National College of Ireland

Identification and Classification of Wildlife from Camera-Trap Images using Machine Learning and Computer Vision

MSc Research Project
MSc in Data Analytics

Nawaz Sheikh
Student ID: x18134637

School of Computing
National College of Ireland

Supervisor: Dr. Vladimir Milosavljevic

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Nawaz Sheikh
Student ID:	x18134637
Programme:	MSc. in Data Analytics
Year:	2019
Module:	MSc Research Project
Supervisor:	Dr. Vladimir Milosavljevic
Submission Due Date:	3rd February 2020
Project Title:	Identification and Classification of Wildlife from Camera-Trap Images using Machine Learning and Computer Vision
Word Count:	6579
Page Count:	30

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	3rd February 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Identification and Classification of Wildlife from Camera-Trap Images using Machine Learning and Computer Vision

Nawaz Sheikh
x18134637

February 03, 2020

Abstract

An active research on flora and fauna is carried out since last few decades. We have focused on analysis of wildlife monitoring acquired from camera-trap networks which provides data from natural scenes. Camera-traps are placed in wildlife sanctuaries, national parks and reserves all over the world. It is the best practice to monitor wildlife from the images captured using camera-traps. Citizen science community consists of many researchers and citizen scientists who work on the data gathered from camera-traps and apply various machine learning and computer vision algorithms so that the results can be used in wildlife conservation. This project focuses on classifying animal species gathered from the Missouri Camera Traps dataset using InceptionV3, MobileNet and VGG-16 architectures of deep convolutional neural networks. Also, the weights from this project can be used in transfer learning to classify similar animal species on another dataset. Our intensive results shows that DCNN provides accuracy of 69.5% for the model of InceptionV3 for six classes of animals from the Missouri Camera Traps dataset.

Keywords: Wildlife, Camera-traps, Machine Learning, Deep convolutional neural networks, Computer Vision, Transfer Learning

1 Introduction

Plants, animals and microbes together form the ecosystem. In order to keep the ecosystems safe, we need to gain knowledge on the behaviour of animals so as to improve our skillsets. Investigating about the ecosystem can help us learn about zoology, ecology, conservation biology and behaviour of animals. A common approach to gather information on the animal species is by using camera-traps. Cameras with motion sensors assists in reducing the cost of capturing images of animals. Gomez Villa et al. (2017) mentioned that capturing images of animals from camera-traps is a huge challenge as the data generated is huge. Presently, there is no approach to identify animals automatically. Citizen science volunteers and researchers analyze thousands of images manually. Missouri Camera Traps provides dataset of 20 animal species which gives a benchmark

for the researchers to work on the data as it is publicly available for the citizen science community to work on.

1.1 Background and Motivation

To enable a computer to solve tasks without programming rigorously to solve them, machine learning techniques are applied according to Norouzzadeh et al. (2017). As per Verma and Gupta (2018), a larger amount of data is available on wildlife activity over time and space domain. Camera traps provides images of animals from their natural habitat.[1] DCNN consists of layers to extract valuable information from the images to perform operations. Level of abstraction that can be extracted can be multiple using deep learning. Output between 0 and 1 is obtained in classification which is a result of the softmax function which is the final layer. State-of-the-art is improving in the fields of machine translation, image recognition and speech recognition.

1.2 Objectives and Tasks

Below is the list that provides details of the contributions targeted in this project:

- To handle class imbalance using data augmentation.
- To classify animal species using architectures of deep convolutional neural networks.
- To use transfer learning to pre-train a ConvNet in order to identify rare classes of animals in a different dataset.
- To transform keras model to tensorflow frozen graph.
- To use cloud based platform to classify the animal species in order to boost the model performance.

1.3 Novel

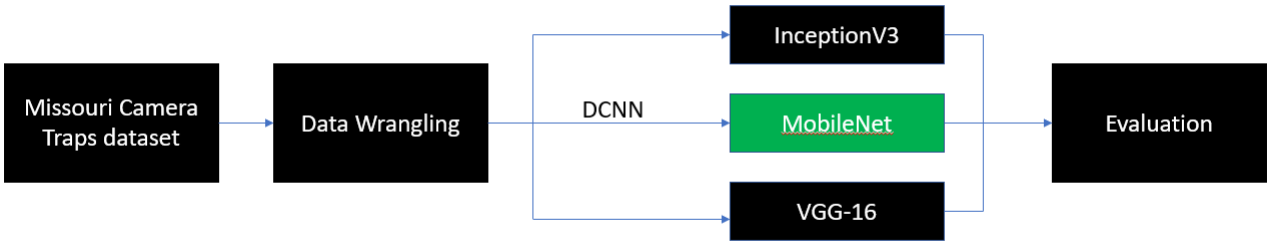

Figure 1: Novel

[1] https://arstechnica.com/science/2018/11/camera-traps-designed-for-animals-are-now-invading-human-privacy/

Missouri Camera Traps dataset is used for the classification of wild animal species. Figure 1 shows the flowchart of the project. MobileNet is a deep convolutional neural network which has never been used before for the classification of animals. Also, approach of deep learning on the Missouri Camera Traps dataset is itself novel.

1.4 Roadmap

The paper is arranges as follows: Section 2 contains literature review of research papers that are recent. Section 3 contains research methodology explained in detail. Design specification is explained in Section 4. Implementation includes machine learning and neural network techniques in Section 5 along with the details of the environment that used to implement the techniques. Section 6 includes evaluation of the metrics and discussion. Section 7 is discussion. In Section 8, ethical implication is explained. Section 9 contains conclusion and future work. Section 10 is acknowledgements.

2 Literature Review

An essential part of the project is related work on the topic. Literature review helps in gaining the knowledge about the algorithms, methodology and techniques.

2.1 Identifying animal species based on deep convolutional neural networks

Images extracted from the Serengeti National Park, Tanzania can be obtained automatically using deep learning as stated by Norouzzadeh et al. (2017). 3.2 million images of 48 species are available in the Snapshot Serengeti dataset captured from camera-traps. 96.6% accuracy was obtained in identifying the animals. Rare classes can have better results after data augmentation. Patterns of behaviour from the rare species can be obtained which can be used for species conservation. 99.3% of the human labor hours dedicated for labelling of images acquired from camera-traps can be saved using deep learning. Human volunteers can dedicate their time working on sequences that are challenging with dynamic and highly cluttered scenes. Labelling of images can be done using active learning only if network is not much confident about the same. Deep neural networks can reduce the cost for data extraction and the information acquired from the knowledge can be utilized for the conservation of wildlife.

Automatic identification of wildlife from the Wildlife Spotter project was done by Nguyen et al. (2018). This dataset is a single-labeled dataset labelled by citizen scientists to train with deep convolutional neural networks in order to identify wildlife automatically. The experiments obtained 96.6% accuracy to detect images of the animals and the three most common species of wild animals (rat, bandicot and bird) achieved the accuracy of 90.4% and the images are captured in the South central region of Victoria in Australia. Different experiments allowed the system to deal with imbalanced and balanced images. Resnet-50, VGG-16 and Lite Alexnet are the three architectures of CNN that were used. As compared to all the architectures, 90.3% is the highest accuracy achieved by Resnet-50. Downscaling of all the images to 224 x 224 pixels for training of the data was done and used as the input as the original dataset had high resolution images of 2048 x 1536 pixels and 1920 x 1080. Data quality of the images were helped in

enhancement of the images using augmentation techniques. Frontend was done by Keras and TensorFlow was used for the backend.

Chen et al. (2014) performed classification of wild animals using deep convolutional neural networks using images obtained from the camera-trap image data. 14,436 images were used for training and 9,530 images were used for testing obtained from the 20 common animal species in North America. In order to automate the computer vision based wildlife recognition, an attempt was made using DCNN. BOW based algorithm was outperformed by DCNN for the recognition of the species. Benchmarking and evaluation helped in the performance of the experiment. Random sampling for testing and training of the images was performed which had infrared, gray and coloured images with resolutions ranging from 1024 x 768 to 320 x 240. Out of the 20 species mentioned, each image contained one type of animal. Linear SVM was used as a classifier and histograms were used to represent the images. An accuracy of 33.5% and 38.3% were achieved by BOW and DCNN respectively.

Verma and Gupta (2018) mentioned about the low detection and high false discovery rates obtained from the detection of wild animals from camera-trap images. Verification is done on the basis of patch having an animal or the patch being empty without an animal. 91.4% is the accuracy obtained using deep convolutional neural networks. To formulate candidate animal regions in order to use Iterative Embedded Graph Cut (IEGC) technique, spatial context and motion of animas are used. First model is the animal-background verification model which differentiates the background patches from the animals. Second model is the classification of the features obtained from positive and negative set of KNN, SVMs and ensemble algorithms. Images of 20 different animal species was used and the experiment is performed at an average of 100 images per class. Daytime and nighttime gives efficient results for the experiment.

Kalita and Biswas (2019) mentioned that excellent performance is achieved by convolutional neural networks when working on visual tasks such as classification of two-dimensional images. Deep convolutional neural networks are used to classify hyperspectral images into spectral domain. Input layer, output layer, convolutional layer, max pooling layer and full connection layer are the five laters used with weights. To distinguish the layers between each other, each layer is deployed on the signature of the spectral. Indian Pines dataset is used for image segmentation. Firstly, 90.1% accuracy is achieved for the CNN algorithm. Secondly, LeNet-5, two-layer NN and DNN obtained the accuracy of 88.2%, 86.4% and 87.9% respectively.

1.2 million images from the ImageNet LSVRC-2010 contest were used for training and classification was done using deep convolutional neural network by Hansson (2002). Top-1 and top-5 test data obtained the error rates as 37.5% and 17% respectively. Three fully connected layers and max-pooling layers altogether formed the neural network. In order to train the data quickly, non-saturating neurons were used. Overfitting was stopped using a regulization method known as dropout. In ILSVRC-2012, the top-5 error rate achived was 15.3% as compared to the second place entry who achieved 26.2%. Data Augmentation helped in handling overfitting too. To compare, CNN, SIFT + FVs and Sparse coding were used.

2.2 Wild animals monitoring using Very Deep Convolutional Networks

26 common species from the Snapshot Serengeti dataset were studied by Gomez Villa et al. (2017). 88.9% top-1 accuracy and 98.1% top-5 accuracy was achieved from the labels obtained from state of the art of DCNN. Segmentation algorithm is proposed to remove the empty images which are gathered from the camera-traps. Data Augmentation can help solve the class imbalance issue. Auto-occlusion, complex poses of the animals and poor illumination makes it difficult to automate the camera-trap methods. To feed in more images and improve the performance of the model, cropped images from the ImageNet dataset can be utilized. Panama dataset helped in providing colour and infrared images. Stacking pooling layers from the first layer to the last layer to detect fur and edge details of the animals are a part of Convolutional Neural Networks (ConvNets). If deeper architectures are fine-tuned, then they can surely outperform the state of the art.

Chung et al. (2018) checked the CNN depth from the dataset obtained from Imagenet Challenge 2014 to work on the infrastructure of large-scale image recognition. The depth of the networks were pushed from 3 x 3 convolutional layers to 16-19 layers. Visual representation is done in computer vision once research done from the ConvNet models are gathered from the research. The images are cropped to a size of 224 x 224. ILSVRC-2012 is the dataset on which ConvNet architectures are applied. The range of the ConvNet architectures starts from A, A-LRN, B, C, D till E. Errors are based on top-1 and top-5 and are compared with VGG.

As per Schneider et al. (2018), data is automated using deep learning techniques in computer vision for the dataset obtained from camera trap images. In order to monitor the ecosystem of the animal population, insight about the environment is collected before it becomes intrusive. Faster R-CNN achieved 93% accuracy whereas YOLO achieved 76.7% accuracy. Six variations of ResNet, GoogleNet, AlexNet and VGG are the nine architectures that were trained. Images that were considered were labelled with bounding box coordinates and the number of images were 946.

Yousif et al. (2018) introduced Minimum Feature Difference (MFD) to model the background changes of the images taken from the sequences of the camera-traps that generated foreground proposals of the objects. Development of region proposals helped in reducing the false alarms. Classification is done using DCNN to separate wild animals, humans and background patches. Infrared images are taken during the night time. Test sequence is sent into patches by the system. Cross-frames are verified on the patches on the basis of performance of background modelling. Furthermore, the labelled sequence is developed by the proposals classification. Local Binary Pattern (LBP), Gray Level Co-Occurence Matrix (GLCM) and Histogram of Oriented Gradient (HOG) are the features of Block-wise Background Modeling. Foreground regios is classified using the Alexnet architecture. Accuracy decreases from 95.6% to 93.4% as the size of the image decreases from 256 x 256 to 96 x 96. To maintain the high accuracy, time of the classification was reduced by 14 times.

Gutierrez-Galan et al. (2018) proposed the animal monitoring system, classification and behaviour recognition. A smart collar device along with a sensor network that works on wireless technology is used. Multi-Layer Perceptron-based feed-forward neural network is embedded for classification. 81% accuracy is achieved. Animal information is collected using the wireless sensor network. Real-time classification is done using the 3-axis acceleration sensor and ANN is the method used with 3 output classes. Donana

National Park provided the dataset generated from the MINERVA research project. The segregation of the Fast Artificial Neural Network (FANN) on the animal data is done with 70% training, 15% validation and 15% testing. NN architectures are simulated using the FANN library.

Miao et al. (2019) classified 20 animal species from Africa achieving an accuracy of 87.5%. The dataset contained 111,467 images of wild animals from Gorongosa National Park, Mozambique. VGG-16 and ResNet-50 are the CNN architectures used. To extract the features in the last convolutional layer, gradientweighted class-activation-mapping (Grad-CAM) procedure is used.

2.3 Classification based on SVM and camera-trap technologies

Matuska et al. (2014) proposed object recognition that are based on hybrid local descriptors. Speeded Up Robust Features (SURF) and Scale-invariant feature transform (SIFT) blends together to be used for the object recognition. Classification of animals from the Slovak country includes deer, fox, brown bear, wold and wild boar. Firstly, feature extraction has been completed on the training data. Secondly, Naive Bayes and SVM are used to classify the images. Also, Support Vector MAchine (SVM) and Bags of Keypoints (BOW) are utilized for classification. BruteForce matcher, SISURF detector and OpponentSIFT descriptor provided the highest accuracy of 86%. SUSIFT detector provided poor results with 50% accuracy.

C. Miguel et al. (2019) stated that conservation biologists studied snow leopards using camera trapping techniques. Snow leopards are separated from the empty images using the technique. In order to form motion templates, binary morphology, thresholding and Robust Principal Component Analysis are used. Cascade Object Detector is used to find spots from the images obtained from the the camera traps. SVM Classifier has input from density of the motion template and overlapping spots. 93.7% is the classification accuracy. To conduct the experiment, conservation bilogists manually sort the images. Advanced data analysis can be done rather than investing too much time on manual sorting of the images. Snow leopards are endangered species and their study will help the conservation biologists to learn about their patterns and behaviour in order to maintain the snow leopard's population. 70% dataset is used for training and 30% dataset is used for testing. Motion extraction, spot detection and support vector machine classifier are the three steps used in conducting the experiments.

Swanson et al. (2015) worked on the dataset gathered from 225 camera traps of Serengeti National Park that accumulated approximately 1.2 million images since 2013. 28000 registered users helped in the classification. A simple algorithm is used to aggegrate the classification done individually on the final dataset. 97% is the accuracy achieved. Citizen scientists used computer vision to classify the species of the Serengeti National Park.

Extraction of features like cell-structured LBP (cLBP) and dense SIFT descriptor are used on the images that are cropped and improved sparse coding spatial pyramid matching (ScSPM) is applied on the images by Yu et al. (2013). Furthermore, classification is done using linear support vector machine algorithm. Global features are generated using max pooling and weighted sparse coding along with multi-scale pyramid kernel. 7000 camera-trap images consists of 18 species from different cities and the accuracy achieved is 82%. Tropical rainforest and temperate forest provided the images of the animal species. Therefore, variataion in conditions can be observed. Conversion of the images into

gray scale helped to apply cLBP and SIFT with the size of K = 1024. 74.5% accuracy is obtained by cLBP, 78.9% accuracy is obtained by 78.9% and an ensemble model provided an accuracy of 82%.

Yousif et al. (2017) performed deep learning classification and joint background modelling on the camera-trap images to detect humans and animals. Development of background modelling is performed first. Later on, regions for the foreground objects are generated using subtraction scheme. Secondly, cross-frame image patch verificaton to minimize proposals of the objects in the foreground is developed. Division of three categoris such as humans, animals and background objects is performed using deep convolutional neural networks. The accuracy obtained was 82% using linear SVM. On background subtraction, minimum feature distance background (MFD) achieved an accuracy of 83.7%.

Rey et al. (2017) proposed about the detection of mammals in semi-arid Savanna. Counting of large herbivores, marine mammals and birds in various environments is done using unmanned aerial vehicles (UAVs). Machine learning techniques are used to detect the animals wherein the volunteers helps in annotating sub-decimeter coloured images. Detecting of false alarms helps in achieving high recall rate which is assisted by the human operators. UAV wings are fixed on the RGB cameras in order to process the data. As training is completed using annotations, the paradigm is based on supervised learning. Ensemble of Exemplar Support Vector Machines (EESVM) is the adopted algorithm. Hyperparametrization is used on the Kuzikus dataset.

2.4 Removal of empty images with no animals

Computer vision is used to develop the algorithms to classify and identify moving objects as per Yousif et al. (2019). Segregation of animals from empty frames and humans is done. To avoid false triggers, it is a necessity to remove empty images. Background is subtracted using object segmentation and deep learning to provide better accuracy. C++ is used to develop the command prompt and the program is written in Matlab. 99.5% accuracy is obtained for empty images versus object classification from the Serengeti dataset. The task is time consuming as 98% of the vegetation is moving. Images are taken from infrared cameras as well and the model runs on both, infrared and color photos. Confusion matrix is plotted for animals, humans and empty images. Evaluation is performed on the sequence- level as per the deployment of the camera traps.

2.5 Transfer Learning

Pan and Yang (2009) mentioned that it is a difficult task to train the data and then apply the results on a future data because they maybe in different distribution. Both the domain distribution can be different after the classification. To avoid data labelling efforts which are expensive, knowledge transfer can be used for the automation of data labelling. Regression, clustering and classification problems can be resolved using transfer learning. To avoid negative transfer learning, a thorough study of transferability needs to be done between the source and the target domains.

3 Research Methodology

Discovering knowledge from the data that is collected is the task of Knowledge discovery in databases (KDD). Data from the repository is prepared, cleaned, knowledge is incorporated on the dataset and finally observed results are evaluated.

KDD includes multidisciplinary activities.[2] Artificial intelligence supports observations and experiments by the discovery of empirical laws in KDD. KDD provides knowledge in the form of patterns. KDD has the following steps involved:

- Customer's point of view is taken into consideration when setting goals in KDD.
- Application of the domain is understood.
- Selection of data takes place in order to perform discovery.
- Class imbalance issue in image classification is handled by data preprocessing according to the requirements.
- To match the goals of KDD, data mining methods are applied to get hidden patterns.
- Models and algorithms are decided to discover hidden patterns.
- Extracting knowledge from the mined patterns.
- Allowance of the knowledge gained to be incorporated in a different system.

3.1 Data Repository

Missouri Camera Traps contains approximately 25,000 images of 20 different species. The images can be used for scientific research in object detection and classification. Citizen scientists across the globe labels the images manually so that it can be used by researchers. Metadata of the Missouri Camera Traps dataset is available in .json format.[3]

3.2 Selected Data

Out of the 20 species, 6 species are selected for further classification. Images of the minority class are focused more on to handle class imbalance. 6 selected classes include species like Collared Peccary, European Hare, Ocelot, Red Deer, Red Squirrel and White-nosed Coati.

3.3 Pre-process Data

Data augmentation handles the class imbalance. Images of the minority class are augmented in order to balance all the classes. The dataset is divided into three categories: train, validation and test. Transfer learning can be used on the test dataset after training and validation on similar animal species. The knowledge obtained can be used on different dataset with similar sets of animals.

[2]https://www.techopedia.com/definition/25827/knowledge-discovery-in-databases-kdd
[3]http://lila.science/datasets/missouricameratraps

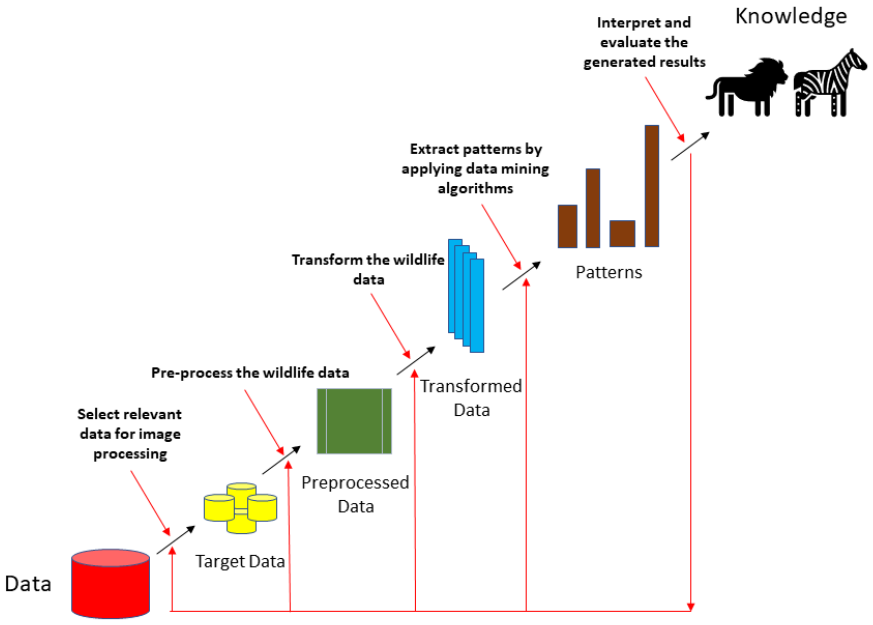

Figure 2: KDD methodology

3.4 Transform Data

Data transformation includes detection of layers on the images. Transfer Learning is used by using the weights from the ImageNet. To match and limit the number of classes to be observed, top layers are removed. To speed up the training process, transfer learning is used.

3.5 Patterns

InceptionV3, MobileNet and VGG-16 helps in boosting the performance of the model. Classification of the animals is done using deep convolutional neural networks. Precision-Recall curve, ROC curve and confusion matrix helps in evaluating the metrics.

3.6 Knowledge

Same weights can be taken and used in another dataset to detect similar species using transfer learning. This shall help the citizen science community to focus on more complex projects as the animals can be predicted on another dataset. Human labors of the citizen community will be reduced. The results obtained can be utilized by the ecologists to boost and enhance the wildlife ecosystem. Figure 2 shows the steps starting from data extraction to knowledge gained from the experiment.

4 Design Specification

A three-tier design architecture is proposed which starts with the database layer and the results are processed to the application layer and the output obtained will be presented using the presentation layer.

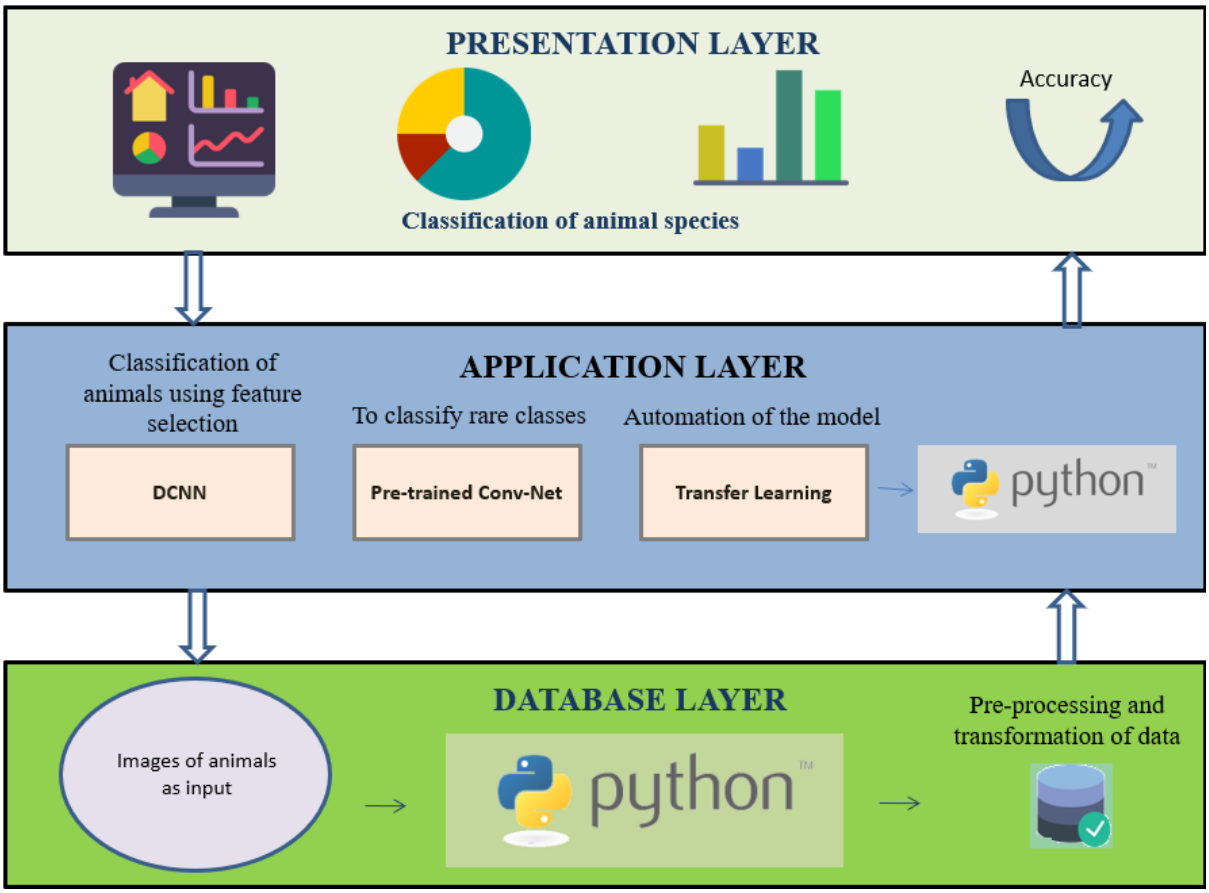

Figure 3: Design Architecture

Figure 3 shows the three layers of the design architecture. In the first layer which is the database layer, Missouri Camera Traps dataset is obtained and python programming on Google Colab is used to clean the data. Data attributes are understood using Exploratory Data Analysis.

In the second layer which is the application layer, architectures of deep convolutional neural networks are applied to obtain the results using Precision, Recall, F-1 Score, Top-5 accuracy, accuracy and loss function.

In the last layer which is the presentation layer, visualization using the ROC curve and precision-recall curve is provided for a better understanding to the clients.

Right from data gathering to data transformation and then finally achieving results, the three layers provides the work flow that was done in the project.

5 Implementation

Implementation of this project is bifurcated on addressing the deliverables mentioned in section 1. Also, transfer learning can be applied to smaller datasets with similar classes where the learning from the original dataset can be used to get results from the new dataset without spending the training time for the new dataset. Different architectures of convolutional neural networks can be helpful to increase the accuracy of the model where results from each architecture can be compared.

Researchers in the past assisted in selecting appropriate methods for feature selection, preprocessing of data, predicting the correct output and try to beat the state-of-the-art as algorithms have been successful in classifying the images. The next section is evaluation and the metrics observed are based on the implementation discussed i this section.

5.1 Data Wrangling

The process of cleaning, structuring and getting raw data into a format which a user desires is known as data wrangling.[4] Unstructured data needs to be cleaned and organized in order to get better results after applying the algorithms.

5.1.1 Clean Images

Images not having .JPG extensions are removed. Also, images that cannot be read by cv2 module are removed as well.

5.1.2 Distribution of Selected Classes

PyGal is imported to create a wrapper to render the chart inline when data is passed through the charting function. Figure 4 refers to the original distribution of the classes.

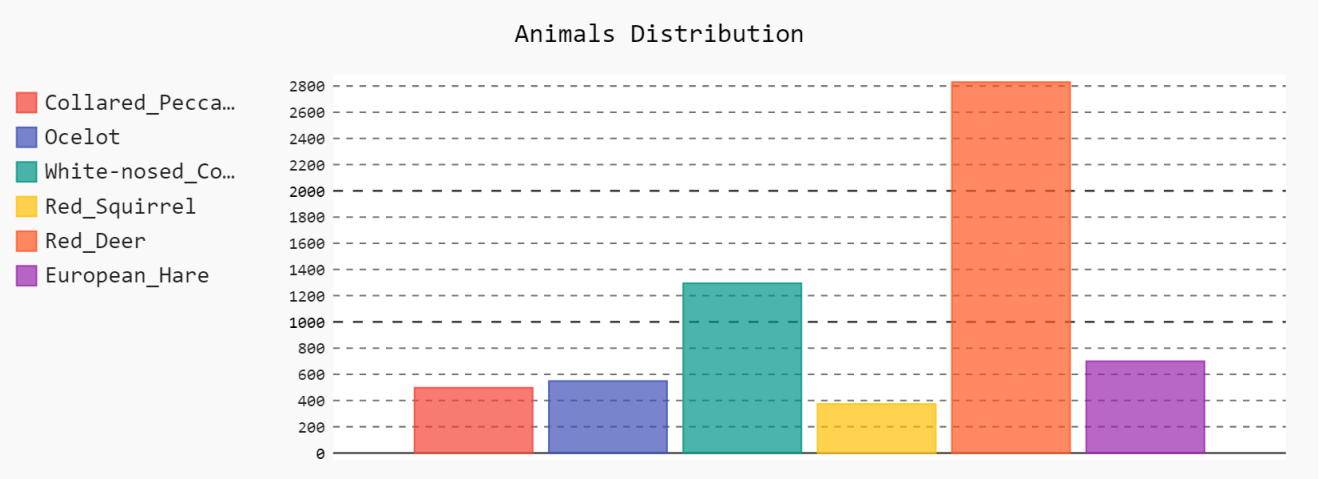

Figure 4: Animals Class Distribution

[4]https://www.trifacta.com/data-wrangling/

5.1.3 Confirm the folder structure

This is to summarize and confirm the progress and also to verify the folder structure. Figure 5 shows the folder structure before training, testing and validating the dataset.

```
Six_Classes/
    Collared_Peccary/
        SEQ88200_IMG_0003.JPG
        SEQ88200_IMG_0007.JPG
        ...
    Ocelot/
        SEQ75294_IMG_0005.JPG
        SEQ75294_IMG_0003.JPG
        ...
    White-nosed_Coati/
        SEQ84536_IMG_0001.JPG
        SEQ84536_IMG_0008.JPG
        ...
    Red_Squirrel/
        SEQ75972_IMG_0002.JPG
        SEQ76082_IMG_0001.JPG
        ...
    Red_Deer/
        SEQ80452_IMG_0016.JPG
        SEQ80452_IMG_0019.JPG
        ...
    European_Hare/
        SEQ75140_IMG_0004.JPG
        SEQ75140_IMG_0001.JPG
        ...
```

Figure 5: Folder Structure

5.1.4 Creating train, test and validation folders

Training, test and validation folders are created for data ingestion and the ratio is 0.7, 0.2 and 0.1 for this purpose. Figure 6 indicates the movement of images into training, validation and testing folders for the six classes.

```
Moving 348 train images to category folder Collared_Peccary
Moving 49 validation images to category folder Collared_Peccary
Moving 101 test images to category folder Collared_Peccary
Moving 384 train images to category folder Ocelot
Moving 54 validation images to category folder Ocelot
Moving 111 test images to category folder Ocelot
Moving 906 train images to category folder White-nosed_Coati
Moving 129 validation images to category folder White-nosed_Coati
Moving 260 test images to category folder White-nosed_Coati
Moving 261 train images to category folder Red_Squirrel
Moving 37 validation images to category folder Red_Squirrel
Moving 76 test images to category folder Red_Squirrel
Moving 1980 train images to category folder Red_Deer
Moving 283 validation images to category folder Red_Deer
Moving 567 test images to category folder Red_Deer
Moving 489 train images to category folder European_Hare
Moving 70 validation images to category folder European_Hare
Moving 141 test images to category folder European_Hare
Done.
```

Figure 6: Moving images into their target folders

5.1.5 Data Augmentation

While looking at our distribution above we saw that certain classes were significantly lower than others. To help mitigate that issue we're going to augment some of our data set so that we have a dataset that is more closely distributed. Firstly, an image is augmented using a certain threshold of modification. Secondly, these random augmentations are applied to the data.

Checking the distribution for all the classes gave us an indication that certain classes are significantly lower than others. To solve this issue, images for the minority classes are augmented so that the distribution becomes close and there is no bias for the majority class.

5.1.6 Resizing of images

Resizing of images is done depending on the topology with the expected image format. For InceptionV3, the size is 299 x 299. VGG and MobileNet has the image size 224 x 224.

5.1.7 Distribution of classes after augmentation

Once the data augmentation is completed for the minority classes, each class has the similar number of images. This is only done for the training dataset in order to not create a bias on the validation and testing dataset. Figure 7 refers to the class distribution of the animals after data augmentation.

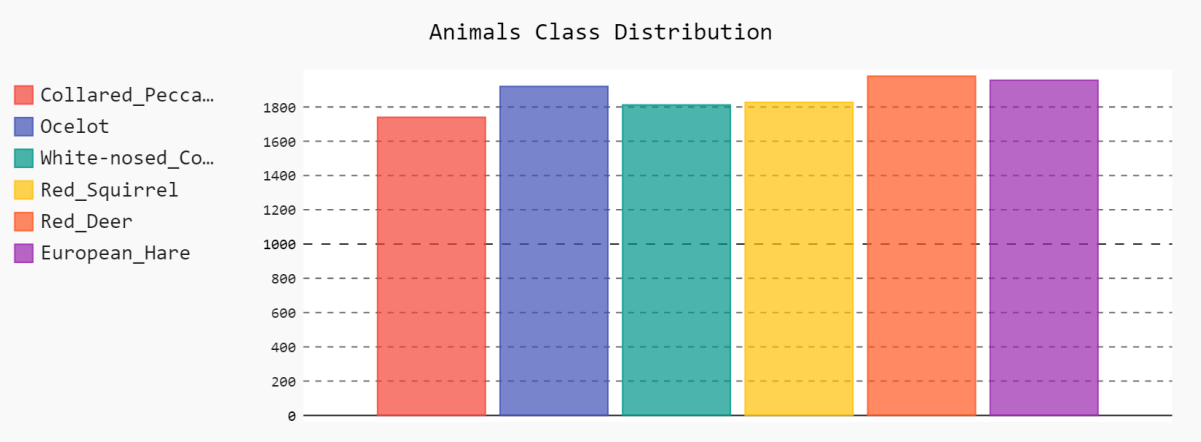

Figure 7: Animals Class Distribution after Augmentation

5.2 Convolutional Neural Networks (ConvNet)

Convolutional Neural Networks are made up of pooling layers and stacked convolutional layers. A non-linear function is generated using feature maps. Pooling layer helps in extracting the scale invariant capacity. First layer has low features like orientation and edges. High level features such as fur and wrinkles are part of the last layer which is observed in animals. InceptionV3, VGG-16 and MobileNet are the architectures of Convolutional Neural Networks. In order to fit the topology of a ConvNet, images are resized.

5.3 Model optimization

5.3.1 Creating data generators for the dataset

Keras ImageDataGenerator class is used to ingest the data for training. This allows to easily read the directories which has each category structured in its respective folder. Train, test and validation sets are created earlier during the exploratory data analysis. For each of the folder classes, generator is utilized.

Height and Weight are set as per the requirement of the topology and it is instantiated during this phase so that it can be used throughout the implementation. Resizing of images is also taken care of by the generator so that before feeding the images to the training, testing and validation, it will make sure that it works efficiently. Batch size of images is set to 32 as it is used as the subset of the training data which is utilized in one iteration.

5.3.2 Hyperparameter selection

1. **Transfer Learning**

   According to Yosinski et al. (2014), transfer of valuable knowledge from one dataset to another is transfer learning. Fine tune process and black box feature extractor are parts of transfer learning. Weights from a different network is loaded on a new network.

   In this project, there is a difference in number of classes from ImageNet and hence the top layers are removed and then the network is re-instantiated in order to match the number of classes. Transfer Learning helps in speeding up the training process by using things such as edge detection learned from the previous dataset and the new network can be finely tuned to the dataset.

2. **Initialize Training Top Layers**

   Pre-defined Inception V3 network which is provided by Keras is loaded. Weights from ImageNet is utilized for transfer learning because it speeds up the training process. Top layers are excluded as 1001 classes are not to be predicted and it is modified to fit the network of our dataset.

   Base model is taken into consideration and GlobalAveragePooling2D layer is added and passed on to the output of the base model. With softmax activation of a final Dense Layer, we predicted the number of classes in the dataset. Train_flow generator class indicies number is utilized to make sure that the model is versatile and it will automatically correct the number of classes in the dataset and use them.

   Initial layers of the base model is iterated and they are disabled to change the layer.trainable variable for training which is set to False. It means that we have trained the new layers which is added explicitly for our dataset.

   The model is compiled and optimized is added. We have used Adam Optimizer with 0.001 learning rate. Since we are using a multi-class classification problem, loss of categorical crossentropy is considered.

   Total number of nodes needed for the training and checked if the last layers are added perfectly to the network.

5.3.3 Training Callbacks

3 epochs are used and callbacks are utilized to make sure that the model does not overfit. For VGG-16, each epoch took approximately 2.30 hours. For InceptionV3, each epoch took approximately 2.18 hours. For MobileNet, each epoch took approximately 2.15 hours.

- **ModelCheckpoint** - Only the least value of the loss function is saved after every epoch by comparing the value from the previous epoch. This is the best way to save the disk size as the best minimum value of the loss function is overwritten.
- **TensorBoard** - This allows the tf.events file to write out and it can be viewed on the TensorBoard. The log file can be accessed on the TensorBoard to see how the training goes.
- **EarlyStopping** - The training will be stopped if there is no change in the value obtained from the last 5 epochs.
- **CSVLogger** - In case the tensorboard is not to be used, this will append the metrics to the CSV file.

5.3.4 Transform Keras Model to Tensorflow Frozen Graph

Keras saves the model with the .hdf5 file format. The file is converted into .pb file if the model is needed to be used with TensorFlow.[5]

6 Evaluation

To test the implementation mentioned in section 5 and to obtain the results of the objectives discussed in section 1, many experiments are conducted. VGG-16, InceptionV3 and MobileNet are the architectures on which the evaluation results are based. F1-score, Precision, Recall, Accuracy and Confusion Matrix are the key features in evaluation.

The metrics mentioned above are used to classify animal species.

6.1 Confusion Matrix

A graph with true label vs predicted label can be plotted and color code can be achieved accordingly. When the confusion matrix is ideal, there is a diagonal line from top left to the bottom right with no other color. Ideal confusion matrix means that each true value matches the predicted value. Normally, every class leans towards one or two different classes which may have similarities with the true class.

[5] https://www.dlology.com/blog/how-to-convert-trained-keras-model-to-tensorflow-and-make-prediction/

1. VGG-16

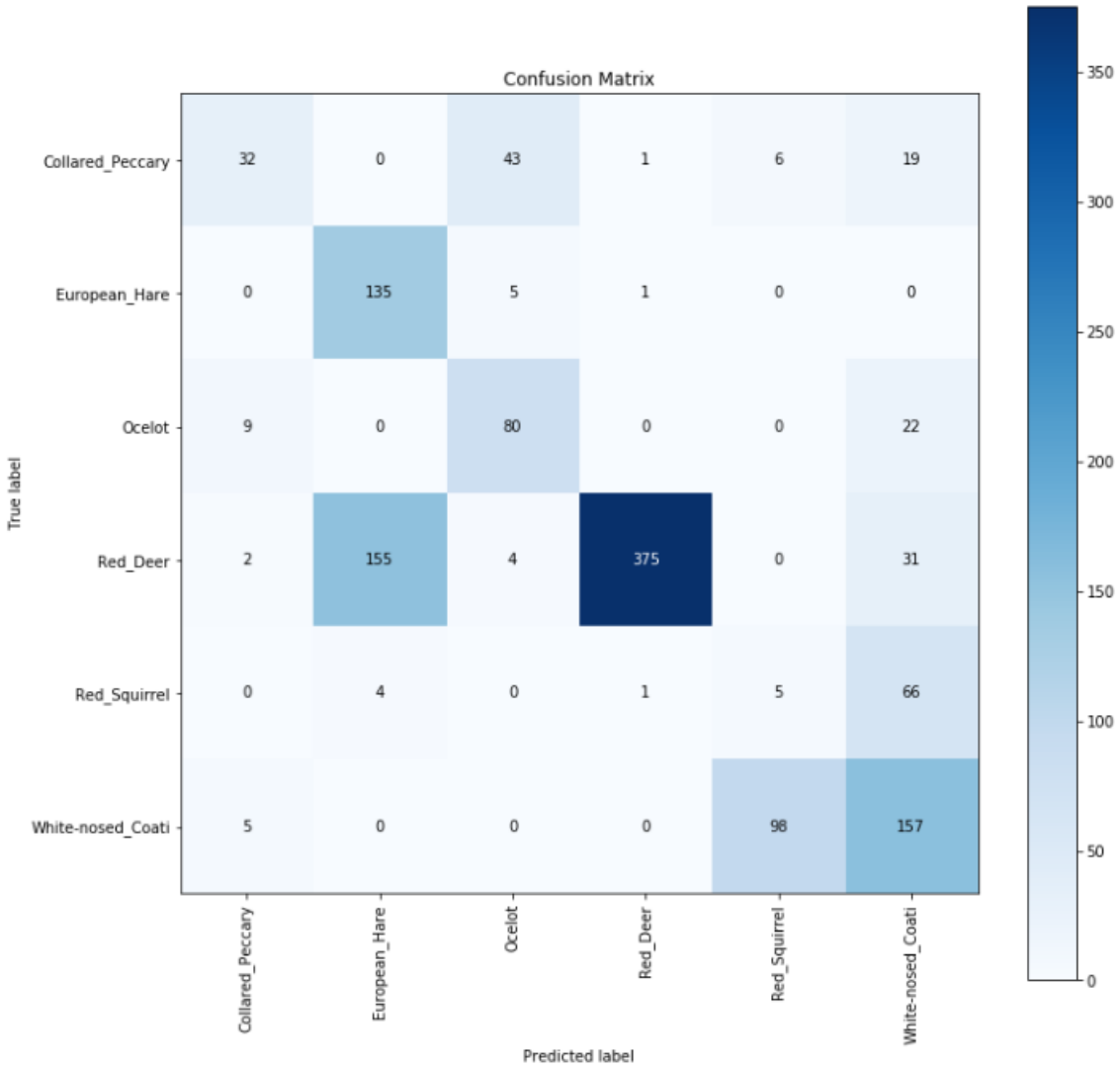

Figure 8: Confusion Matrix - VGG16

Figure 8 refers to the confusion matrix for VGG-16. As per the results obtained for VGG-16, predicted label for the classes Collared Peccary and Red Squirrel differs from the true label of the classes. For all the classes, majority values for the predicted label and true label are same. Red Deer is the majority class and VGG-16 has the lowest value for true label vs predicted label.

2. InceptionV3

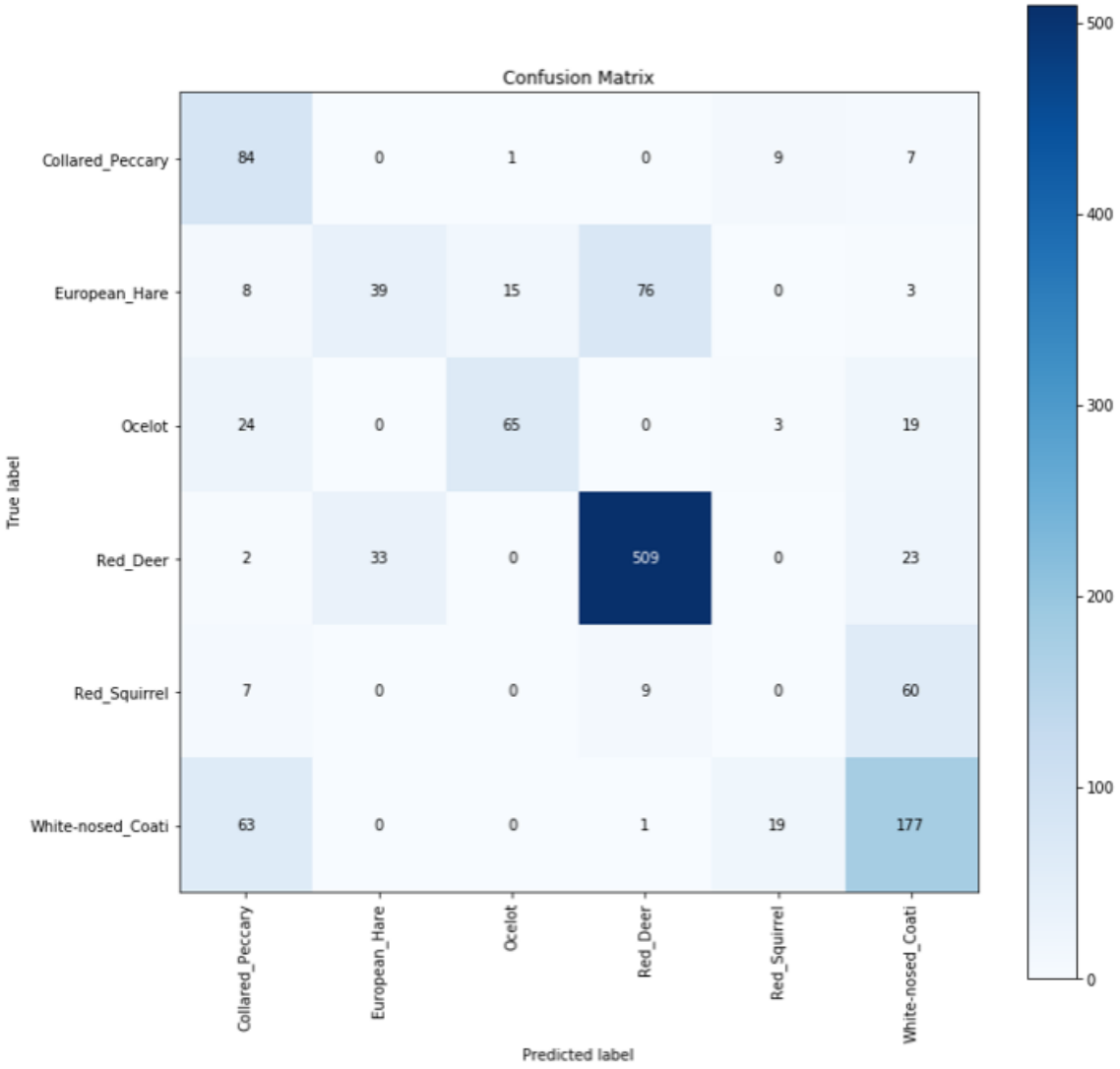

Figure 9: Confusion Matrix - InceptionV3

Figure 9 refers to the confusion matrix for InceptionV3. As per the results obtained for InceptionV3, predicted label for the classes European Hare and Red Squirrel differs from the true label of the classes. For all the other classes, majority values are same for the predicted and true labels. The number of true vs predicted value for InceptionV3 for the class Red Deer is better than both the architectures VGG-16 and MobileNet. The second highest true label vs predicted label is for the class White-nosed Coati.

3. MobileNet

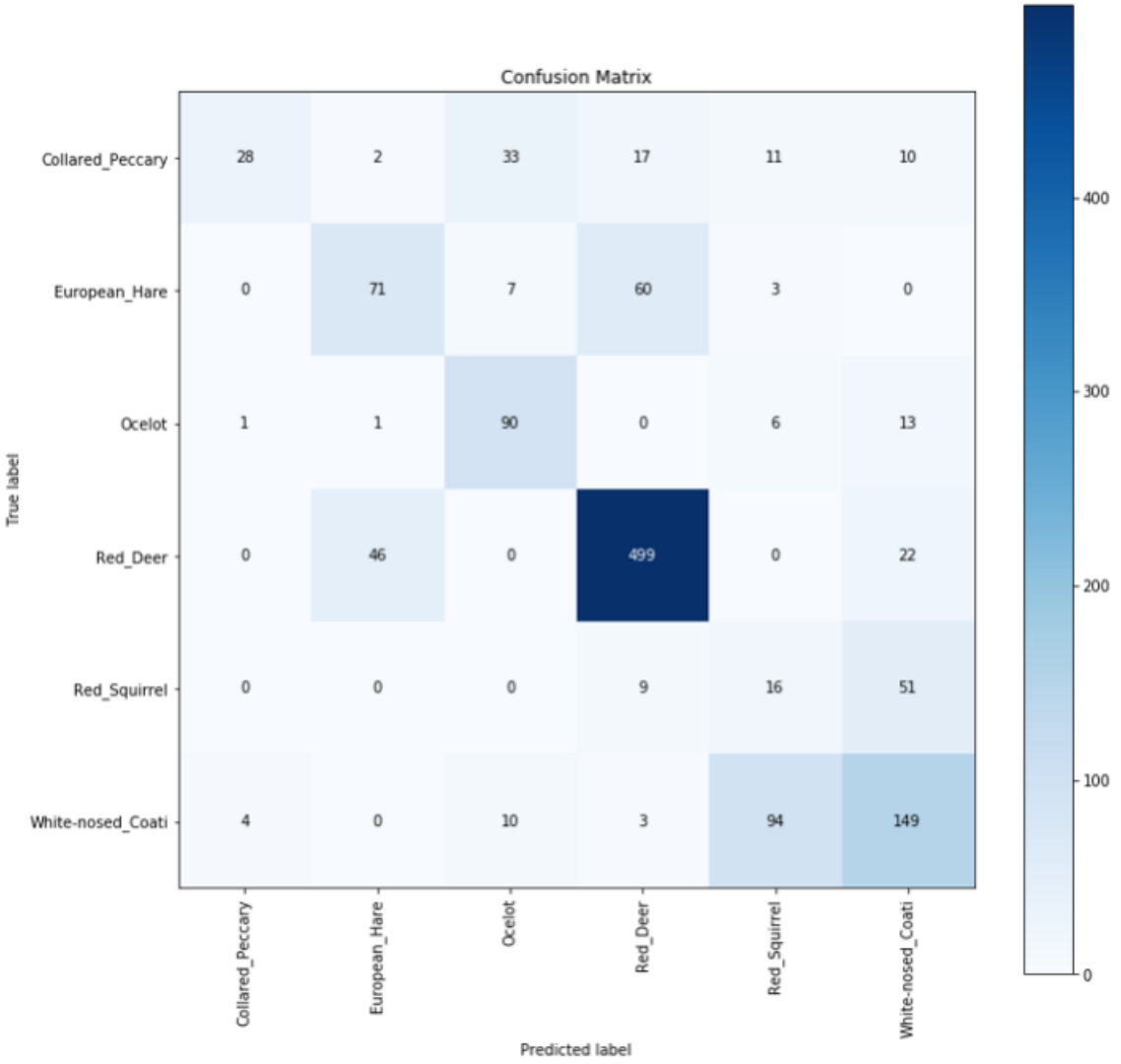

Figure 10: Confusion Matrix - MobileNet

Figure 10 refers to the confusion matrix for MobileNet. As per the results obtained for MobileNet, predicted label for the classes European Hare and Red Squirrel differs from the true label of the classes. For all the classes, majority values for the predicted label and true label are same. True label for Red Squirrel shows the maximum values for predicted label for White-nosed Coati.

6.2 Accuracy, Loss and Top-5

Achieving good accuracy is an important criteria in image classification. While identifying animals from the camera-trap images, accuracy can differ during the day and night.The ensemble classifiers can obtain different top-1 and top-5 accuracy for the species during the day and night. Categorical crossentropy has been evaluated for loss.[6] Top-5 will

[6]https://subscription.packtpub.com/book/big$_d$ata$_a$nd$_b$usiness$_i$ntelligence/9781789132212/3/ch03lvl1sec30/understa
categorical − cross − entropy − loss

evaluate the results so as to check if the observed output is in the top-5 confidence.

$$\text{Accuracy } = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Table 1: Accuracy, Loss and Top-5 confidence

Model	**Accuracy (%)**	**Loss**	**Top-5 confidence (%)**
VGG-16	62.42	1.86	99.92
InceptionV3	69.58	1.61	99.76
MobileNet	67.91	1.35	99.76

Table 1 shows the results for accuracy, loss and top-5 confidence. InceptionV3 achieved the highest accuracy of 69.58% whereas Top-5 confidence for VGG-16 is 99.92% which is the highest. Minimum values for predicted probability for the classes in MobileNet diverges from the actual value as compared to both the models.

6.3 Classification report

	precision	recall	f1-score	support
Collared_Peccary	0.67	0.32	0.43	101
European_Hare	0.46	0.96	0.62	141
Ocelot	0.61	0.72	0.66	111
Red_Deer	0.99	0.66	0.79	567
Red_Squirrel	0.05	0.07	0.05	76
White-nosed_Coati	0.53	0.60	0.57	260
accuracy			0.62	1256
macro avg	0.55	0.55	0.52	1256
weighted avg	0.72	0.62	0.64	1256

Figure 11: Classification report for VGG-16

	precision	recall	f1-score	support
Collared_Peccary	0.45	0.83	0.58	101
European_Hare	0.54	0.28	0.37	141
Ocelot	0.80	0.59	0.68	111
Red_Deer	0.86	0.90	0.88	567
Red_Squirrel	0.00	0.00	0.00	76
White-nosed_Coati	0.61	0.68	0.64	260
accuracy			0.70	1256
macro avg	0.54	0.55	0.52	1256
weighted avg	0.68	0.70	0.68	1256

Figure 12: Classification report for InceptionV3

	precision	recall	f1-score	support
Collared_Peccary	0.85	0.28	0.42	101
European_Hare	0.59	0.50	0.54	141
Ocelot	0.64	0.81	0.72	111
Red_Deer	0.85	0.88	0.86	567
Red_Squirrel	0.12	0.21	0.16	76
White-nosed_Coati	0.61	0.57	0.59	260
accuracy			0.68	1256
macro avg	0.61	0.54	0.55	1256
weighted avg	0.71	0.68	0.68	1256

Figure 13: Classification report for MobileNet

Figure 11, Figure 12 and Figure 13 refers to the classification reports for VGG-16, InceptionV3 and MobileNet respectively. The report includes precision, recall and F1 score for individual classes as well as for all the classes together. Support is the number of observations.

1. Precision

   The ratio in equation (2) is the precision where tp is the number of true positives and fp is the number of false positives. It is the ability of a classifier to not label a negative sample as positive. 1 is the best value and 0 is the worst value.[7]

   $$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

   0.72 is the highest precision observed for VGG-16 whereas InceptionV3 and MobileNet has the precision score of 0.68 and 0.71 respectively.

2. Recall

   The ratio in equation (3) is the recall where tp is the number of true positives and fn is the number of false negatives. It is the ability of the classifier to find all positive samples. Similar to precision, 1 is the best value and 0 is the worst value.

   $$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

   Highest recall of 0.70 is achieved for InceptionV3 whereas, 0.62 and 0.68 are the values obtained for VGG-16 and MobileNet respectively.

3. F1 Score

   Weighted average of precision and recall is the F1 score. 1 is the best value and 0 is the worst value.

   $$F1\text{-score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

   0.7 is the highest F1 score observed for InceptionV3. 0.62 and 0.68 are the values obtained for VGG-16 and MobileNet respectively.

[7] https://www.scikit-yb.org/en/latest/api/classifier/classification$_r eport.html$

6.4 Precision-Recall Curve

```
0 : Collared_Peccary
1 : European_Hare
2 : Ocelot
3 : Red_Deer
4 : Red_Squirrel
5 : White-nosed_Coati
```

Figure 14: Class labels

Figure 14 refers to the numeric value for the animal classes which helps in reading the precision-recall and ROC curves.

1. VGG-16

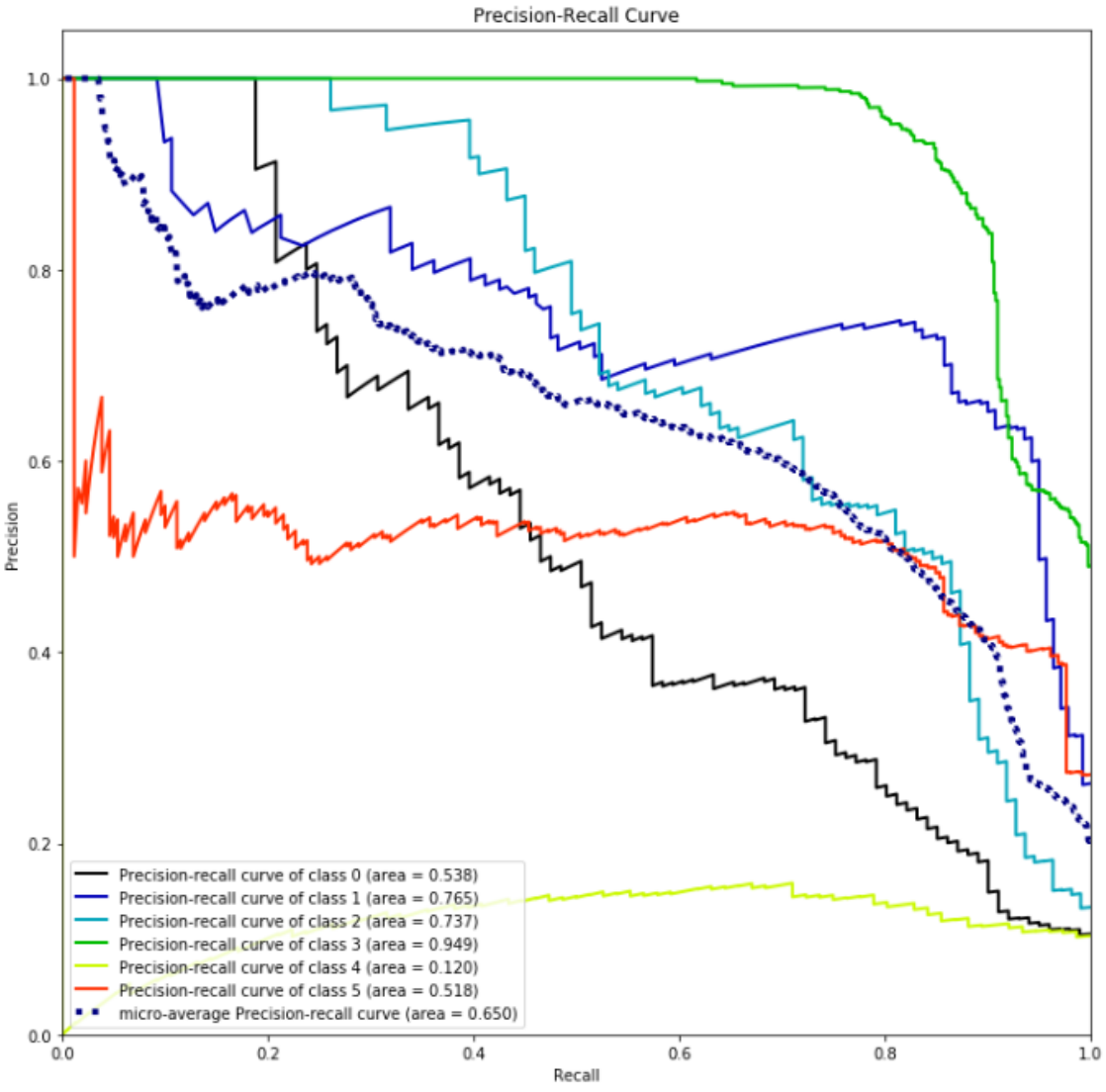

Figure 15: Precision-Recall Curve - VGG16

Figure 15 shows the precision-recall curve for VGG-16. The value obtained is the best for the class Red Deer which is class 3. It performs worst for the class Red Squirrel which is class 4 and it is also the minority class.

2. InceptionV3

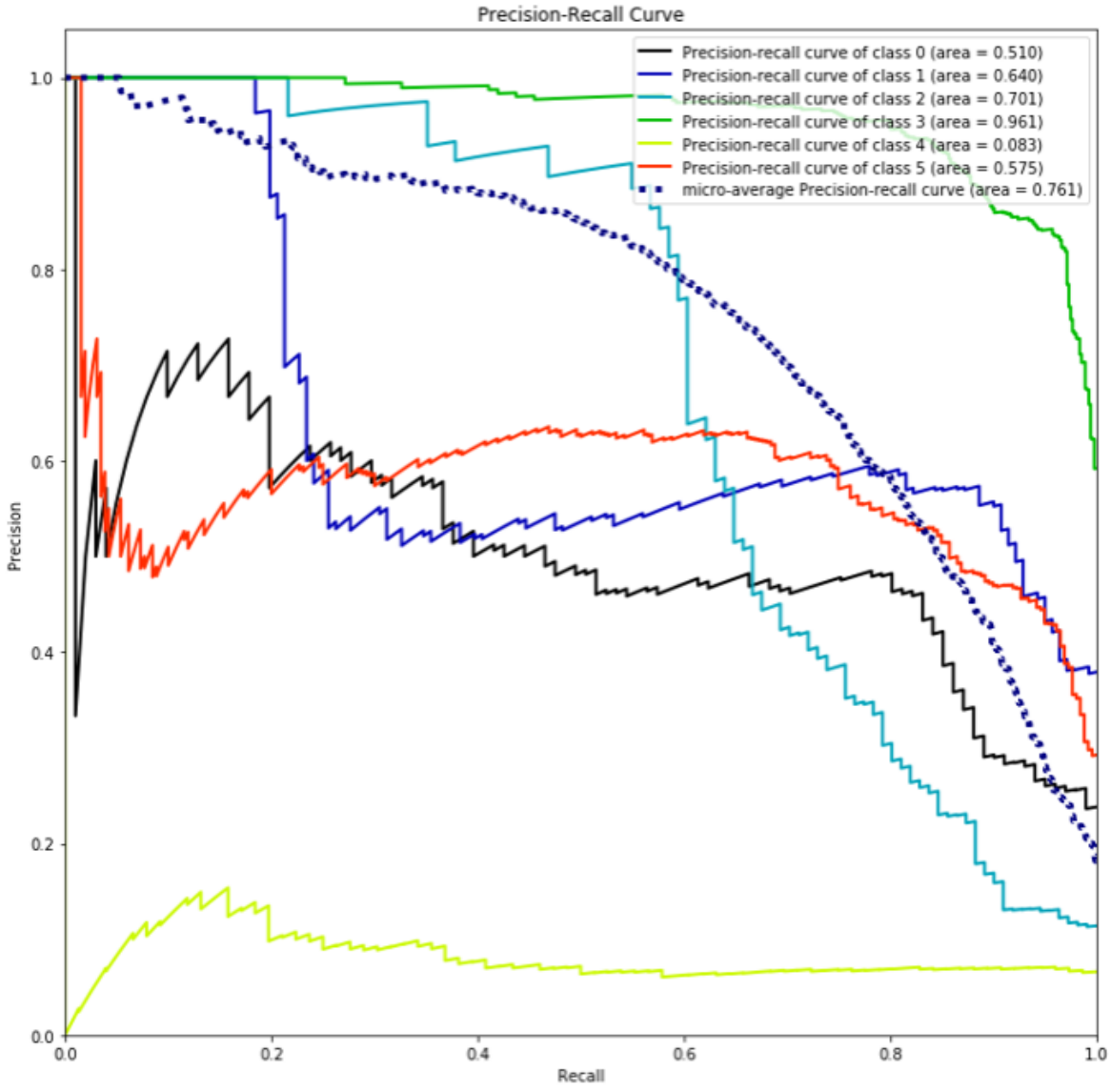

Figure 16: Precision-Recall Curve - InceptionV3

Figure 16 refers to the preicison-recall curve for InceptionV3. The highest value obtained is for class 3 which is Red Deer as shown in Figure 14. The lowest value achieved is area = 0.510 for class 0 which is for collared peccary.

3. MobileNet

   Figure 17 shows the precision-recall curve for MobileNet. Value of area = 0.965 is obtained for class 3. Lowest value is achieved for class 4 which is Red Squirrel as per Figure 14.

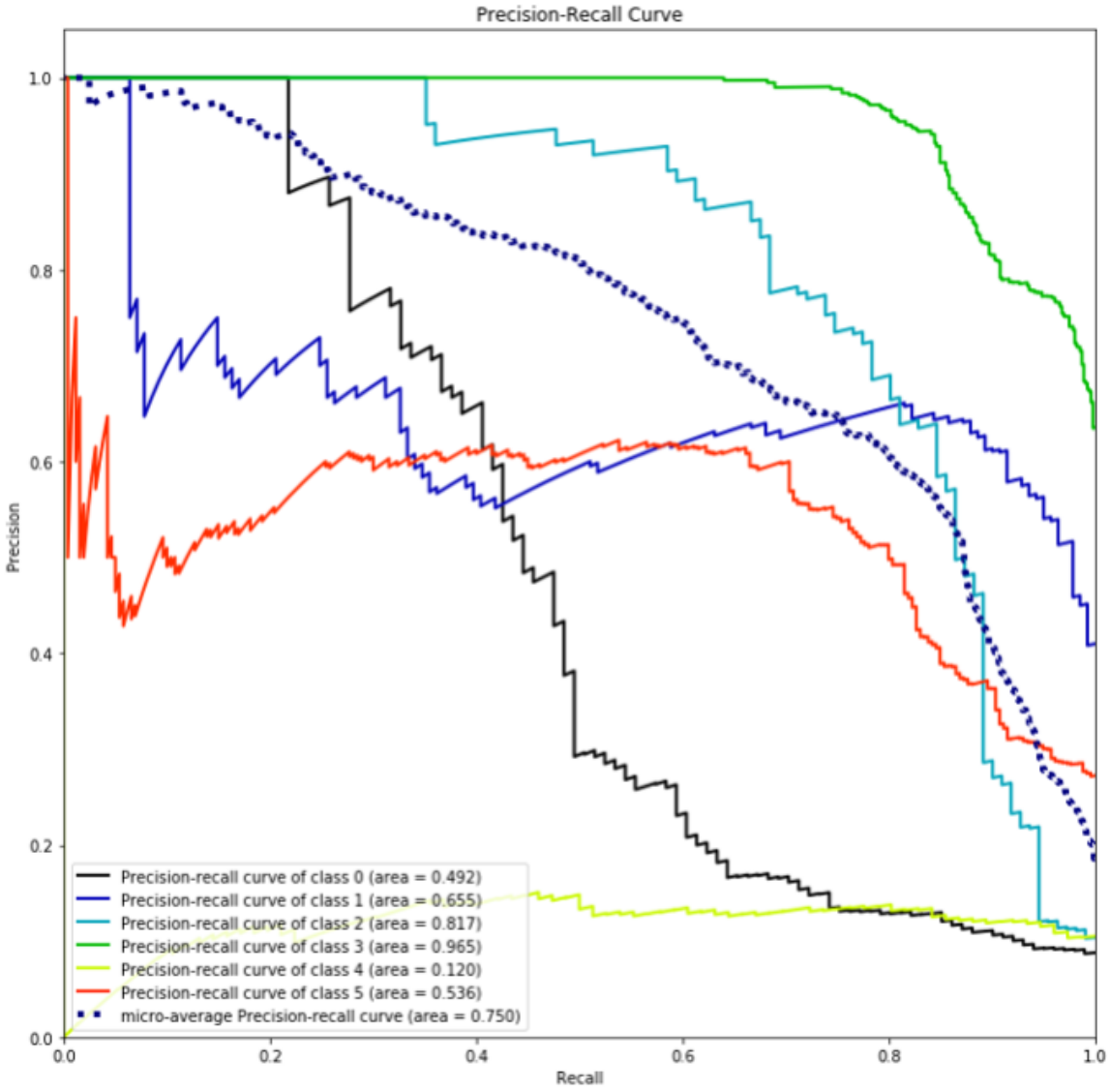

Figure 17: Precision-Recall Curve - MobileNet

Figure 15, Figure 16 and Figure 17 shows the Precision-Recall Curves for the models VGG-16, InceptionV3 and MobileNet. The observation shows that all the models performs the best for class 3 which is Red Deer as per Figure 14.

6.5 ROC Curve

1. VGG-16

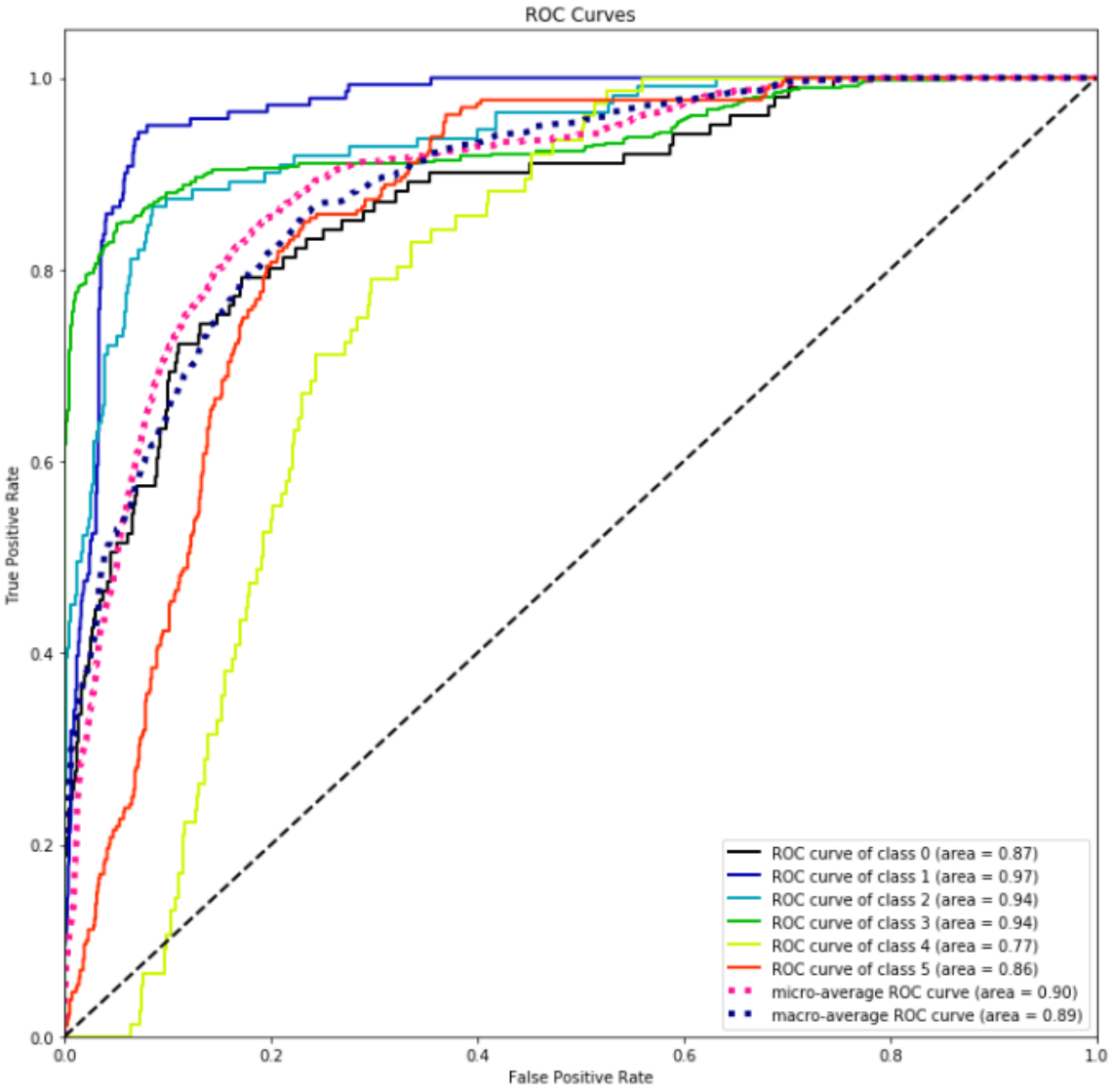

Figure 18: ROC Curve - VGG16

Figure 18 shows the ROC curve for VGG-16. The highest value is achieved for class 1 which refers to class labels from Figure 14. Class 1 refers to European Hare. The lowest value obtained is area = 0.77 for class 4 which is Red Squirrel.

2. InceptionV3

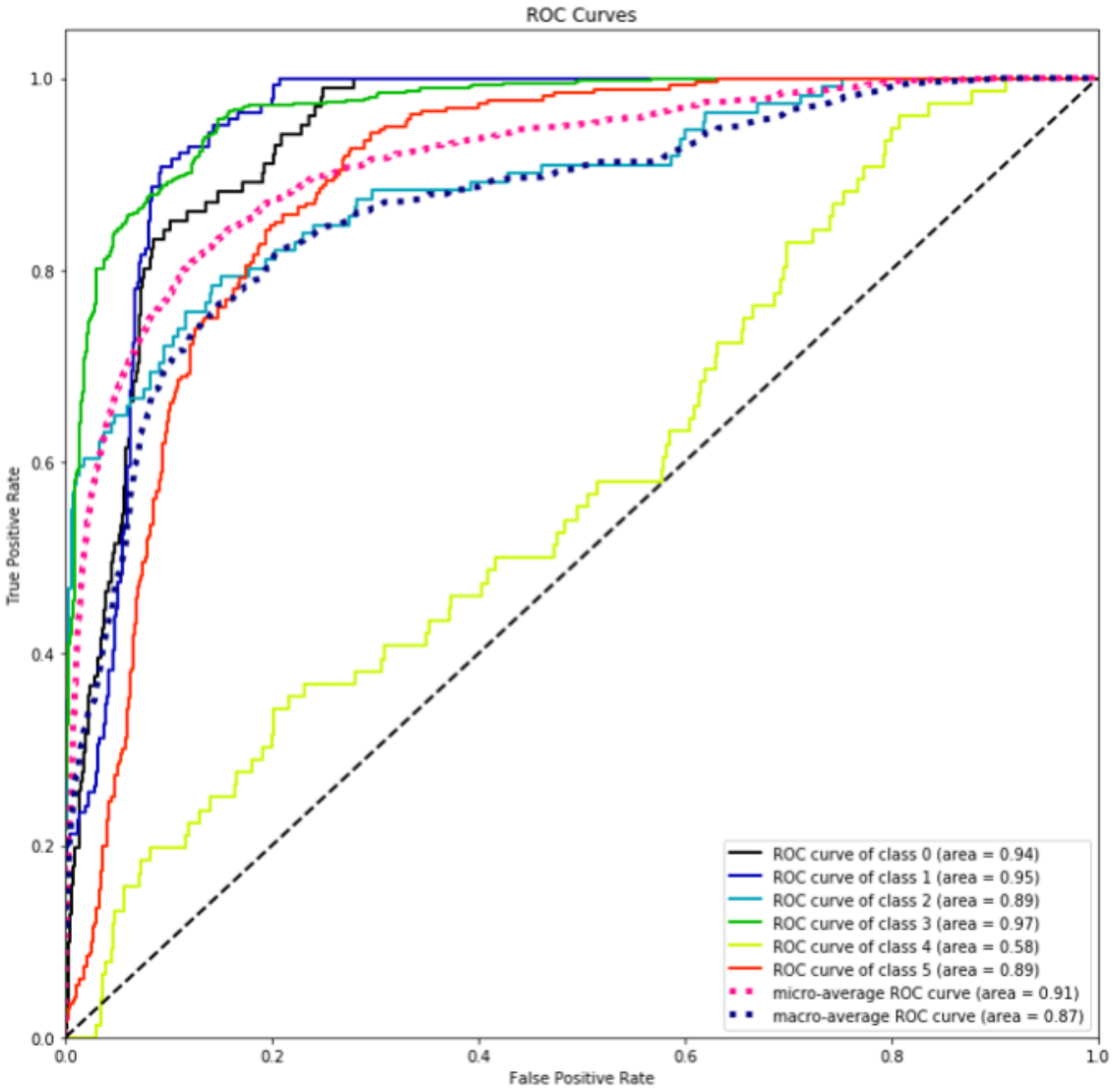

Figure 19: ROC Curve - InceptionV3

Figure 19 shows the results for the ROC curve for the six classes. Area = 0.97 is the highest value achieved for class 3, whereas 0.58 is the lowest value obtained for class 4. According to Figure 14, class 3 is Red Deer and class 4 is Red Squirrel.

3. MobileNet

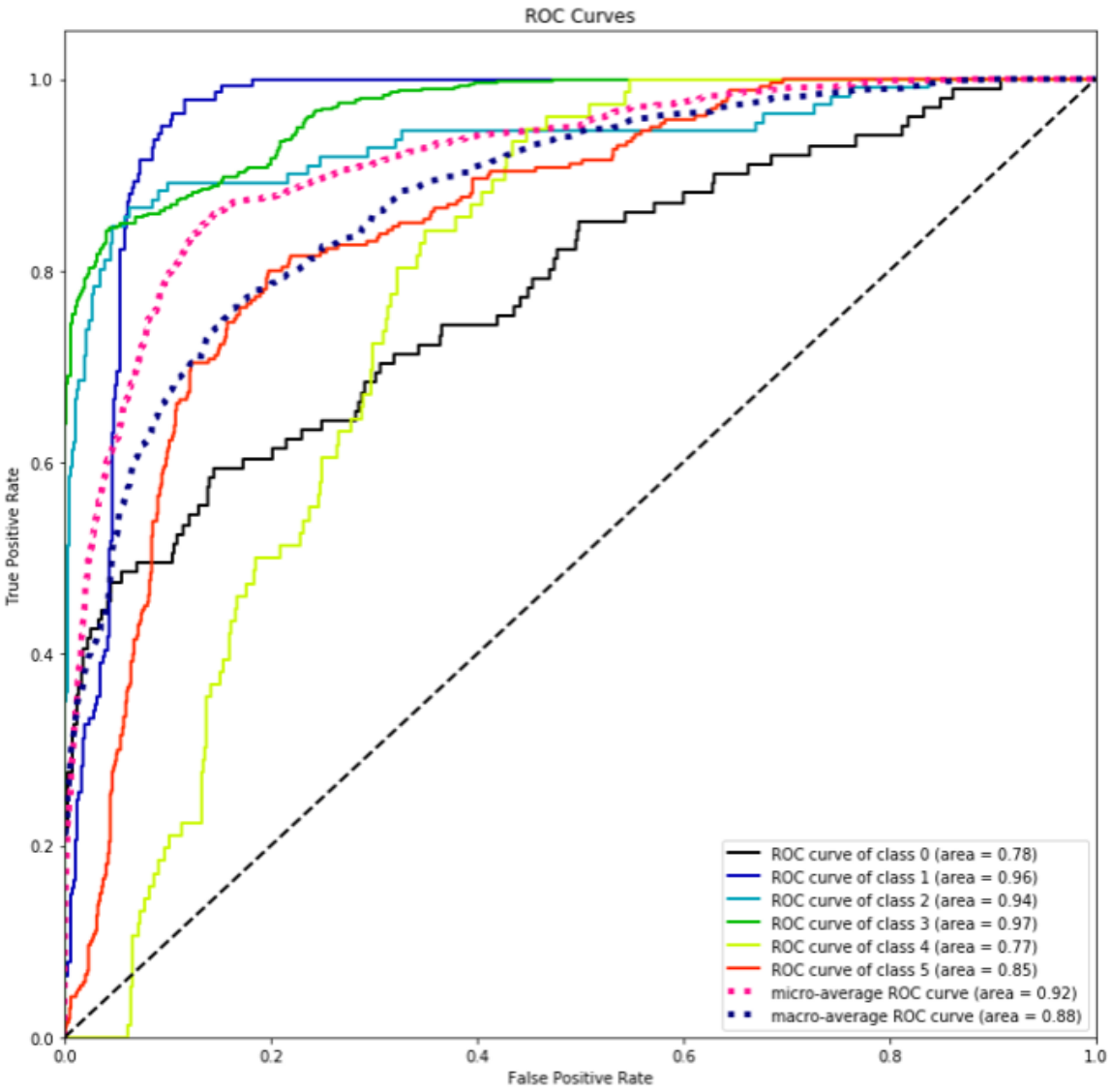

Figure 20: ROC Curve - MobileNet

Figure 20 refers to the results of ROC curve for the six classes. The highest value is achieved for class 3 which is Red Deer and the lowest value is obtained for class 4 which is Red Squirrel according to Figure 14.

VGG-16, InceptionV3 and MobileNet provides highest results of ROC curve for the class Red Deer as per Figure 14.[8]

[8]https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/

7 Discussion

Chen et al. (2014) used deep convolutional neural networks for animal classification and the accuracy achieved was 38.315%. We achieved the highest accuracy of 69.58% for the model InceptionV3. On the other hand, Kalita and Biswas (2019) achieved the highest accuracy of 88.27% for LeNet-5 architecture as compared to the highest accuracy of 69.58% as per our observations. Gutierrez-Galan et al. (2018) achieved an accuracy of 81% using Multi-Layer Perceptron.

Pan and Yang (2009) used transfer learning for classification. The weights generated in this experiment for VGG-16, InceptionV3 and MobileNet can be used in another dataset with similar animal species as transfer learning can help the new dataset to train quickly and efficiently in order to increase the accuracy.

The experiment is performed using Google Colaboratory on cloud and the major challenge faced in the experiment is the time taken to train the dataset. Six species out of 20 were taken into consideration. 13,807 is the image count for training, validation and testing after handling the class imbalance issue. For all the three models, VGG-16, InceptionV3 and MobileNet, 3 epochs were used to run the experiment. Average time taken for each epoch is nearly about 2 hours. Higher number of epochs as compared to 3 would have given a better accuracy. Distributed computing on Hadoop or Apache Spark can be used to cluster the dataset and run on higher number of epochs so that less time can be taken for the training of the dataset.[9]

Also, the results obtained could have been better if we had more number of images for the minority class. Here, the minority class is Red Squirrel. More observations for the rare class can allow the model to train well and increase the accuracy. For the majority class Red Deer, the results are the highest when it comes to evaluation metrics like accuracy, precision, recall and F1 score.

MobileNet is the architecture used for training the model and the results obtained are not bad as compared to the other two architectures. More epochs can help the model to learn well and perform better.

8 Ethical Implication

Missouri Camera Traps Dataset: Dataset of 20 species with approximately 25,000 images.

Zhang et al. (2016) worked on the dataset and it is open to researchers all over the world for research purposes as there is no issue of privacy.[10] The dataset can be used for computational purposes so that the models can perform better than the state-of-the-art. There won't be any ethical implications on the results obtained by this project and it can be published without complications.

[9] https://spark.apache.org/

[10] http://lila.science/datasets/missouricameratraps

9 Conclusion and Future Work

MobileNet architecture is the model trained on the Missouri Camera Traps dataset and it seems that the model performed well with F1 score of 0.68 as compared to the values 0.70 and 0.62 for InceptionV3 and VGG-16 respectively. Previous researchers used InceptionV3 and VGG-16 architectures and achieved better accuracy as observed in this experiment. Now since the datasets can be trained with MobileNet as well, higher accuracy can be achieved with the architecture in future. Weights from this experiment for all the three models can be used to train other datasets with similar classes using transfer learning. Models can perform better if more observations of the rare class can be captured. In this case, Red Squirrel is the rare class. More observations can assist the citizen community to conserve the species better. Wildlife conservation can be improved by implementing the deep learning architectures on the dataset generated from camera-trap images all across the world.

10 Acknowledgements

I would like to sincerely thank my mentor and guide Dr. Vladimir Milosavljevic for guiding me with correct approaches and continuous feedbacks. Also, I am very grateful to the librarian Mr. Keith Brittle of National College of Ireland, Dublin for conducting a seminar on using the resources from the library website. Finally, I would like to appreciate the support of my family and friends for inspiring and supporting me for the duration of this project.

References

C. Miguel, A., Bayrakcismith, R., Ferre, E., Bales-Heisterkamp, C., Beard, J., Dioso, M., Grob, D., Hartley, R., Nguyen, T. and Weller, N. (2019). Identifying individual snow leopards from camera trap images, p. 36.

Chen, G., Han, T. X., He, Z., Kays, R. and Forrester, T. (2014). Deep convolutional neural network based species recognition for wild animal monitoring, *2014 IEEE International Conference on Image Processing, ICIP 2014* pp. 858–862.

Chung, C., Patel, S., Lee, R., Fu, L., Reilly, S., Ho, T., Lionetti, J., George, M. D. and Taylor, P. (2018). Very Deep Convolutional Networks for Large-Scale Image Recognization, *[Vgg]* **75**(6): 398–406.

Gomez Villa, A., Salazar, A. and Vargas, F. (2017). Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks, *Ecological Informatics* **41**: 24–32.

Gutierrez-Galan, D., Dominguez-Morales, J. P., Cerezuela-Escudero, E., Rios-Navarro, A., Tapiador-Morales, R., Rivas-Perez, M., Dominguez-Morales, M., Jimenez-Fernandez, A. and Linares-Barranco, A. (2018). Embedded neural network for real-time animal behavior classification, *Neurocomputing* **272**: 17–26.

Hansson, P. (2002). Fracture Analysis of Adhesive Joints Using The Finite Element Method, *Lund Institute of Technology* (February).

Kalita, S. and Biswas, M. (2019). Improved Convolutional Neural Networks for Hyperspectral Image Classification, *Advances in Intelligent Systems and Computing* **740**: 397–410.

Matuska, S., Hudec, R., Kamencay, P., Benco, M. and Zachariasova, M. (2014). Classification of Wild Animals based on SVM and Local Descriptors, *AASRI Procedia* **9**(Csp): 25–30.

Miao, Z., Gaynor, K. M., Wang, J., Liu, Z., Muellerklein, O., Norouzzadeh, M. S., McInturff, A., Bowie, R. C., Nathan, R., Yu, S. X. and Getz, W. M. (2019). Insights and approaches using deep learning to classify wildlife, *Scientific Reports* **9**(1): 1–9.

Nguyen, H., Maclagan, S. J., Nguyen, T. D., Nguyen, T., Flemons, P., Andrews, K., Ritchie, E. G. and Phung, D. (2018). Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring, *Proceedings - 2017 International Conference on Data Science and Advanced Analytics, DSAA 2017* **2018-Janua**(Figure 1): 40–49.

Norouzzadeh, M. S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M., Packer, C. and Clune, J. (2017). Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning, (1): 1–10.

Pan, S. J. and Yang, Q. (2009). Oppenheim - Se-ales y Sistemas.pdf, *IEEE Transactions on Knowledge and Data Engineering* pp. 1–15.

Rey, N., Volpi, M., Joost, S. and Tuia, D. (2017). Detecting animals in African Savanna with UAVs and the crowds, *Remote Sensing of Environment* **200**(October): 341–351.

Schneider, S., Taylor, G. W. and Kremer, S. (2018). Deep learning object detection methods for ecological camera trap data, *Proceedings - 2018 15th Conference on Computer and Robot Vision, CRV 2018* pp. 321–328.

Swanson, A., Kosmala, M., Lintott, C., Simpson, R., Smith, A. and Packer, C. (2015). Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna, *Scientific Data* **2**: 1–14.

Verma, G. K. and Gupta, P. (2018). *Proceedings of 2nd International Conference on Computer Vision & Image Processing*, Vol. 704, Springer Singapore.

Yosinski, J., Clune, J., Bengio, Y. and Lipson, H. (2014). How transferable are features in deep neural networks?

Yousif, H., Yuan, J., Kays, R. and He, Z. (2017). Fast human-animal detection from highly cluttered camera-trap images using joint background modeling and deep learning classification, *Proceedings - IEEE International Symposium on Circuits and Systems* .

Yousif, H., Yuan, J., Kays, R. and He, Z. (2018). Object detection from dynamic scene using joint background modeling and fast deep learning classification, *Journal of Visual Communication and Image Representation* **55**: 802–815.

Yousif, H., Yuan, J., Kays, R. and He, Z. (2019). Animal Scanner: Software for classifying humans, animals, and empty frames in camera trap images, *Ecology and Evolution* **9**(4): 1578–1589.

Yu, X., Wang, J., Kays, R., Jansen, P. A., Wang, T. and Huang, T. (2013). Automated identification of animal species in camera trap images, *Eurasip Journal on Image and Video Processing* **2013**.

Zhang, Z., He, Z., Cao, G. and Cao, W. (2016). Animal detection from highly cluttered natural scenes using spatiotemporal object region proposals and patch verification, *IEEE Transactions on Multimedia* **18**(10): 2079–2092.