layers to extract or filter out features from the images from a 3-dimensional image array to a 2-dimensional output array which will the input to the fully connected Artificial Neural Network.
  The CNN consists of 4 layers: The Input Layer, Convolution layer, Activation Layer and the pooling layer, post which we get a 2-Dimensional output.

- FULLY CONNECTED ARTIFICIAL NEURAL NETWORK

  This layer resembles basic neural network layer that takes contribution from the past layer or the Convolution Neural Network layer and yields the 1-D cluster of size equivalents to number of classes to finally give an output of

- PREDICTION OF ANIMAL AND ITS ENDANGERED STATUS

  At the last step which is after training the model, an image input is given and the model predicts which animal it is and finally gives the details of its population in previous years according to which the future prediction and the endangered status is given as output. All the parameters are graphically represented.

2.3 EXPECTED OUTCOMES

- The Deep Learning model will help us in recognizing endangered animals and hence save them from extinction.
- These models can be incorporated on different image datasets to get good results.
- We could also make our trained model public so that the community can use them without training the models, and save a lot of time.
- These trained models can be used in transfer learning to classify similar datasets.

3. TECHNICAL SPECIFICATIONS

Convolutional Neural Network (CNN)

CNN is a deep learning algorithm that takes in input as an image, assign importance (bias and weights) to different aspects of the image and be able to differentiate one from another. In primitive methods filters are hand engineered, but conv networks have the ability to learn these filters. Conv networks successfully captures the temporal and spatial dependencies in the image through application of different filters. Convnet consists of sequence of layers. Every layer transforms one volume to another through differentiable functions. Below is the description of different layers by taking example of running a conv. network on image of size 128*128*3:

- Input Layer: This layer holds raw input image of height 128px, width 128px and depth 3px.
- Convolutional Layer: Output volume is computed by this layer by computing dot product between all the filters and image patch. This layer consists of set of learnable filters. Each filter has small height and width and depth same as that of input image. If for this layer we use 128 filters, then output will be of shape 128*128*3.
- Activation Layer: Element wise activation to the output of convolutional layer is providing by this layer. Some of the common activation functions are –Relu, Sigmoid, tanh, leaky Relu. The volume will remain unchanged after this layer.
- Pooling Layer: The main function of this layer is to reduce the size of volume which makes computation fast and also reduces over fitting. This layer is periodically inserted in convnet. Max pooling and average pooling are the two most commonly used types of pooling. If max pool is used with 2*2 filters and strides of 2, dimension of resultant volume will be 16*16*2.
- Fully-connected Layer: This layer is like common neural network layer also known as an artificial neural network layer that takes input from the previous layer and outputs the 1-D array of size equals to number of classes.

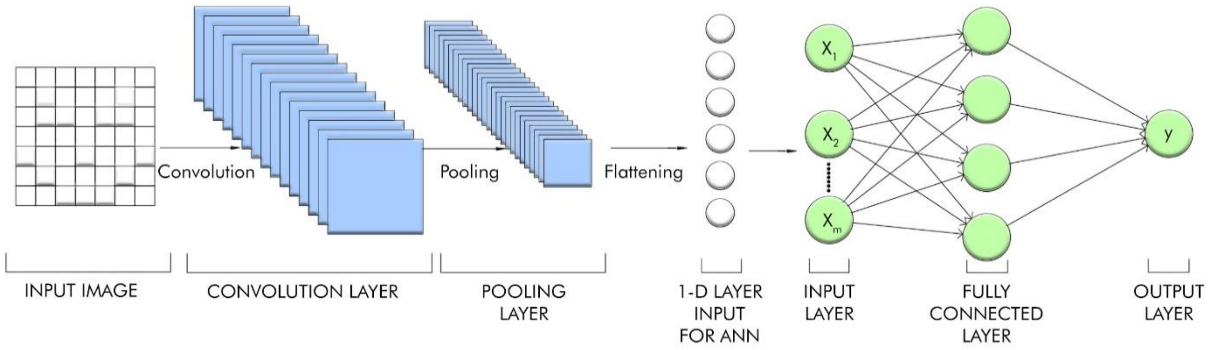


Figure 5: Convolution Neural Network (Source: https://www.superdatascience.com/)

Artificial Neural Networks (ANN)

ANN or Artificial Neural Networks are constructed like the human cerebrum, with neuron hubs or nodes are interconnected like a spider's web. The human cerebrum has many billions of cells which are called neurons. Every neuron is comprised of a cell body that is liable for handling data via conveying data towards and away from the cerebrum.

A simple ANN has around hundreds or even thousands of Artificial Neurons called processing or refining units, which are interrelated by nodes. These processing or refining units are comprised of input and output nodes. The input nodes get different structures and designs of data dependent on an internal weighting framework. The Artificial Neural Network tries to find out about the data introduced to process one output report. Exactly like humans need rules and regulations to think of an outcome or result, ANNs, likewise utilize a set of learning regulations called as backpropagation, a feedback of the slight error which is later added up to the input, to increase the efficiency of the output result.

- INPUT: The Input of the layer is a 1-D array.
- HIDDEN LAYER: The hidden layers are present in-between the input and the output layers. This can be referred to as the processing unit of the ANN. It performs the calculations according to the weights to find hidden features and patterns with the help of the following combination function:

$$\sum_{i=1}^{n} Wi * Xi + b$$

Equation 1: Combination Function

The Artificial Neural Network (ANN) takes the whole input and computes, with the help of the combination function, the weighted sum of all the inputs and includes a bias(b), shown in the above equation.

- OUTPUT: The 1-D array input is passed through a series of hidden layers in which the weights of features is calculated and finally a total weighted output is determined.

  This output is passed through another function known as Activation Function to finally compute the output. The role of the Activation Function is to decide which output node is to be fired.

Activation Function

- SIGMOID ACTIVATION FUNCTION: The sigmoid function is used when the output comprises of 2 nodes. Therefore, to predict the probability the sigmoid function is used. Since the probability exists in the range of 0 and 1, sigmoid activation function is the right choice. As soon as the probability is computed the node with the higher probability is fired.

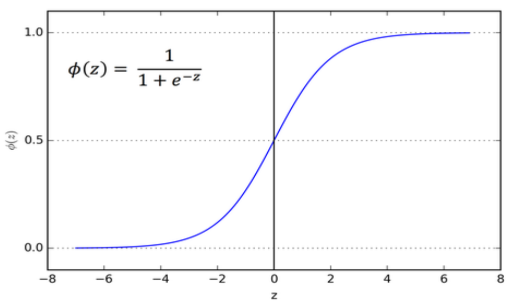
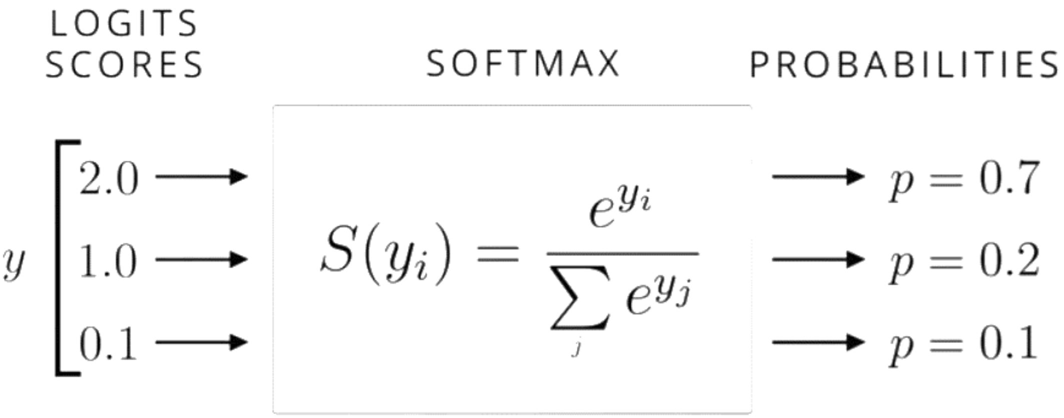


Figure 6: Sigmoid Activation Function

- SOFTMAX ACTIVATION FUNCTION: The SoftMax function or normalized exponential function, is used in Neural Networks to normalize the output from the ANN to a probability distribution over the output classes as the last activation function.



Equation 2: SoftMax Activation Function

Python Libraries

- Numpy, which stands for Numerical Python is a library which is being used when we work with arrays, and it overcomes the slow processing speeds of python lists. This is because they are being stored at a continuous place in memory which enables quick access and manipulation.
- Pandas is a python library which is used to analyze, clean, explore and manipulate data obtained from data sets. Moreover, it allows us to analyze big data and make conclusions.
- Matplotlib and Seaborn are graph plotting libraries in python which enable a visualization utility.

4. DESIGN APPROACH AND DETAILS

4.1 DESIGN APPROACH

Using CNN:

➢ **Binary Classification:** We used two layers of CNN Model. In this architecture, we have a conv2D layer followed by MaxPooling2D layer which is followed by a flattening layer. We had two such sets of layers. After these two sets of layers, a fully connected layer is there having activation function as "relu". Finally in the output layer, we had two neurons having "Sigmoid" activation function. We have taken Adam as optimizer and binary-cross entropy as loss function. For training the model we ran 20 epochs. Complete Architecture is shown below:

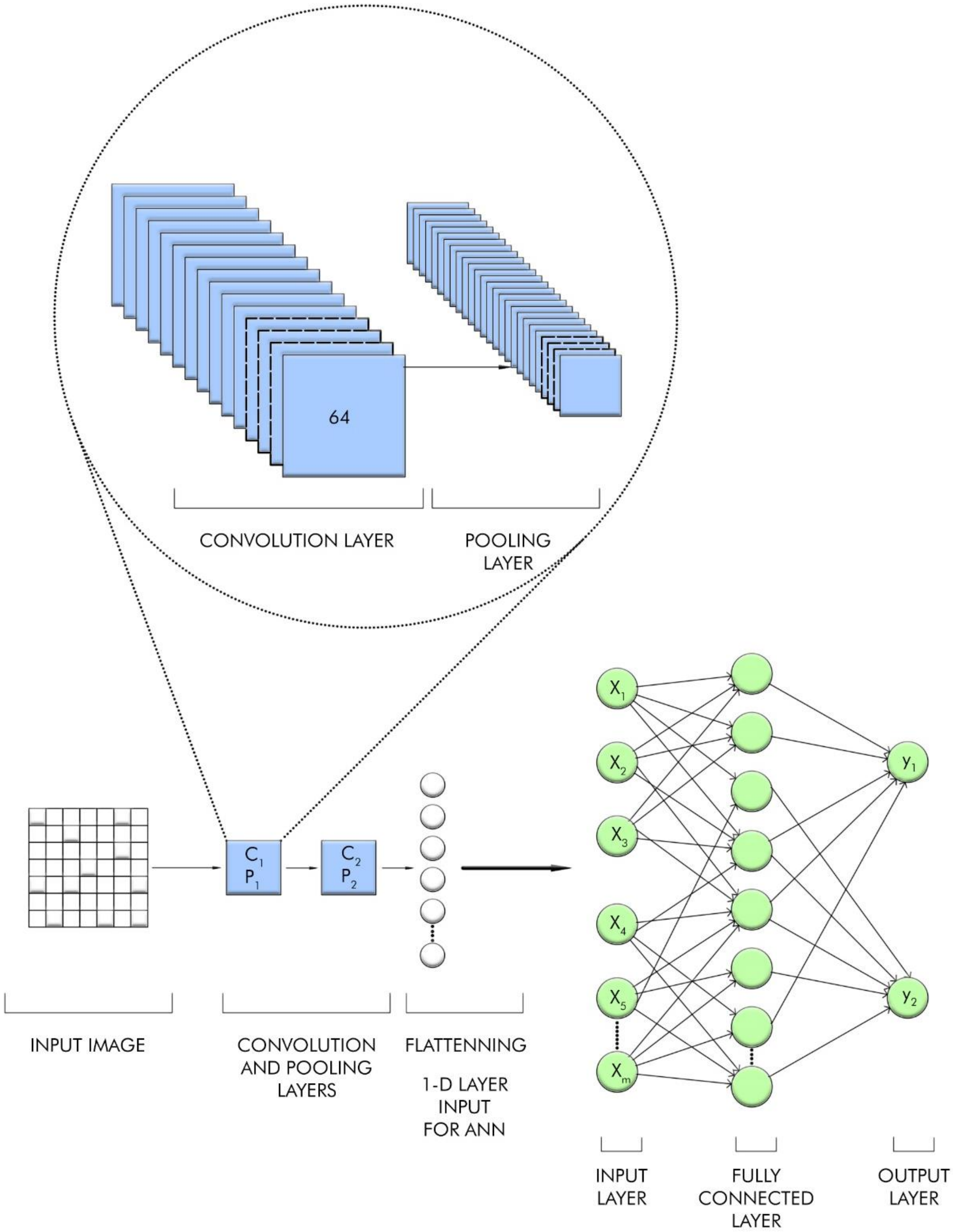


Figure 7: Binary Classification using CNN (Source: https://www.superdatascience.com/)

- **Multilevel Classification**: We have used four layers of CNN Model. In this architecture, same as the previous one, we have conv2D layer followed by Maxpooling2D layer which is followed by a flattening layer. Here we have four such sets of layers. After these four sets of layers, a fully connected layer is there having "relu" as the activation function. Here the output layer contains three neurons as number of classes are three, with "SoftMax" as the activation function. Here the number of filters used at different layers is different from that of binary classification part. We have taken Adam as optimizer and categorical-cross entropy as loss function. For training the model we ran 30 epochs. Complete Architecture is shown below:

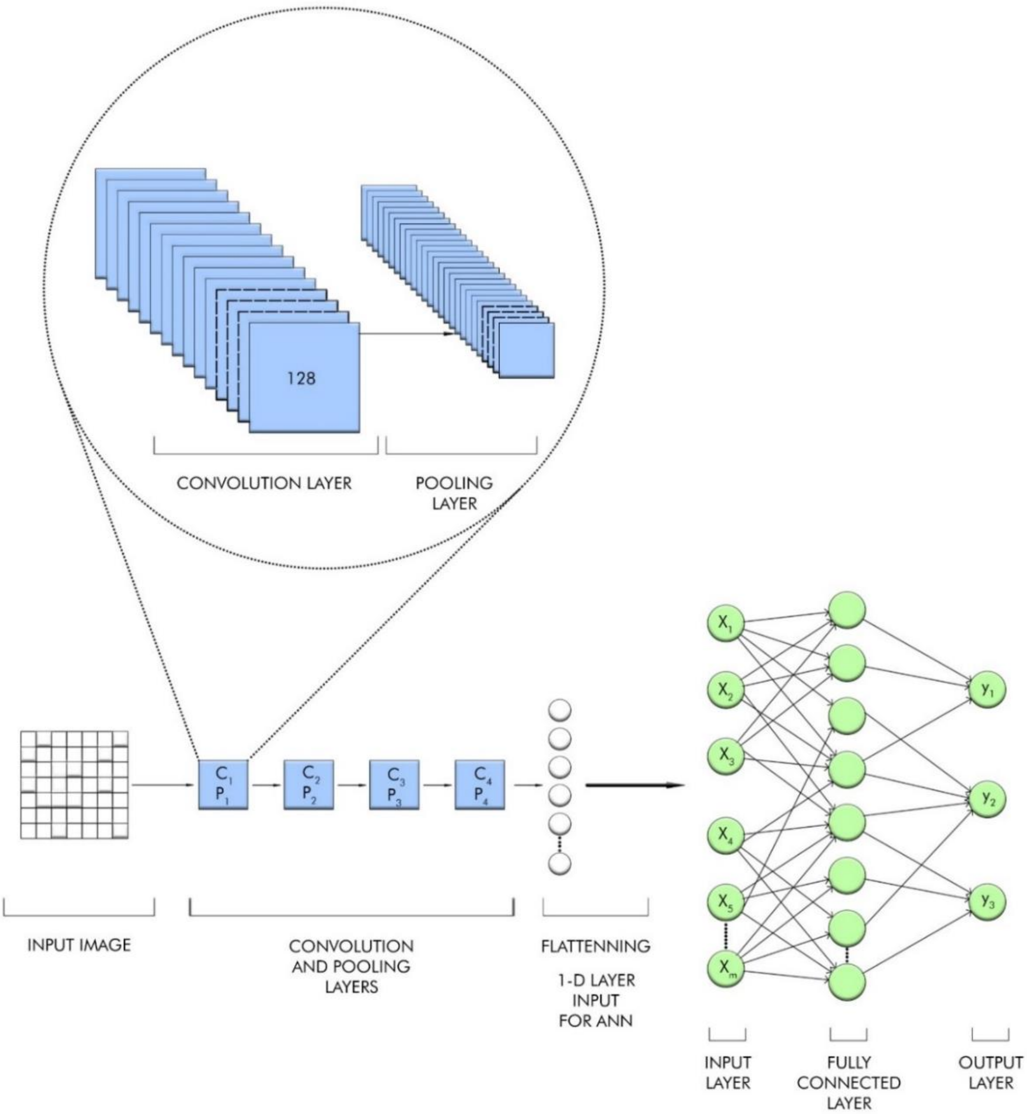


Figure 8: Multilevel Classification using CNN (Source: https://www.superdatascience.com/)

Using Python Libraries:

- **NumPy and Pandas:** We have made use of the read csv function which enables us to track the details (population over a certain span of years) of the detected animal and store the data in arrays which would then be used for giving a visual output for a better interpretation of the animal population.

- **Matplotlib and Seaborn:** The desired output is being stored in arrays which comprise of the X and Y axis, and by using the plt function of the given libraries, the population v/s year graph of the animal is shown.

Algorithm/Steps followed in the code:

- The images from the dataset are taken for preprocessing.
- The training as well as the testing dataset is rescaled to "1./255" and the resized to 128*128.
- Two techniques are used to increase the number of images. The first is to zoom in by scale 0.2 and the second is to flip the pictures on a level plane in a cluster of 32 pictures.
- For the training of our model, Convolution Neural Network is used.
- There are a total 4 layers of convolution and pooling operations to extract features and summing it into a smaller feature map.
- The feature map is flattened to a 2-Dimentional layer.
- The 2-D layer is the input of the fully connected Artificial Neural Network which combines the features provided in the input.
- The attributes of the combination of the features helps in classifying the images using the ReLU activation function.
- The final output of the classification is shown using the SoftMax function and the cross-entropy function.
- After predicting the class, we get the population data from the csv file stored in the computer.
- Store the population and year of the detected animal in 2 different arrays.
- Use the population array as the y axis and the year array as the x axis while plotting a bar graph for population v/s year statistics using matplotlib library.
- Show the initial condition of the animal (data acquired from Wikipedia).

- Calculate the percentage change in the population levels between subsequent population censuses taken of the animal detected, and store it in an array.
- Run a for loop to find out whether the percentage changes in the population were negative, positive or zero (no change).
- If the changes were negative or neutral, then append the required lists, and use ‘length’ option to find out how many values were negative or neutral.
- If the population has increased (i.e., changes are positive), then run an if else conditional statement to figure out whether the increment was by a large margin or a comparatively smaller one.
- In this way, we can somewhat predict whether the chances of survival of the detected animal are moderate or high.
- Store the starting and ending years in which population censuses were taken in 2 different NumPy arrays. These are to be used as the first 2 columns for the Population increment/decrement statistics table.
- Print the starting year, ending year, population change and increment decrement in a tabular form.
- In order to give an estimate of future populations of the detected animal, store the percentage change of population between the first and last year of the census taken.
- Assume constant population change, and predict what the future population will look like based upon population trends in the past.
- Store the population in an array, and the year in another array.
- Print a line graph showing the future trends of the population with year in the y axis and population in the x axis.

5. SCHEDULE, TASKS AND MILESTONES

Schedule:

- February: Finalizing the problem statement, acquiring of images and datasets for the animals (both test and train)
- March: Making the Deep Learning Model using Convolutional Neural Networks for the detection of the animal present in the image, and further train the model with training dataset
- April (first half): Using the test dataset of the model to predict the possibility of the detected animal going extinct

- April (last two weeks): Obtaining the output and the compilation of results
- May (first week): Report completion and submission.

Tasks & Milestones:

- Initial work:
    - Construct a table on the population census (taken in various years) of the animals present in Kaziranga National Park, Assam.
    - Acquire the datasets (Testing and Training images) of these animals.
- Application of Deep Learning:
    - Design a deep learning model using Convolutional Neural Networks.
    - Train the model using the Train images dataset.
    - Test the model using the Test images dataset to find out the accuracy.
    - After the identification of the animal, find out its population history from the population census table.
- Obtaining output and results:
    - Depending on the population increment/ decrement over time and initial category of animal (Vulnerable, endangered etc.), give insights on chances of survival, or methods to prevent extinction.

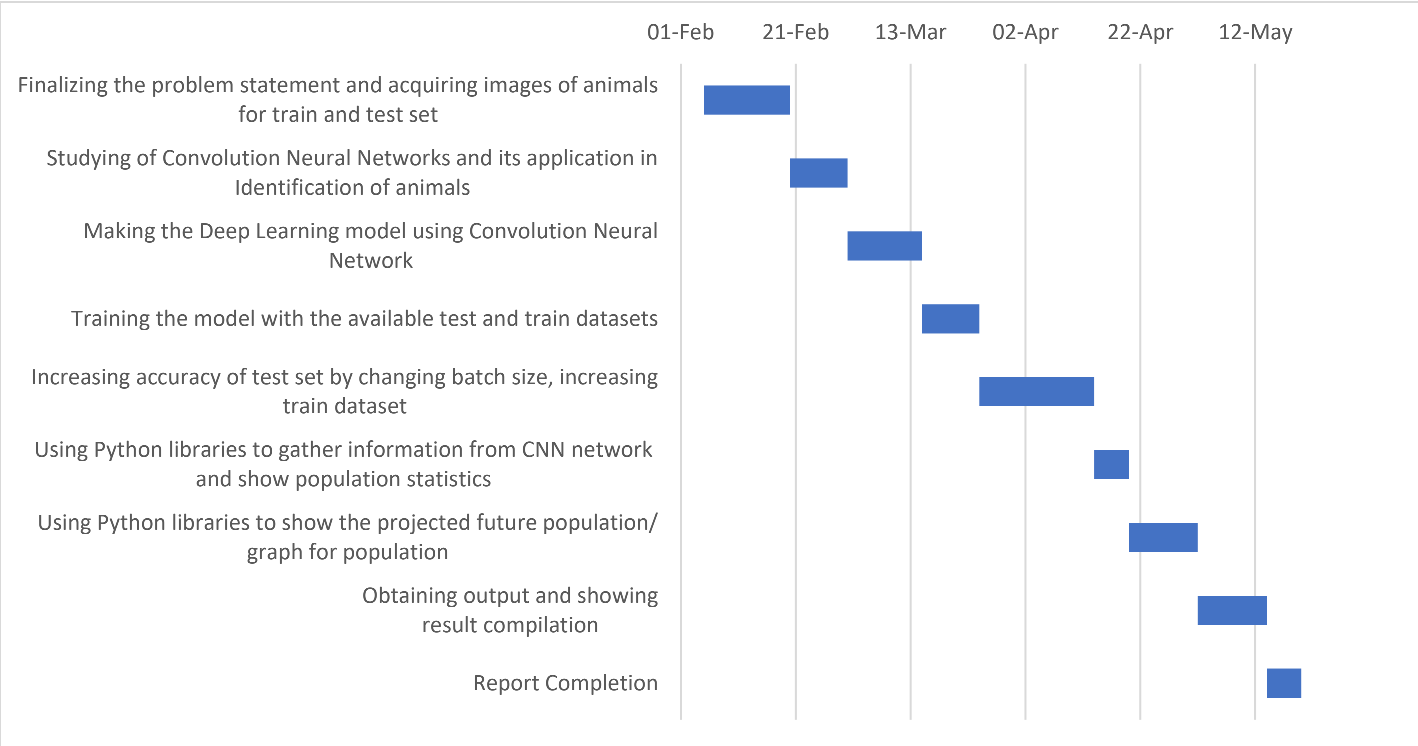


Figure 9: Gantt Chart

6. PROJECT DEMONSTRATION

Model Training:

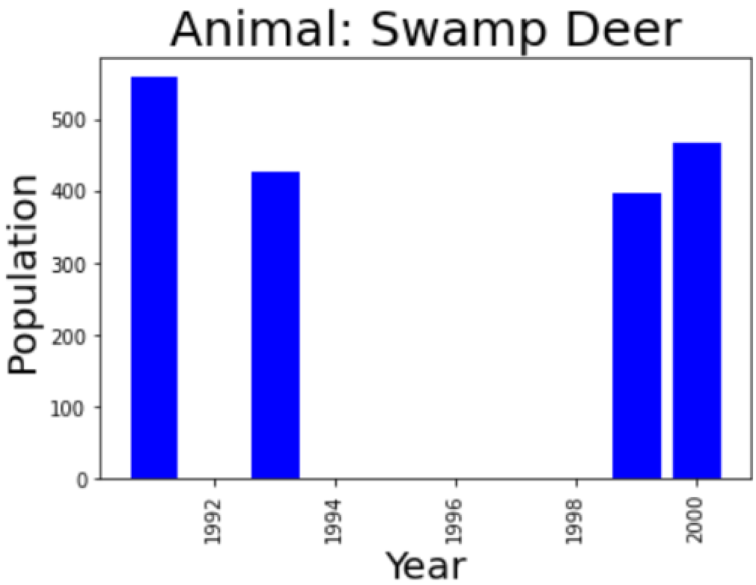
```
Found 2617 images belonging to 3 classes.
Found 658 images belonging to 3 classes.
Epoch 1/30
82/82 [==============================] - 675s 8s/step - loss: 1.0017 - accuracy: 0.5845 - val_loss: 0.9217 - val_accuracy: 0.6201
Epoch 2/30
82/82 [==============================] - 292s 4s/step - loss: 0.9010 - accuracy: 0.6155 - val_loss: 1.1940 - val_accuracy: 0.6201
Epoch 3/30
82/82 [==============================] - 288s 4s/step - loss: 0.8373 - accuracy: 0.6277 - val_loss: 0.7160 - val_accuracy: 0.6900
Epoch 4/30
82/82 [==============================] - 288s 4s/step - loss: 0.7231 - accuracy: 0.6698 - val_loss: 0.7346 - val_accuracy: 0.7036
Epoch 5/30
82/82 [==============================] - 292s 4s/step - loss: 0.6863 - accuracy: 0.6983 - val_loss: 0.8051 - val_accuracy: 0.6444
Epoch 6/30
82/82 [==============================] - 291s 4s/step - loss: 0.6343 - accuracy: 0.7381 - val_loss: 0.7186 - val_accuracy: 0.6596
Epoch 7/30
82/82 [==============================] - 294s 4s/step - loss: 0.5765 - accuracy: 0.7619 - val_loss: 0.6191 - val_accuracy: 0.7340
Epoch 8/30
82/82 [==============================] - 290s 4s/step - loss: 0.5422 - accuracy: 0.7706 - val_loss: 0.5679 - val_accuracy: 0.7644
Epoch 9/30
82/82 [==============================] - 292s 4s/step - loss: 0.4703 - accuracy: 0.8137 - val_loss: 0.5845 - val_accuracy: 0.7508
Epoch 10/30
82/82 [==============================] - 290s 4s/step - loss: 0.4656 - accuracy: 0.8145 - val_loss: 0.5141 - val_accuracy: 0.7933
Epoch 11/30
82/82 [==============================] - 290s 4s/step - loss: 0.4377 - accuracy: 0.8252 - val_loss: 0.6464 - val_accuracy: 0.6991
Epoch 12/30
82/82 [==============================] - 293s 4s/step - loss: 0.4184 - accuracy: 0.8235 - val_loss: 0.5128 - val_accuracy: 0.7827
Epoch 13/30
82/82 [==============================] - 291s 4s/step - loss: 0.3743 - accuracy: 0.8560 - val_loss: 0.4914 - val_accuracy: 0.8237
Epoch 14/30
82/82 [==============================] - 290s 4s/step - loss: 0.3816 - accuracy: 0.8469 - val_loss: 0.4727 - val_accuracy: 0.8146
Epoch 15/30
82/82 [==============================] - 292s 4s/step - loss: 0.3016 - accuracy: 0.8877 - val_loss: 0.6491 - val_accuracy: 0.7416
Epoch 16/30
82/82 [==============================] - 292s 4s/step - loss: 0.2680 - accuracy: 0.8936 - val_loss: 0.5204 - val_accuracy: 0.7903
Epoch 17/30
82/82 [==============================] - 289s 4s/step - loss: 0.2972 - accuracy: 0.8838 - val_loss: 0.5484 - val_accuracy: 0.8070
Epoch 18/30
82/82 [==============================] - 288s 4s/step - loss: 0.2638 - accuracy: 0.8994 - val_loss: 0.6684 - val_accuracy: 0.7553
Epoch 19/30
82/82 [==============================] - 287s 4s/step - loss: 0.2467 - accuracy: 0.9094 - val_loss: 0.5858 - val_accuracy: 0.7964
Epoch 20/30
82/82 [==============================] - 288s 4s/step - loss: 0.2448 - accuracy: 0.9004 - val_loss: 0.4944 - val_accuracy: 0.8389
Epoch 21/30
82/82 [==============================] - 289s 4s/step - loss: 0.1964 - accuracy: 0.9231 - val_loss: 0.4651 - val_accuracy: 0.8450
Epoch 22/30
82/82 [==============================] - 287s 3s/step - loss: 0.1541 - accuracy: 0.9414 - val_loss: 0.7166 - val_accuracy: 0.7629
Epoch 23/30
82/82 [==============================] - 292s 4s/step - loss: 0.2163 - accuracy: 0.9192 - val_loss: 0.5807 - val_accuracy: 0.8009
Epoch 24/30
82/82 [==============================] - 293s 4s/step - loss: 0.1630 - accuracy: 0.9447 - val_loss: 0.5596 - val_accuracy: 0.8207
Epoch 25/30
82/82 [==============================] - 294s 4s/step - loss: 0.1695 - accuracy: 0.9379 - val_loss: 0.5699 - val_accuracy: 0.8450
Epoch 26/30
82/82 [==============================] - 293s 4s/step - loss: 0.1428 - accuracy: 0.9470 - val_loss: 0.6352 - val_accuracy: 0.8222
Epoch 27/30
82/82 [==============================] - 290s 4s/step - loss: 0.1391 - accuracy: 0.9530 - val_loss: 0.5789 - val_accuracy: 0.8511
Epoch 28/30
82/82 [==============================] - 288s 4s/step - loss: 0.1186 - accuracy: 0.9564 - val_loss: 0.5315 - val_accuracy: 0.8495
Epoch 29/30
82/82 [==============================] - 292s 4s/step - loss: 0.1290 - accuracy: 0.9534 - val_loss: 0.6294 - val_accuracy: 0.7812
Epoch 30/30
82/82 [==============================] - 289s 4s/step - loss: 0.1127 - accuracy: 0.9559 - val_loss: 0.5679 - val_accuracy: 0.8541
INFO:tensorflow:Assets written to: /content/drive/MyDrive/CAPSTONE/Model/assets
```

Figure 10: Screenshot of Epochs processed along with test/train accuracy

An epoch defines how many times the algorithm designed will go through the entire dataset. In our Convolution Neural Network, we have chosen the number of epochs to be 30. Increasing this number is linked with obtaining higher accuracy on the test dataset, but consequently going through the data will take a longer time. By using

Google Collaboratory, we have run our CNN algorithm with 30 epochs and obtained an accuracy of 85.41% on the test set.

Case 1: If the detected animal is Swamp Deer



Initial condition: Vulnerable

Figure 11: Graphical representation of population of Swamp Deer

The above graph (Figure 10) depicts the population v/s year statistics of the Swamp Deer, during the years in which an official population census was carried out for the same.

Year (from)	Year (to)	Population Change (%)	Population +/-
1991	1993	-23.613596	Decrement
1993	1999	-6.791569	Decrement
1999	2000	17.587940	Increment
1991	2000	-16.279070	Decrement

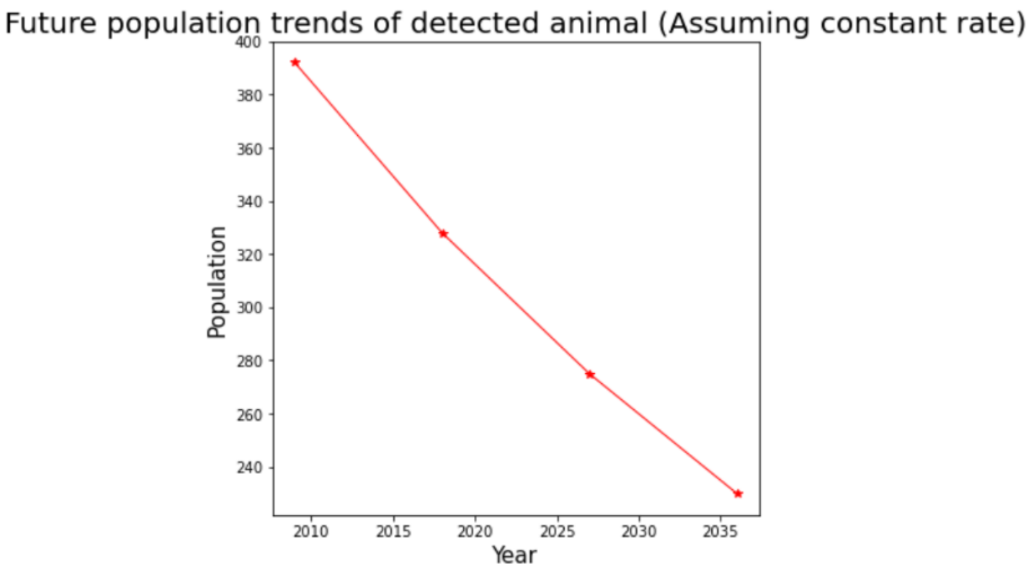
Table 1: Tabular representation of population change (%) for Swamp Deer

The above printed table (Table 1) shows the percentage increment or decrement of the population of the detected animal (In this case, Swamp Deer) between any two subsequent population censuses. The last row depicts the overall population change, i.e., between the first and last census taken.

Year	Population (Estimated)
2009	392.0
2018	328.0
2027	275.0
2036	230.0

Table 2: Population projection for the future (Swamp Deer)

In this table (Table 2), provided same trends are being followed for the population change, we have shown an estimate of what the population of the detected animal (here, Swamp Deer) will be like in the future.

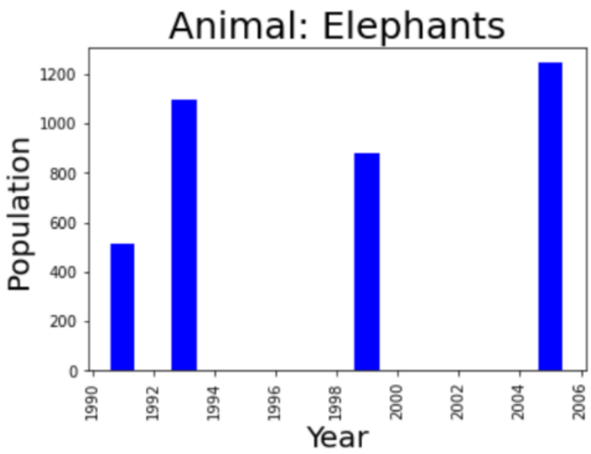



```
Possibilty of surviving in the future:  Has very less chances of survival, ie an endangered species
```

Figure 12: Graphical Population projection for the future (Swamp Deer)

The above graph (Figure 11) shows the future trend of the animal population (data was gathered from the table above, Fig 13), at the end a prediction is made on whether the species will survive or not by examining their projected population for the coming decades.

Case 2: Elephants



Initial condition: Endangered

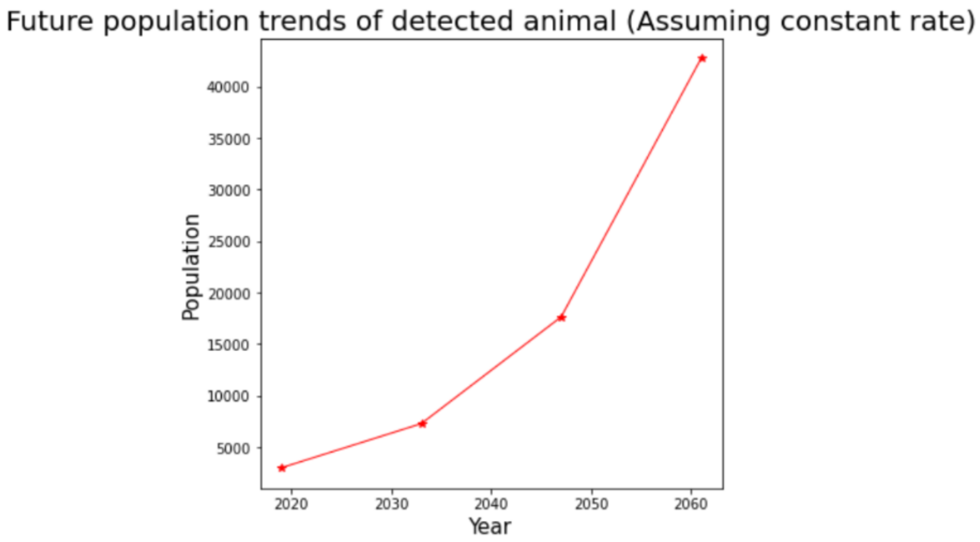
Figure 13: Graphical representation of population of Elephant

Year (from)	Year (to)	Population Change (%)	Population +/-
1991	1993	112.427184	Increment
1993	1999	-19.378428	Decrement
1999	2005	41.269841	Increment
1991	2005	141.941748	Increment

Table 3: Tabular representation of population change (%) for Elephant

Year	Population (Estimated)
2019	3015.0
2033	7294.0
2047	17646.0
2061	42693.0

Table 4: Population projection for the future (Elephant)




```
Possibilty of surviving in the future:  moderate to good chances of survival
```

Figure 14: Graphical Population projection for the future (Elephant)

7. SUMMARY

Deep learning, being a subfield of Machine Learning has various applications and is used in plethora of ways. In our project, we have used CNN's (part of DL) to help identify certain animal species which are homed in Kaziranga National Park, a UNESCO declared World Heritage Place in Assam, India. The model takes test images of these animals as inputs to further process, and provide the animal's name. In practical terms, the images of animals can be obtained by drone imaging, or aerial imaging whenever a survey is being carried out for various purposes such as land quality check etc. By using certain techniques such as changing of batch size, increase in test datasets, altering the number of hidden CNN layers, we have successfully achieved a decent level of accuracy (Above 80%) on the CNN models test set.

After the model has detected the animal, it will read through the population census of the animal from a csv file using Python libraries such as Pandas and NumPy. Furthermore, a graph will be displayed using Matplotlib and Seaborn, which will show the population statistics of the animal detected by the CNN, along with the year-on-year percentage increment or decrement of the population. In this visual representation, it will be easier to take surveys and assess whether wildlife in Kaziranga National Park is being protected or not. By this, we can maintain an ecological balance and prevent the extinction of the animals which are being hunted down and killed for commercial purposes.

8. REFERENCES

[1] Mary Sobha P.G & Princy Ann Thomas, “Deep Learning for Plant Species Classification Survey”, IEEE.

[2] Ammar Mahmood, Mohammed Bennaumoun, Farid Boussaid, Renae Hovey, “Deep Image Representations for Coral Image Classification”, IEEE Journal of Oceanic Engineering 2018.

[3] Senjian An, Ferdous A Sohel, Gary A Kendrick, Robert B Fisher, “Automatic Annotation of Coral Reefs using Deep Learning”, IEEE 2016.

[4] Bilal Arshad, Johan Barthelemy, Elliott Pilton, “Where is my Deer? – Wildlife Tracking and Counting via Edge Computing and Deep Learning”, IEEE 2020.

[5] Clemens Alexander Brust, Tilo Burghardt, Milou Greonenberg, Joachin Denzler, “Towards Automated Visual Monitoring of Individual Gorillas in the Wild”, IEEE International Conference on Computer Vision Workshop 2017.

[6] Syeda Allena Riaz, Imran Razzak and Saaeda Naz, “Multipath Deep Shallow Convolutional Networks for Large Scale Plant Species Identification in Wild Image”, IEEE 2020.

[7] Chenchen Qui, Shaoyong Zhang, Chao Wang, Zhibin Yu, Haiyong Zheng, “Improving Transfer Learning & Squeeze and Excitation Networks for Small-Scale Fine-Grained Fish Image Classification”, IEEE Access 2018.

[8] Jie Xie, Kai Hu, Mingying Zhu, Jinghu Yu, “Investigation of Different CNN Based Models for Improved Bird Sound Classification”, IEEE Access 2019.

[9] Ziqiang Zheng, Chunfeng Guo, Xueer Zheng, Zhibin Yu, Weiwei Wang, “Fish Recognition from a vessel camera using deep convolutional neural network and data augmentation”, IEEE 2018.

[10] Rosalia Maglietta, Vito Reno, Rocco Caccioppoli, Emanuelle Seller, “Convolution Neural Networks for Rhino’s Dolphins Identification”, IEEE Access 2020.

[11] David J Klien, Matthew W McKown, Bernie R Tershy, “Deep Learning for Large Scale Biodiversity Monitoring”, Bloomberg Data for Good Exchange Conference 2015.

[12] Geovanni Figueoroa Mata, Erick Mata Montero, Juan Carlos Valverde Otarola, “Using Deep Convolutional Networks for Species Identification of Xylotheque Samples”.

[13] Nidhal K. El Abbadi, Elham Mohammed Thabit, “An Automated Vertebrate Animals Classification Using Deep Convolution Neural Networks”, International Conference on Computer Science and Software Engineering, 2020.

[14] Hung Nguyen, Sarah J. Maclagan, Tu Dinh Nguyen, Paul Flemons, “Animal Recognition and Identification with Deep Convolutional Neural Networks For Automated Wildlife Monitoring”, International Conference on Data Science and Advanced Analytics 2017.

[15] Guobin Chen, Tony X. Han, Zhihai He, Roland Kays, Tavis Forrester, “Deep Convolutional Neural Network Based Species Recognition for Wildlife Monitoring”, IEEE 2014.

[16] Enkeleda Bocaj, Dimitris Uzunidis, Panagoitis Kasnesis, Charalampos Z Patrikakis, “On the Benefits of Deep Convolutional Neural Networks on Animal Activity Recognition”, IEEE 2020.

[17] Fauzan Nusyura, Ketut Eddy Purnama, Reza Fuad Rachmadi, “Transfer Learning for Recognizing Face in Disguise”, International Seminar on Intelligent Technology and its Applications 2020.

[18] Mina Gabriel, Sangwhan Cha, Nushwan Yousif B , Daqing Yun, “Wildlife Detection and Recognition in Digital Images Using YOLOv3”, IEEE Cloud Summit 2020.

[19] Wenlu Zhang, Anthony Martinez, Emily Nicole Meese, Christopher G. Lowe, Yu Yang, “Deep Convolutional Neural Networks for Shark Behavior Analysis”, IEEE Green Energy and Smart Systems Conference 2019.

[20] Brian H Curtin, Suzanne J Matthews, “Deep Learning for Inexpensive Image Classification of Wildlife on the Raspberry Pi”.

[21] Manjunath Jogin, Mohana, Meghana RK, Apoorva S, “Feature Extraction using Convolution Neural Networks (CNN) and Deep Learning”, International Conference on Recent Trends in Electronics, Information and Communication Technology ,IEEE 2018.