# Wildlife Species Classification in Camera Trap Images: Taï National Park


- Aseem Mishra
- Marttin Emmanuel
- Sagar Sheth


# Introduction

- The project is primarily motivated by the urgent need to enhance wildlife conservation and ecological research due to the imminent threats faced by Earth's biodiversity, including habitat loss, climate change, and human activities.

- Through the development of an image classification model, we aim to support wildlife preservation and environmental stewardship by enabling the efficient monitoring and categorization of wildlife species, facilitating more informed conservation decisions.

# Background

- In the realm of wildlife conservation and ecological research, data collection is often a challenging and resource-intensive task.
- Traditional methods for monitoring and tracking wildlife populations involve field surveys, camera traps, and manual observations, which are not only time-consuming but also labor-intensive.
- However, with the advancements in machine learning and computer vision, it is now possible to leverage technology to automate this process to a large extent.
- Our project focuses on the problem of image classification to identify seven specific species: birds, civets, duikers, hogs, leopards, other monkeys, and rodents. We are also addressing the issue of images that contain no animals.
- The potential impact of this project extends to various areas, including the efficient management of protected areas, understanding species distribution and behavior, and aiding researchers in their studies.

# Problem Statement

Our aim is to develop a model capable of classifying images into one of seven specified species categories or as containing no animals.

The Categories are :

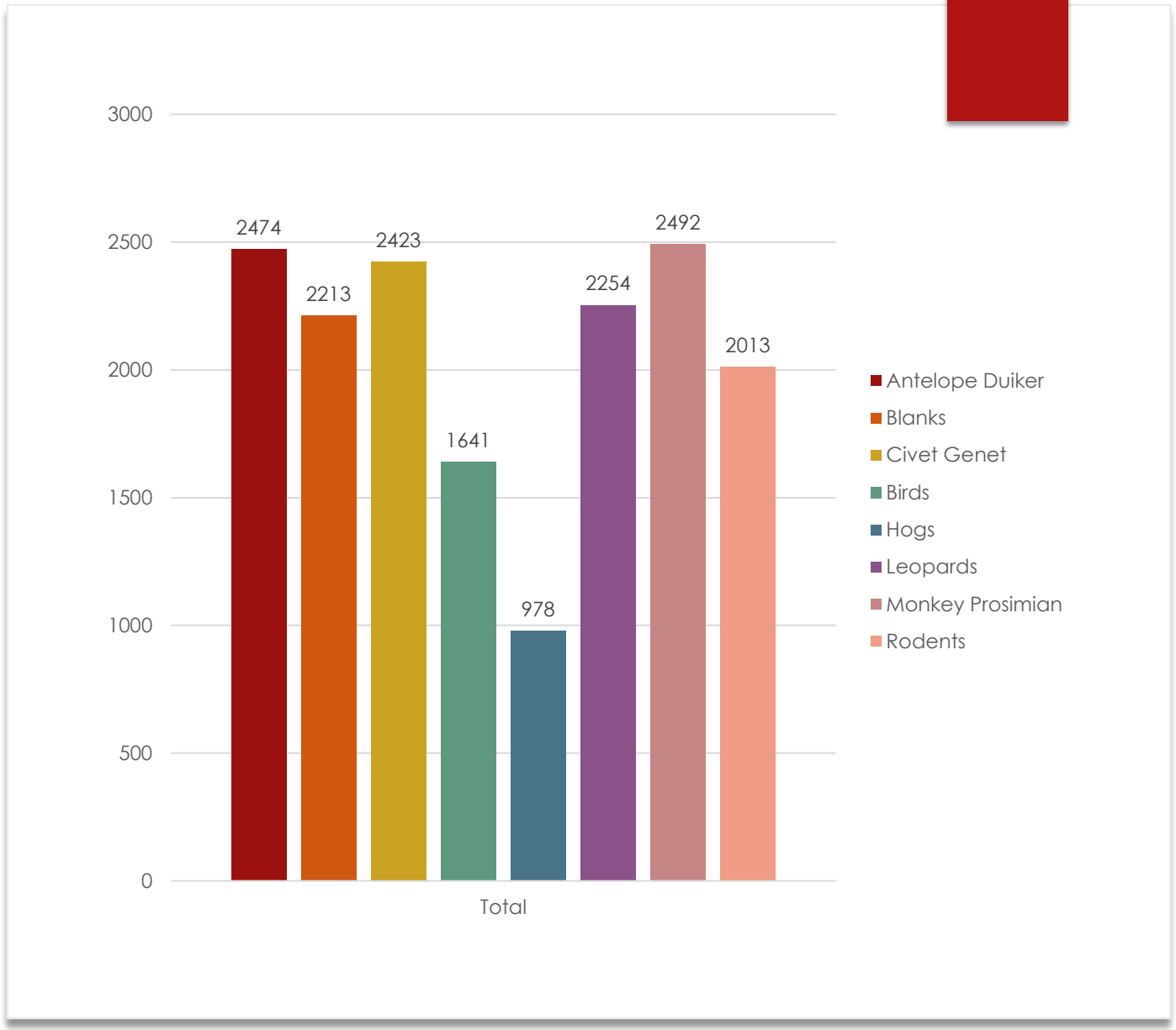
| | | | | | | |
|---|---|---|---|---|---|---|
| Antelope Duiker | Civet Genet | Bird | Hog | Leopard | Monkey Prosimian | Rodent |

# The Dataset

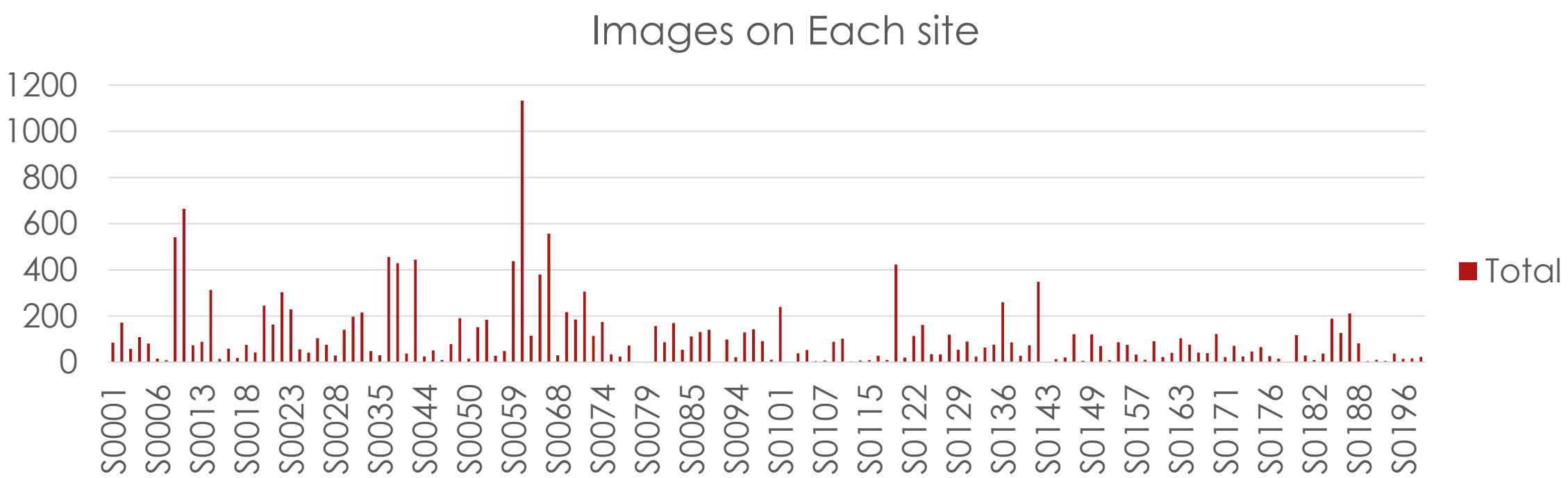
- The Dataset has been provided by drivendata.org.
- It has over 16000 training images and 4000 test images.
- Each image is .jpg file
- Each image has been assigned its site, filepath and its label for training set.
- The site is the location where the image was taken
- The resolution of the image varies and is not fixed.
- There is no overlap between sites in testing data and training data

# Data Exploration

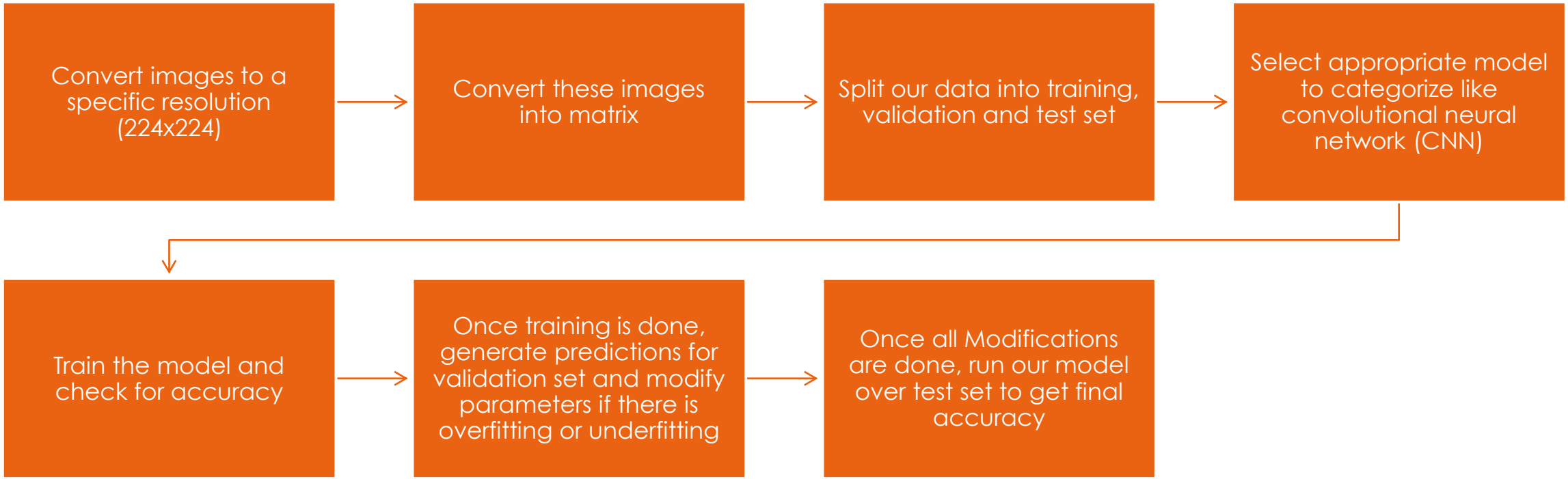
- Data Description



# Data Exploration



# Baseline Approach



# Preliminary Test Results

- Accuracy of 72% on validation set and 89.7% on Training set indicating variance over following Neural net configuration:
  - 2 Convolution layer with 32 filter and 64 filters respectively
  - 2 Max pooling layers each after a convolution layer
  - 1 Flatten Layer
  - 2 Dense layers with 128 neurons and 8 neurons respectively

```
Epoch 1/10
110/110 [==============================] - 73s 641ms/step - loss: 2.0637 - accuracy: 0.3180
Epoch 2/10
110/110 [==============================] - 67s 608ms/step - loss: 1.2347 - accuracy: 0.5751
Epoch 3/10
110/110 [==============================] - 747s 7s/step - loss: 0.8978 - accuracy: 0.7071
Epoch 4/10
110/110 [==============================] - 70s 637ms/step - loss: 0.6955 - accuracy: 0.7611
Epoch 5/10
110/110 [==============================] - 69s 629ms/step - loss: 0.5788 - accuracy: 0.8111
Epoch 6/10
110/110 [==============================] - 73s 660ms/step - loss: 0.5028 - accuracy: 0.8369
Epoch 7/10
110/110 [==============================] - 74s 672ms/step - loss: 0.4342 - accuracy: 0.8586
Epoch 8/10
110/110 [==============================] - 67s 608ms/step - loss: 0.3615 - accuracy: 0.8840
Epoch 9/10
110/110 [==============================] - 78s 710ms/step - loss: 0.3213 - accuracy: 0.8923
Epoch 10/10
110/110 [==============================] - 72s 656ms/step - loss: 0.3150 - accuracy: 0.8974
47/47 [==============================] - 9s 179ms/step - loss: 1.2960 - accuracy: 0.7213
Test accuracy: 0.7213333249092102
```

# Final Approach

Convert images to a specific resolution (224x224)

→

Convert these images into cv2 matrix

→

Split our data into training, validation and test set

→

Train the Data on the below 3 Models

1. 2 CNN Layers and 1 Dense Layer.
2. 2 CNN Layers and 2 Dense Layers.
3. Using Transfer learning from Image net.

↓

Evaluate accuracy of each model and choose the best approach

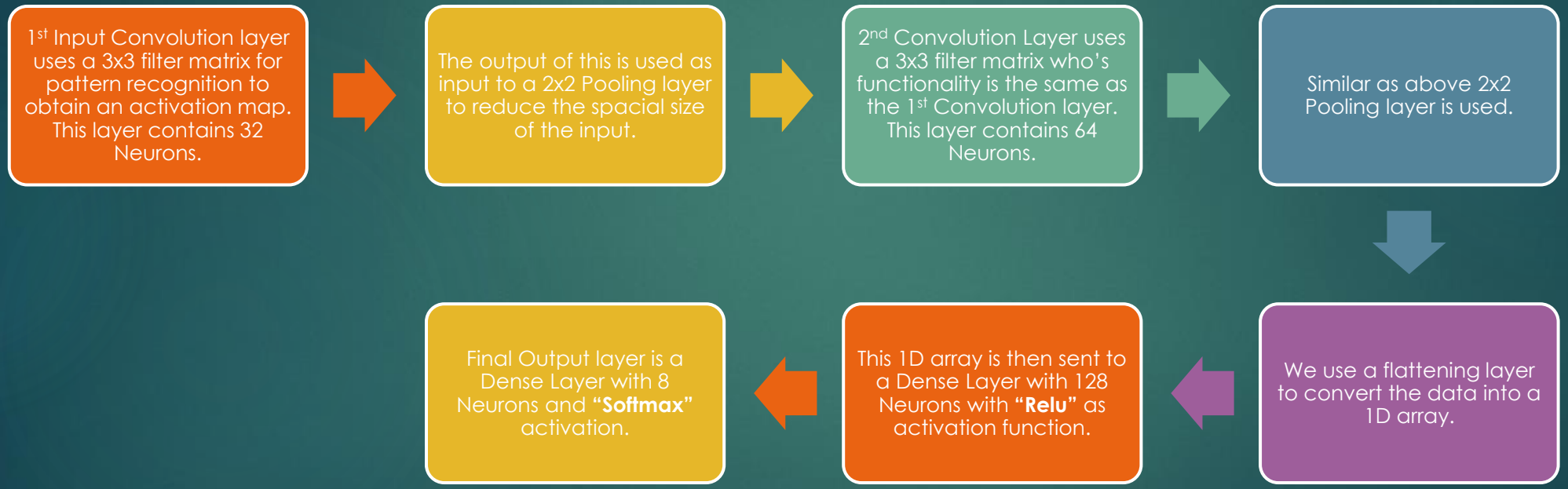
→

After choosing the best model, generate predictions for validation set and modify parameters if there is overfitting or underfitting

→

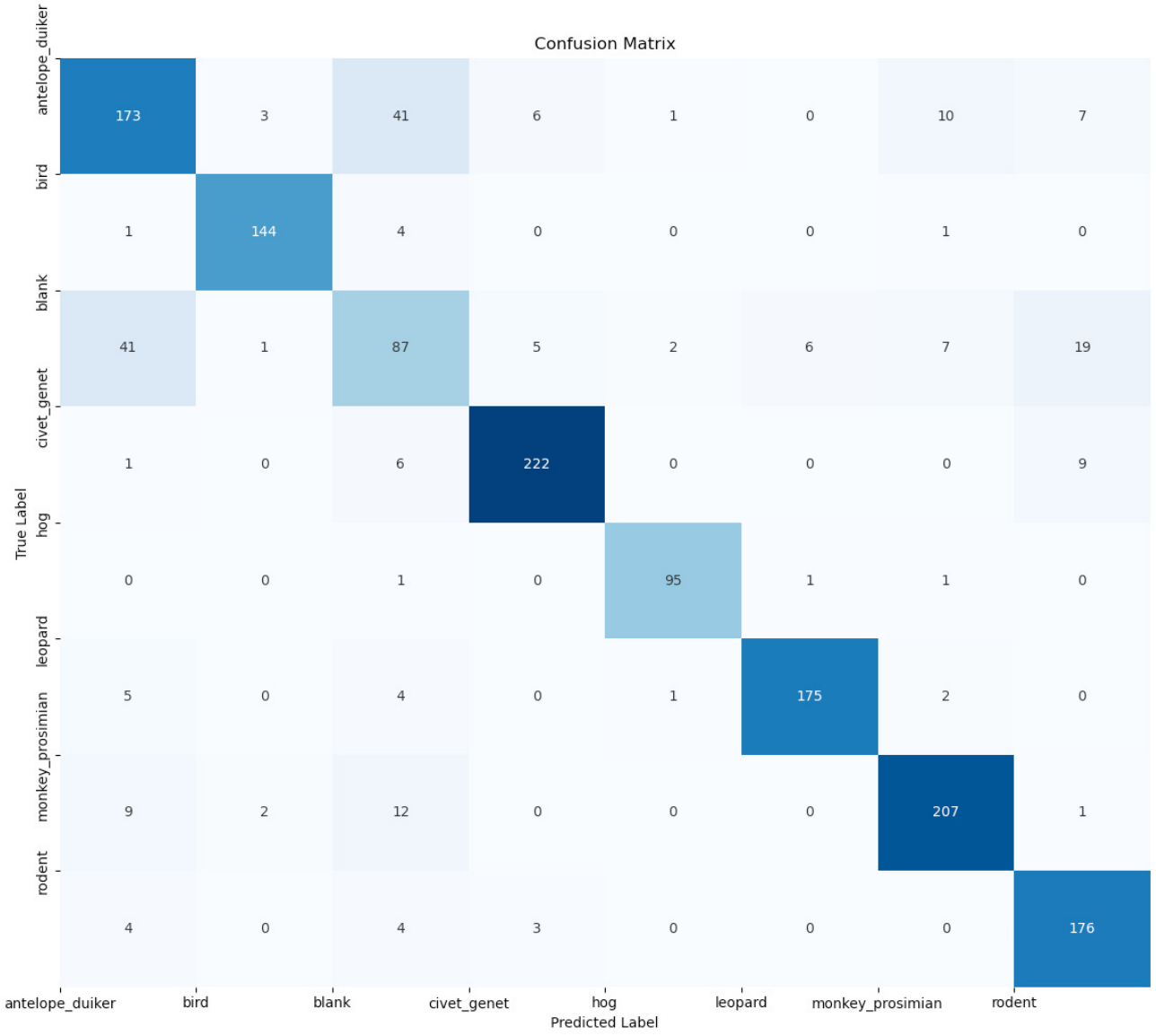
Once all Modifications are done, run our model over test set to get final accuracy

# Model 1


1st Input Convolution layer uses a 3x3 filter matrix for pattern recognition to obtain an activation map. This layer contains 32 Neurons.
The output of this is used as input to a 2x2 Pooling layer to reduce the spacial size of the input.
2nd Convolution Layer uses a 3x3 filter matrix who's functionality is the same as the 1st Convolution layer. This layer contains 64 Neurons.
Similar as above 2x2 Pooling layer is used.
We use a flattening layer to convert the data into a 1D array.
This 1D array is then sent to a Dense Layer with 128 Neurons with "Relu" as activation function.
Final Output layer is a Dense Layer with 8 Neurons and "Softmax" activation.

# Model 1 Training Performance

| Epoch | loss | Train Accuracy | val_loss | val_accuracy |
|---|---|---|---|---|
| 1/10 | 1.6485 | 0.4776 | 1.0136 | 0.6533 |
| 2/10 | 0.8253 | 0.7172 | 0.7521 | 0.7533 |
| 3/10 | 0.5804 | 0.8067 | 0.7077 | 0.7659 |
| 4/10 | 0.4464 | 0.8516 | 0.5932 | 0.8148 |
| 5/10 | 0.3589 | 0.879 | 0.5986 | 0.8278 |
| 6/10 | 0.2985 | 0.9022 | 0.67 | 0.8226 |
| 7/10 | 0.2571 | 0.9136 | 0.5778 | 0.8481 |
| 8/10 | 0.2184 | 0.93 | 0.6611 | 0.8433 |
| 9/10 | 0.2042 | 0.9331 | 0.5969 | 0.8478 |
| 10/10 | 0.1775 | 0.9462 | 0.5841 | 0.8552 |




# Model 1 Confusion Matrix (Holdout Set)

# Model 1 Evaluation (Holdout Set)

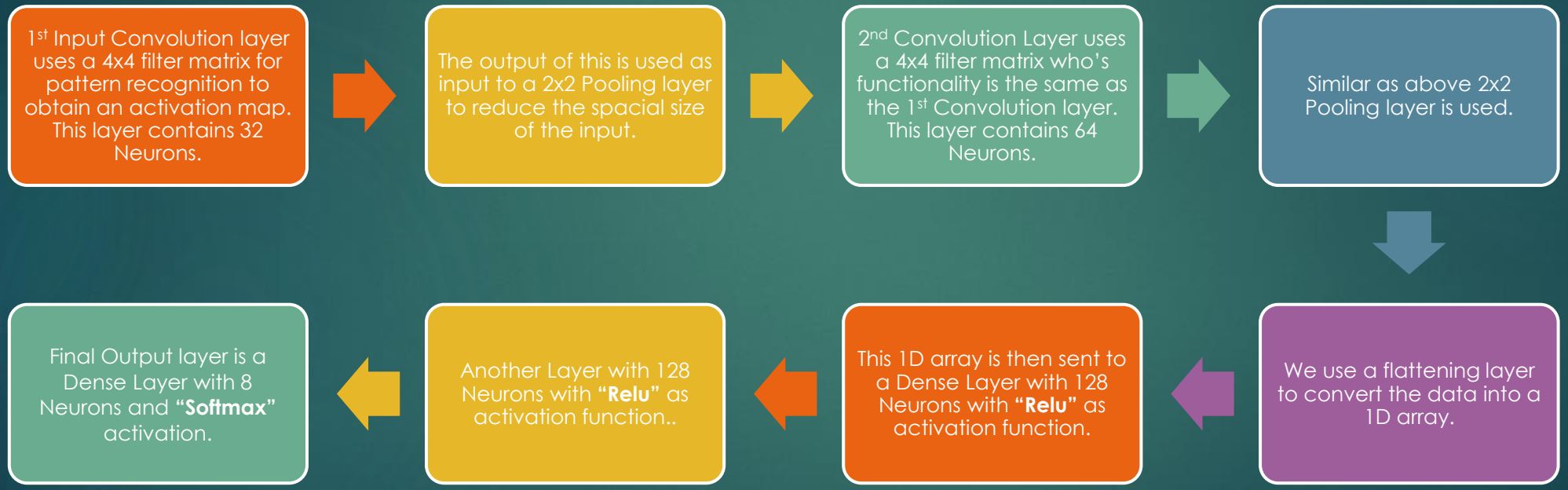
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| antelope_duiker | 0.74 | 0.72 | 0.73 | 241 |
| bird | 0.96 | 0.96 | 0.96 | 150 |
| blank | 0.55 | 0.52 | 0.53 | 168 |
| civet_genet | 0.94 | 0.93 | 0.94 | 238 |
| hog | 0.96 | 0.97 | 0.96 | 98 |
| leopard | 0.96 | 0.94 | 0.95 | 187 |
| monkey_prosimian | 0.91 | 0.9 | 0.9 | 231 |
| rodent | 0.83 | 0.94 | 0.88 | 187 |
| accuracy | | | 0.85 | 1500 |
| macro avg | 0.86 | 0.86 | 0.86 | 1500 |
| weighted avg | 0.85 | 0.85 | 0.85 | 1500 |

$$loss = -\frac{1}{N} \cdot \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} \log p_{ij}$$

# Model 1 Evaluation Metric (Holdout set) As per Competition

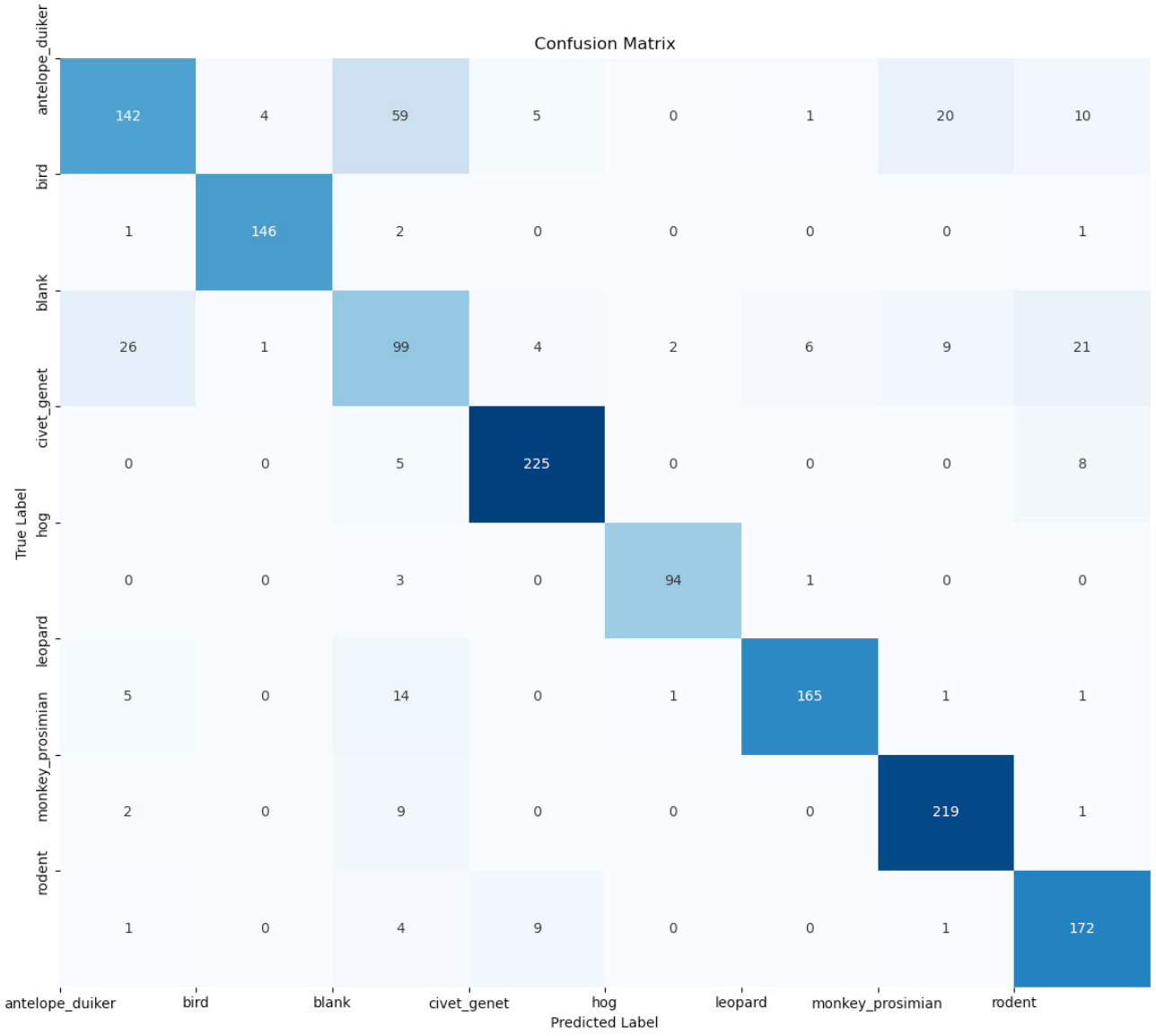
**LOG LOSS: 5.310431599329929**

# Model 2


1st Input Convolution layer uses a 4x4 filter matrix for pattern recognition to obtain an activation map. This layer contains 32 Neurons.
The output of this is used as input to a 2x2 Pooling layer to reduce the spacial size of the input.
2nd Convolution Layer uses a 4x4 filter matrix who's functionality is the same as the 1st Convolution layer. This layer contains 64 Neurons.
Similar as above 2x2 Pooling layer is used.
We use a flattening layer to convert the data into a 1D array.
This 1D array is then sent to a Dense Layer with 128 Neurons with "Relu" as activation function.
Another Layer with 128 Neurons with "Relu" as activation function..
Final Output layer is a Dense Layer with 8 Neurons and "Softmax" activation.

# Model 2 Training Performance

| Epoch | loss | Train Accuracy | val_loss | val_accuracy |
|---|---|---|---|---|
| 1/10 | 1.4448 | 0.467 | 1.0601 | 0.6341 |
| 2/10 | 0.8382 | 0.7059 | 0.7758 | 0.7478 |
| 3/10 | 0.5618 | 0.8055 | 0.6781 | 0.7696 |
| 4/10 | 0.4719 | 0.8381 | 0.6456 | 0.7937 |
| 5/10 | 0.3534 | 0.8797 | 0.6427 | 0.8211 |
| 6/10 | 0.2934 | 0.8994 | 0.634 | 0.8304 |
| 7/10 | 0.251 | 0.9138 | 0.6247 | 0.8422 |
| 8/10 | 0.2173 | 0.9242 | 0.6263 | 0.8522 |
| 9/10 | 0.2059 | 0.9299 | 0.6492 | 0.8404 |
| 10/10 | 0.171 | 0.9426 | 0.6462 | 0.8463 |




# Model 2 Confusion Matrix (Holdout Set)

# Model 2 Evaluation (Holdout Set)

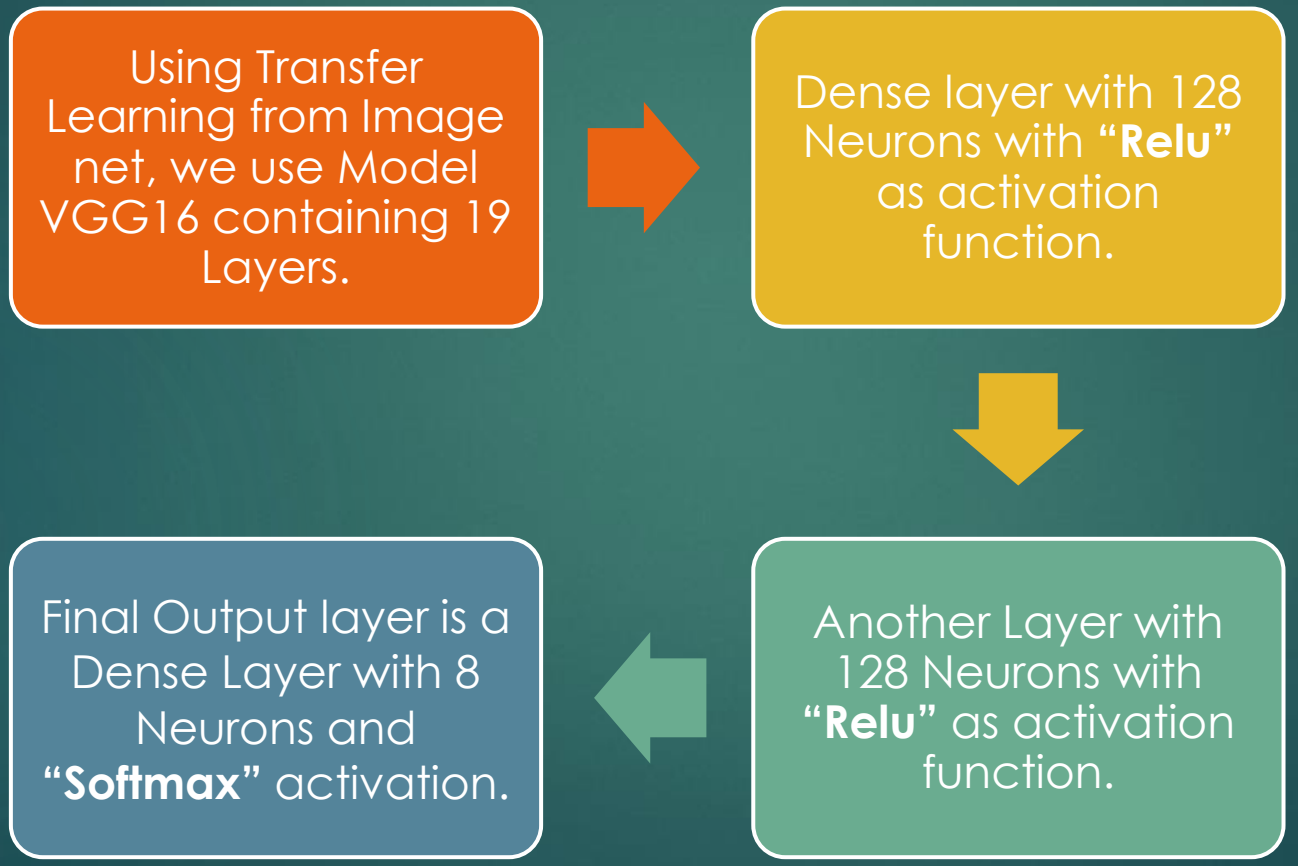
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| antelope_duiker | 0.8 | 0.59 | 0.68 | 241 |
| bird | 0.97 | 0.97 | 0.97 | 150 |
| blank | 0.51 | 0.59 | 0.55 | 168 |
| civet_genet | 0.93 | 0.95 | 0.94 | 238 |
| hog | 0.97 | 0.96 | 0.96 | 98 |
| leopard | 0.95 | 0.88 | 0.92 | 187 |
| monkey_prosimian | 0.88 | 0.95 | 0.91 | 231 |
| rodent | 0.8 | 0.92 | 0.86 | 187 |
| accuracy | | | 0.84 | 1500 |
| macro avg | 0.85 | 0.85 | 0.85 | 1500 |
| weighted avg | 0.85 | 0.84 | 0.84 | 1500 |

$$loss = -\frac{1}{N} \cdot \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} \log p_{ij}$$

# Model 2 Evaluation Metric (Holdout set) As per Competition

LOG LOSS: 5.718926337739923

# Model 3

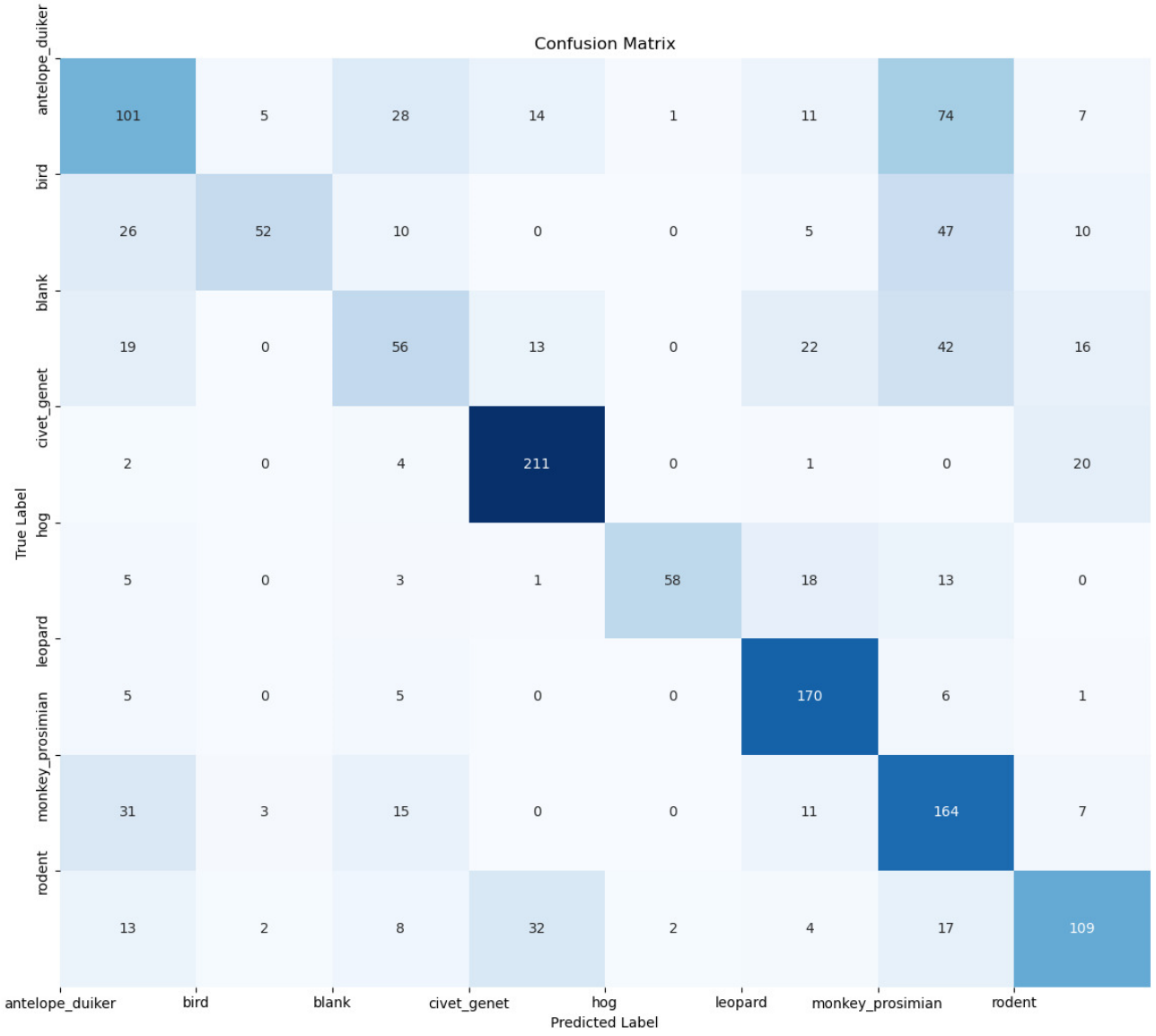

Using Transfer Learning from Image net, we use Model VGG16 containing 19 Layers.
Dense layer with 128 Neurons with "Relu" as activation function.
Another Layer with 128 Neurons with "Relu" as activation function.
Final Output layer is a Dense Layer with 8 Neurons and "Softmax" activation.

| Layer (type) | Output Shape | Param # |
|---|---|---|
| input_4 (InputLayer) | [(None, None, None, 3)] | 0 |
| block1_conv1 (Conv2D) | (None, None, None, 64) | 1792 |
| block1_conv2 (Conv2D) | (None, None, None, 64) | 36928 |
| block1_pool (MaxPooling2D) | (None, None, None, 64) | 0 |
| block2_conv1 (Conv2D) | (None, None, None, 128) | 73856 |
| block2_conv2 (Conv2D) | (None, None, None, 128) | 147584 |
| block2_pool (MaxPooling2D) | (None, None, None, 128) | 0 |
| block3_conv1 (Conv2D) | (None, None, None, 256) | 295168 |
| block3_conv2 (Conv2D) | (None, None, None, 256) | 590080 |
| block3_conv3 (Conv2D) | (None, None, None, 256) | 590080 |
| block3_pool (MaxPooling2D) | (None, None, None, 256) | 0 |
| block4_conv1 (Conv2D) | (None, None, None, 512) | 1180160 |
| block4_conv2 (Conv2D) | (None, None, None, 512) | 2359808 |
| block4_conv3 (Conv2D) | (None, None, None, 512) | 2359808 |
| block4_pool (MaxPooling2D) | (None, None, None, 512) | 0 |
| block5_conv1 (Conv2D) | (None, None, None, 512) | 2359808 |
| block5_conv2 (Conv2D) | (None, None, None, 512) | 2359808 |
| block5_conv3 (Conv2D) | (None, None, None, 512) | 2359808 |
| block5_pool (MaxPooling2D) | (None, None, None, 512) | 0 |
| global_average_pooling2d_3 (GlobalAveragePooling2D) | (None, 512) | 0 |
| dense_26 (Dense) | (None, 128) | 65664 |
| dense_27 (Dense) | (None, 128) | 16512 |
| dense_28 (Dense) | (None, 8) | 1032 |

# VGG16

# Model 3 Training Performance

| Epoch | loss | Train Accuracy | val_loss | val_accuracy |
|---|---|---|---|---|
| 1/20 | 0.9799 | 0.636 | 1.0168 | 0.6263 |
| 2/20 | 0.952 | 0.6466 | 0.9589 | 0.6515 |
| 3/20 | 0.922 | 0.6615 | 1.0022 | 0.6448 |
| 4/20 | 0.8968 | 0.6724 | 0.9483 | 0.6574 |
| 5/20 | 0.8803 | 0.6806 | 0.9231 | 0.6696 |
| 6/20 | 0.8553 | 0.6868 | 0.9054 | 0.6793 |
| 7/20 | 0.8372 | 0.6917 | 0.9324 | 0.6626 |
| 8/20 | 0.8184 | 0.7031 | 0.8867 | 0.6759 |
| 9/20 | 0.8091 | 0.7017 | 0.927 | 0.6619 |
| 10/20 | 0.7805 | 0.7138 | 0.8801 | 0.6833 |
| 11/20 | 0.7694 | 0.7175 | 0.8674 | 0.6856 |
| 12/20 | 0.7583 | 0.7226 | 0.8499 | 0.7011 |
| 13/20 | 0.7558 | 0.7194 | 0.8393 | 0.7044 |
| 14/20 | 0.7308 | 0.7336 | 0.8546 | 0.6944 |
| 15/20 | 0.7178 | 0.7374 | 0.8113 | 0.7107 |
| 16/20 | 0.7018 | 0.7426 | 0.8036 | 0.717 |
| 17/20 | 0.6918 | 0.7468 | 0.7951 | 0.7152 |
| 18/20 | 0.6743 | 0.7537 | 0.8207 | 0.7163 |
| 19/20 | 0.67 | 0.7533 | 0.8125 | 0.7237 |
| 20/20 | 0.6631 | 0.7594 | 0.7688 | 0.72 |




# Model 3 Confusion Matrix (Holdout Set)

# Model 3 Evaluation (Holdout Set)

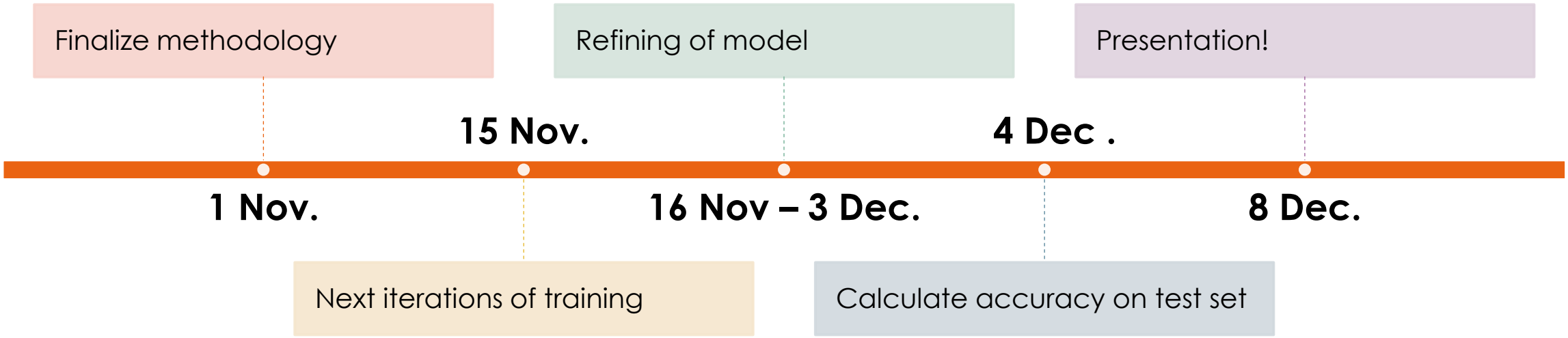
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| antelope_duiker | 0.5 | 0.42 | 0.46 | 241 |
| bird | 0.84 | 0.35 | 0.49 | 150 |
| blank | 0.43 | 0.33 | 0.38 | 168 |
| civet_genet | 0.78 | 0.89 | 0.83 | 238 |
| hog | 0.95 | 0.59 | 0.73 | 98 |
| leopard | 0.7 | 0.91 | 0.79 | 187 |
| monkey_prosimian | 0.45 | 0.71 | 0.55 | 231 |
| rodent | 0.64 | 0.58 | 0.61 | 187 |
| accuracy | | | 0.61 | 1500 |
| macro avg | 0.66 | 0.6 | 0.6 | 1500 |
| weighted avg | 0.64 | 0.61 | 0.6 | 1500 |

$$loss = -\frac{1}{N} \cdot \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} \log p_{ij}$$

# Model 3 Evaluation Metric (Holdout set) As per Competition

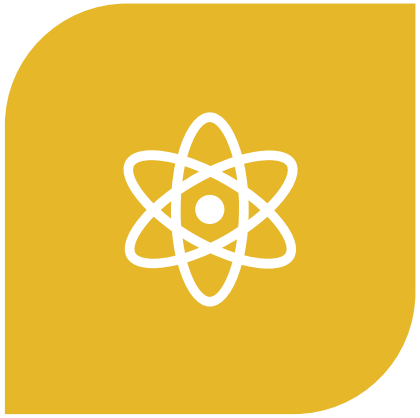
LOG LOSS: 13.912850208199224

# Project Timeline



# Team Roles

ASEEM MISHRA – MODEL DESIGN AND TRAINING



MARTTIN EMMANUEL – TESTING AND REFINING MODEL

SAGAR SHETH – DATA PRE PROCESSING

# Thank you