Performing Image Classification for 10 Different Monkey Species using CNN

By

Emmanuel Maduwuba, Dharanikota Rajendra Kamal and Kamaljeet Singh Mann.
Western University of Ontario
ECE 9309B: Machine Learning: From Theory to Applications.

***Abstract*— the main aim of this project is to achieve fine grain image classification by applying a suitable machine learning architecture to the set of images present in the dataset. The chosen dataset is taken as a part of the Kaggle competition and is selected from Wikipedia's monkey cladogram and this dataset contains 10 different species of monkeys which are to be classified with the help of a machine learning architecture augmented by Image processing. After having brief exposure and using several architectures to classify this dataset, the Convolutional Neural network was found to be the best fit.**

Nomenclature--

Convolutional neural networks, Image Augmentation, Feature extraction, Uniform aspect ratio, dimensionality reduction, training and validation accuracy, Image Scaling, activation function, Deep neural networks, multi-layer perceptron and confusion matrix.

I. Introduction

This document is used to describe, analyze, organize and explain the chosen project in a systematic and elaborate manner so that it could be read and understood by anyone. Machine learning aided with image processing is the kind of advancement that people require to make the life of the living better, Its applications range from creating the architectures that are needed to aid the drones for rescuing and identifying the people from the calamity and disaster struck areas to identifying the wild life and also to preserve ant identify the species of animals that are near extinction. This can play a key role by reducing the time and effort that it takes for a normal human being on foot to do some tasks such as making a record of animals in the sanctuary or the zoo and segregate them so this can be done with the help of machine learning where we build an analytical model/ architecture to do this task. So in order to make this task come to fruition we have opted the data set containing monkey images of ten different species and this data set will be explained in detail in the **DATASET DESCRIPTION.** So, in order to classify the monkey images, we have chosen the convolutional neural network neural network architecture, and this is applied in order to classify the set of images into their species. In order to make sure that the deep learning architecture is working properly we need to process the images beforehand to look for some unique features and also pin pointing what makes them different from every other image, by doing so we can make the model to work effectively. This can be done with the help of some preloaded modules in python like "**Sitk"** module that is designed for image processing and also the "**imgaug"** which is an image augmentation module. In order to do this image processing and image augmentation we must realize and understand precisely how the images work and how they are looked at in the perspective of a machine then and also understand the data set in detail and the factors that affect the data set , image processing and image augmentation which in turn effects the architecture. All this clarified bellow.

A. Image

An image is a collection of data. An image can vary from image of a landscape to an image of a person, irrespective of the type it is basic collection of data. This data is captured in the form of a matrix of numbers ranging from 0 to 255 and there are two different types of images they are monochromatic and RGB image. Monochromatic image is nothing but the image that has only two intensities of color (another name for a gray scale or black and white image) and the type is determined by the degrees of freedom given to the pixels of the image and a monochromatic image pixel has only one degree of freedom. The RGB image pixel has 3 degree of freedom, a RGB image has three channels for each colors that is red, blue and green colors and the mixture of these three colors intensities to give all different colors. The pixel of RGB image contains a tuple of these intensities. This tuple has a mixture of R, G and B numbers respectively.

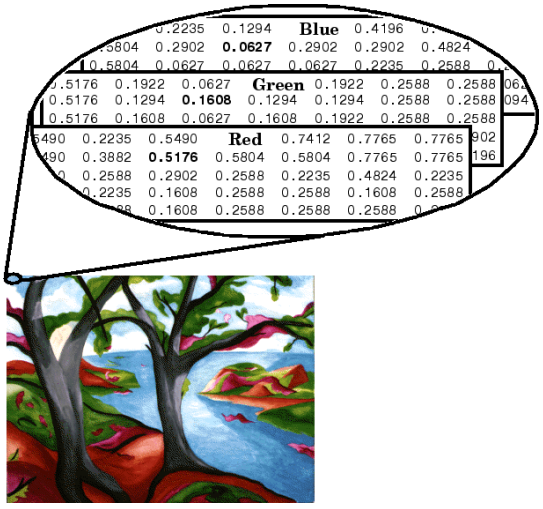

Fig 1: Machine point view of RGB image.

B. Feature Extraction

Feature extraction is used in order to increase the learning rate of the model and it used to basically understand what separates one image from another image. There are many features that can be extracted while image processing which will increase the learning rate at the same time increase the quantity of data that is being used to train and test the model. The more the data the more efficient the artificial neural network works. The features selected by the CNN depend on the type of the data and the images that are being used. The following are some of the features that can be used by the CNN internally.

1. Key points.
2. Changes in contrast.
3. Change In the dimensionality
4. Gray scaling
5. Edge and boundary detection.
6. Blob detection
7. Ridge Detection

The below example image shows the gray scaling, border detection and key points marking of some monkey images.

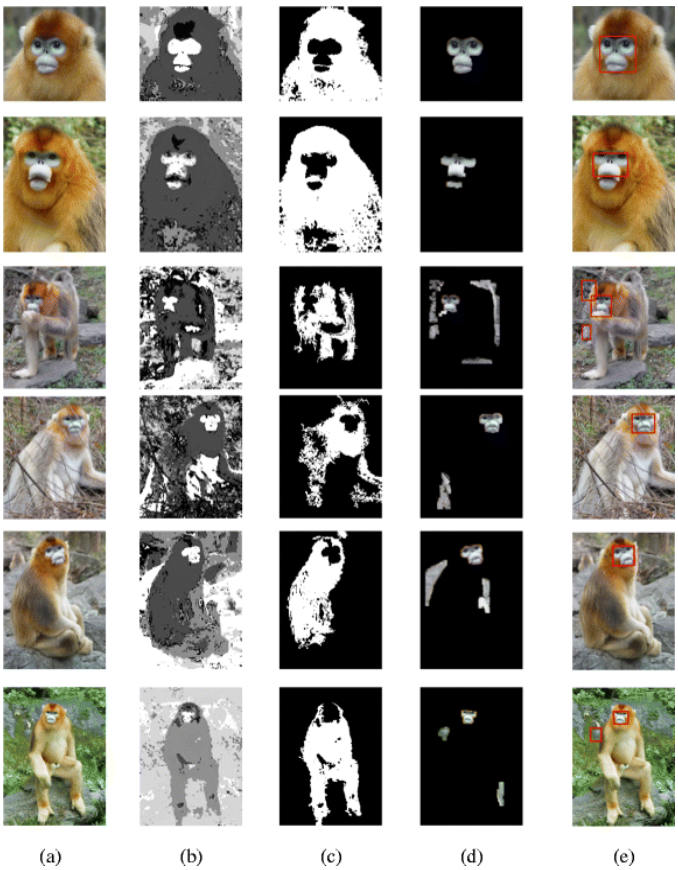

Fig 2: Example image showing some features selections like gray scaling, border selection and key point's selection.

1. Key points: -

The images are processed with key points the points that are unique to an image are selected like eyes, ears, nose, mouth etc. And this is fed during the training model. The intensities at these points are used to classify the images in the test set leading them to classify the images correctly and as accurately as possible. The testing image pixel intensities at these points are compared with the pixel intensities of the training model images for the image classification to be done precisely. These key points selected will not change no matter how the dimensionality changes.

2. Changes in contrast: -

The images are made to undergo contrast changes like increasing the exposure or decreasing it or varying the saturation and by doing so we get more data to use to train and test the model on, this is because the convolutional neural network is a deep learning neural network architecture which better works when there is abundant data. By doing this we even change the intensities of the image and thus making it more competitive to identify and classify the monkey species.

3. Change in the orientation and dimensionality: -

Change in dimensionality is nothing but varying the orientation of the image with respect to the standard plane that is if the back ground is a black space of the same size of the image and the fore ground is the image itself, then changing the angle or rotating the actual image with respect to the original image. This rotation is dealt by the affine component of the image processing. We do this in order to make sure the key points and the border selected are remaining constant no matter the orientation of the actual image with respect to the back ground plane and by doing so we ensure that the key point intensities are measured and classifies accurately.

4. Gray Scaling: -

Gray scaling is nothing but turning the RGB image into a black and white image this is achieved by setting a threshold, the values above the threshold are converted into one color of gray scale and the values below this threshold will be converted into the other color of grayscale. This can be used to accurately determine the structure of the image as well as the edges present in it. This contributes in its own way in feature extraction.

5. Edge and Boundary Detection: -

Edge and Boundary detection are used to determine the structure and spacing of the key components in the image for example the face of the monkey is a key component which helps in classification so the boundary is selected around the face of the monkey and the intensities with in this boundary lead to some fine grain image classification. In the similar fashion the edge detection also works, determining the shape of the key components in the image these key components are the parts that separate one image from another and is extremely helpful in classification.

6. Blob Detection: -

Blob detection is used in detecting and identifying regions in the RGB image with contrast, brightness and color variations in the region. This method is helps us to attain and provide complementary information about the regions that are found from both edge and corner detections. By using this we

are capable of recognizing the objects or components in the images with the help of object tracking and object recognition.

7. Ridge Detection: -

This is devised in order to detect the ridges or a range of hill top like structures that are present in the images. This serves as value set of information that can be used for processing, augmentation and use them along with the other methods mentioned above.

There are other features that can be looked into for the feature extraction like height by width ratio, fill ratio, segment area, perimeter, compactness etc. All these features gives more data for training the model and also increase its learning rate there by augmenting the extent of classification.

II. Dataset Description.

The data set that we have chosen is a part of the Kaggle competition and it is selected from the Wikipedia's monkey cladogram. The data set contains different images of monkeys belonging to different species of monkeys under different label names. To be precise the data set contains 10 different monkey species and each is under the label names ranging from n0,n1,n2.....n9 and each label or species set has a set of monkey images (not similar) and these monkey images are to be processed and utilized in the learning and testing activity of the architecture. These images are of 400x300 pixels in dimension or more and they all are of the JPEG format. All these images are downloaded with the help of the "googliser" open source code. There are roughly around 1500 images of monkeys in this data set. The labels are mapped to the monkey species as follows.

N0 - alouatta_palliata
N1 – erythrocebus_patas
N2 – cacajao_calvus
N3 – macaca_fuscata
N4 – cebuella_pygmea
N5 – cebus_capucinus
N6 – micro_argentatus
N7 – saimiri_sciureus
N8 – aotus_nigriceps
N9 – trachypithecus_johnii.

The above given are the species names of monkeys and their respective labels, each label contains distinct monkey images which belong to that particular species only. Each of these images is an RGB image. In order to effectively augment or process the data we need to understand how the machine sees a RGB image (given in the introduction part).

III. Deep Neural Network.

Deep neural network is an artificial neural network that works similar to the human neural network or circuit, in which the neurons pass the information about the data through its smallest units called neurons which take the input and pass the output to the adjacent neuron. In the similar fashion the deep neural network also works by taking inputs and passing the outputs through different layers. There are several layers between the input and the output layer depending upon the model that is being used. Here the networks moves from the input to the output layers through the layers in between by calculating the probability of each output. The layers between the input and the output layers is known as the hidden layers and the number of hidden layers also effects the architecture of the model. The deep neural network is basically feed forward neural network that basically flows in only one direction that is from input to the output. Each layer has artificial neurons also called as perceptrons and these are key components to the neural networks and their count also effects the model. They are assigned with random number or values or weights when connected in between the layers. These perceptrons constitute to the multi-layer perceptron and each layer except the input layer uses the nodes or neurons of each layer use the nonlinear functions as the activation functions.

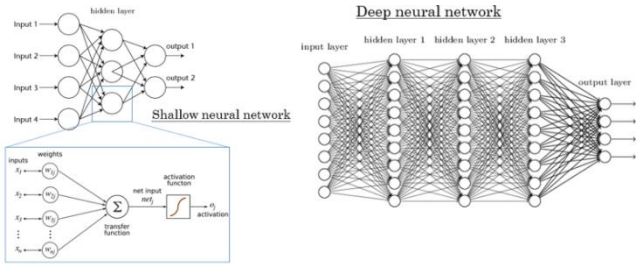

Fig 3: Deep neural network pictorial representation

What is activation function?

The activation function of node is used to determine the output of that particular node which is provided with an input or a set of inputs, this output is then moved on to the neuron of the next layer as input or weights which in turn produces an output of its own and then the cycle is repeated throughout the layers 5that are present till the desired output is achieved.

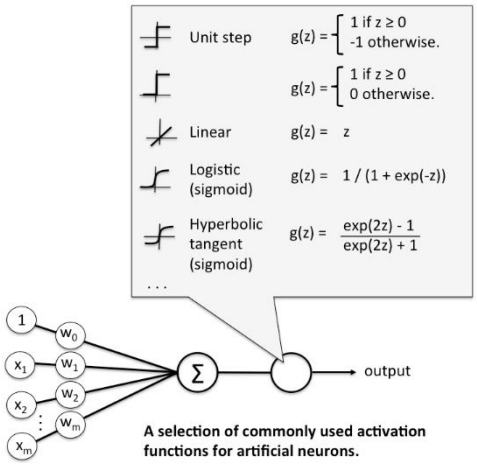

Fig 4: Some commonly used activation functions for the artificial neurons

One of such Artificial / deep neural networks class that is specifically used in image classification and recognition is called as the convolutional neural networks also known as the CNN in short. We will be applying image processing and augmentation along with the convolutional neural networks to attain some fine grain image classifications.

The Convolutional neural networks is used for image classification, object recognition and segmentation which is something that is applied in web cams and other devices that work on cameras (web cam). The following image example shows the image recognition, classification and segmentation.

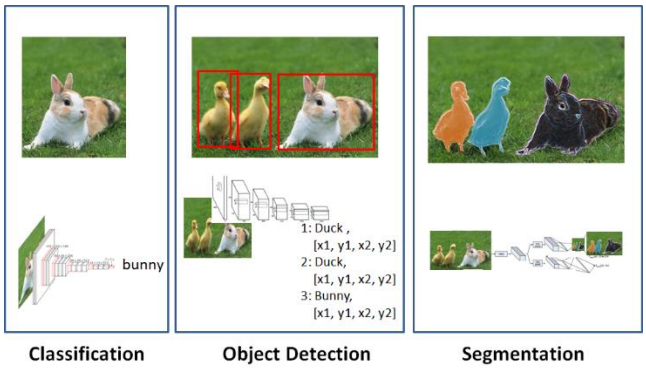

Fig 5: Example showing image recognition, classification and segmentation.

IV. Data Preprocessing.

CNN are analogous to traditional ANNs in what they are comprised of neurons that self-optimize through learning. Each neuron receives an input and perform an operation which could be a scalar product followed by a non-linear function [4]. One of the notable differences between CNN and traditional ANN is that CNN's are primarily used in the field of pattern recognition within images. This allows us to encode image-specific features into the architecture, making the network more suited for image focused tasks –whilst further reducing the parameters required to set up a model. Building an effective neural network model requires careful consideration of the network architecture as well as the input data formats or parameters the input parameters used for this project included the image height, image width, number of channels and also the number of levels per pixel. Typically, we have 3 main data corresponding to the colors Red, Green, Blue (RGB) Pixel levels are usually [0,255]. There were several pre-processing steps that were carried out before completing the project and they are highlighted below.

A. Uniform Aspect Ratio

One of the first steps was to ensure that images have the same size and aspect ratio. Most of the neural network models assume a square shape input imaged, which simply means that each image needs to be checked if it's a square or not and cropped appropriately. Cropping can be done to select a square. At first our images had dimensions that ranged from 250 x 250pixels, but we had to ensure they were evenly distributed by making all the images 150 x 150 pixels

B. Image Scaling

Once we ensured that all the images were squared up, we decided to do some scaling, we needed to scale the width and height of each image by a factor of 0.4. The library function used to perform this task was ***sckitimage.*** Sckitimage is a collection algorithm for image processing and is free to use without restriction by anyone.

C. Dimensionality Reduction

We were able to collapse the RGB into a single gray scale. This enabled the neural network to be invariant to the existing dimension, hence making the problem more tractable. Dimension reduction can be further divided into feature selection and feature projection. The ***Feature selection*** part tries to find a subset of the original variables, in some cases, data analysis such as regression or classification can also be done in the reduced space than in the original space, whereas the ***Feature projection*** part transforms the data in the high dimension space to a space of fewer dimensions. For multidimensional data. Tensor representation can be used in dimensionality reduction through multilinear subspace learning.

D. Data Augmentation

One of the most important preprocessing techniques used on our dataset was this augmentation technique. Data augmentation involves supplementing the existing data-set with irregular versions of the existing images. This is done to expose the neural network to a vast variation, which makes it less likely that the neural network recognizes unwanted characteristics in the dataset. We had to make a few variations from the normal images using the ***sckitimage*** library and the ***imgaug*** library in ***Python.*** Data augmentation aids in increasing the content of the data that is used for processing and training the convolutional neural network, it is similar to the way a human being learn to recognize and identify things that's is by reputedly experiencing the situation and in this case the objects or the images and memorizing them. So more the data that is used for learning the better is the learning process and this image augmentation aids the CNN in heightened learning rate.

Fig 6.0: A Dark augmentation feature

Fig 6.1: Normal Image

Fig 6.2: Inverted Image augmentation feature with light

Fig 6.3: Regular Inverted feature

Fig 6.4 Saturated Image

Fig 6.5 Flipped Image

V. Convolution Neural Network Applied.

CNNs are state-of-the-art deep learning model which have delivered exceptional results in image processing tasks. They use multilayer perceptron's designed to require minimum preprocessing of input data. They take care of feature engineering internally and deliver very good results by learning from all the features. The Efficiency and competency of convolutional neural networks in recognizing the images is one of the reasons why the world has looked up to the efficiency of the deep learning. The word convolution is derived from the Latin word "*convolvere*", which means "*to concolve*", a convolution is nothing but the integrated measure of to which extent any two functions overlap as one function moves over the other function [10].Following is architecture on which CNN models are designed:

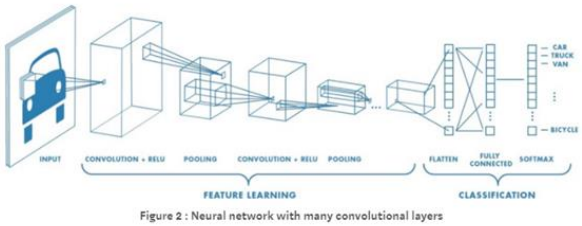

Figure 2 : Neural network with many convolutional layers.

Fig 7: Neural Network with many convolutional layers.

A. *Convolution Layer*

Convolution layer is the most important component of CNN network. This layer creates a feature map of input data using features of image in form of pixel values. Feature map helps the network to learn about the relationship between image pixels which helps network predict context of image. Feature map is produced by filters present in the Convolution layer. A filter will move on input image (in strides) and perform mathematical operation. Filter will consist of random weights which will be tuned using backpropagation during network training. We use Non-Linear activation function in CNN layer. ***Relu*** is most widely used such activation function. Following is sample snapshot of CNN operation to produce feature map:

Figure 4: Image matrix multiplies kernel or filter matrix
Figure 5: 3 x 3 Output matrix

Fig 8: CNN operation to produce feature map.

B. *Pooling Layer*

This layer is used to down-sample the data to decrease the computational complexity of the model. Max Pooling is most import pooling layer which is used in CNN. We have used Max pooling layer in our model after every convolution layer. Pooling layer will also use filter (filter will move on input 2-D feature set in strides) which will perform respective active according to layer type used. Max pooling will take maximum value from values under filter and takes maximum value from them as output. Following is screenshot of sample Max pooling operation:

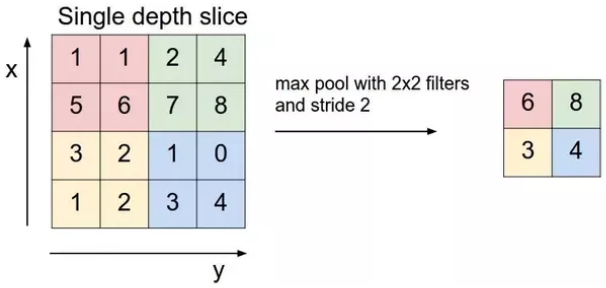

Figure 8 : Max Pooling

Fig 9: A example figure showing max pooling.

C. *Fully Connected Dense Layer*

This layer will be used to predict output classes with the help of SoftMax activation function. SoftMax activation function converts input features into probabilities of possible out classes. Output Dense layer will have same neuron count as number of output labels/classes.

D. *Flatten Layer*

Flatten layer convert 2-D features received after convolutional operation (Convolution Layer + Pooling Layer) into 1-D features. This 1-D features will be used by fully connected dense layers to predict output classes.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 148, 148, 32)	896
activation_1 (Activation)	(None, 148, 148, 32)	0
max_pooling2d_1 (MaxPooling2	(None, 74, 74, 32)	0
conv2d_2 (Conv2D)	(None, 72, 72, 32)	9248
activation_2 (Activation)	(None, 72, 72, 32)	0
max_pooling2d_2 (MaxPooling2	(None, 36, 36, 32)	0
conv2d_3 (Conv2D)	(None, 34, 34, 64)	18496
activation_3 (Activation)	(None, 34, 34, 64)	0
max_pooling2d_3 (MaxPooling2	(None, 17, 17, 64)	0
dropout_1 (Dropout)	(None, 17, 17, 64)	0
flatten_1 (Flatten)	(None, 18496)	0
dense_1 (Dense)	(None, 512)	9470464
activation_4 (Activation)	(None, 512)	0
dropout_2 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 10)	5130
activation_5 (Activation)	(None, 10)	0

Fig 10: Snap shot showing the Model Architecture Used for the project

We used 3 Convolution Layers followed by Max pooling operations. Flatten Layer has been used to manipulate features from 2-D to 1-D form. 2 Dense Layers has been used with output layer having SoftMax Activation function to classify data into 10 classes. Following are hypermeters used to tune our model during training:

1. Filter Count
2. Filter Size
3. Input Image Data Dimensions
4. Epochs

To avoid the problem of overfitting, we have used Dropout as the regularization technique.

VI. Results

A. Model Training

We have trained our model with preprocessed training images with 200 epochs. After model optimization by tuning different hyper parameters, we achieved ~93% training accuracy and ~95% validation accuracy. Various network hyper parameters were tuned during model training to improve model performance. Adam optimizer has been used for model optimization. Following is the training and validation accuracy plot with respect to Epoch count for the finalized model is shown:

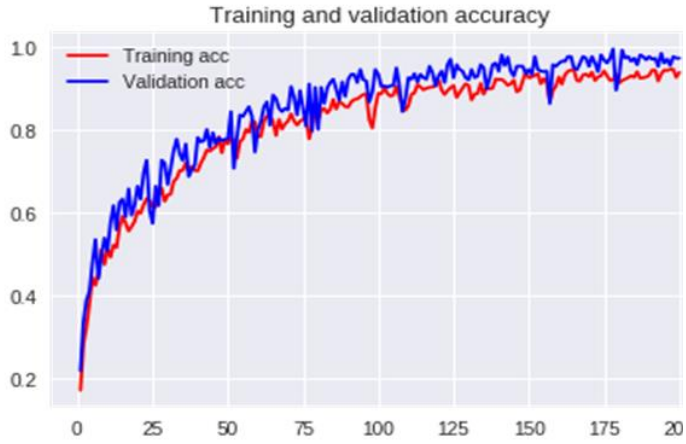

Fig 11: A graph showing training and validation accuracy of the model

B. Model Testing

Since Accuracy can be a misleading parameter sometimes, we have plotted confusion matrix for unprocessed raw testing images, and we were able to get good results. We were able to get a **Precision score of 0.81** for testing data. Following is plotted confusion matrix that clearly shows the way each species are correlated during classification and also the accuracy with which the classification is being done and also the extent of miss classification in total and also in respect to each species of the monkey is shown using each grid of the confusion matrix.

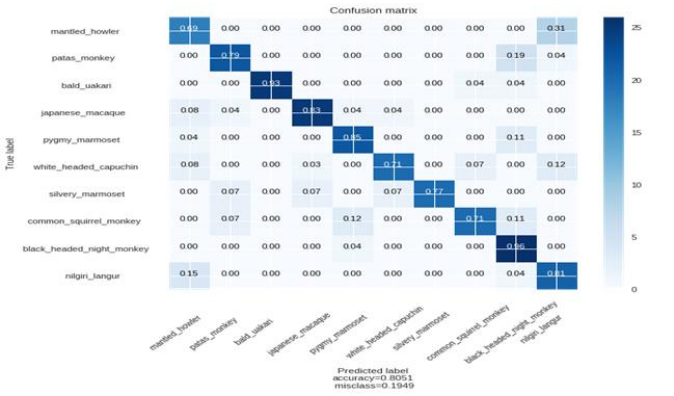

Fig 12: confusion matrix of the model used.

VII. Real World Application

This model or architecture of neural networks combines with the fine grain image processing and augmentation aided by modern day robotics can be used for identifying and rescuing and maintaining species of animals that are endangered to be extinct in the near future or it can also be applied and used to identify and rescue people and wild life from calamity struck areas with ease and speed. It can be used to maintaining zoological parks and wild life sanctuaries. They can be used in the field of cyber security and fortify it in this aspect. It can also be applied to recognize human emotions which can be of a great asset to artificial intelligence and humanoid robotics. The example of this human emotion recognition is clearly shown in this YouTube link

https://www.youtube.com/watch?v=Hs_FeiB7tSQ

This image classification project aided with real world developing technologies and has the potential to unlock great utilities with a little bit of extensive study.

VIII. Appendix

The following is the link for the bit bucket repository for the code: -

https://bitbucket.org/kamalmann07/ece9039b_group17/src/master/

Work Division among the team: -

1. Dharanikota Rajendra Kamal –
   Image processing and image augmentation, feature exploration.
2. Emmanuel Maduwuba –
   Data preprocessing, feature extraction, model optimization (partly)
3. Kamaljeet Singh Mann –
   model optimization and Model fitting, results and discussion.

IX. References

[1] Pratap Dangeti “Introduction to Deep Learning” in Statistics for Machine Learning, Birmingham, UK: Packt Publishing, 2017 Ch. 6, pp. 261–264.

[2] B.Nikhil, *ImageDataPreprocessingforNeural Networks, 2017.* [Online].

Available: https://becominghuman.ai/image-data-pre-processing-for-neural-networks-498289068258. [Accessed 12th April 2019]

[3] W.-K. Chen, *Linear Networks and Systems.* Belmont, CA: Wadsworth, 1993, pp. 123–135.

[4] Keiron O'shea and Ryan Nash , "*An Introduction to Convolution Neural Networks*", School of Computing and Communications, Lancaster University, Lancashire, p3

[5] Pedro Domingos ," *A Few Useful Things To Know about Machine Learning"*, Department of Computer Science and Engineering , University of Washington, Seattle , USA, 2012,pp 3-5

[6] https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f1488

[7] https://ieeexplore.ieee.org/document/4409058

[8] https://en.wikipedia.org/wiki/Main_Page

[9] https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[10] https://skymind.ai/wiki/convolutional-network

[11] http://cs231n.github.io/convolutional-networks/\

[12] https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

[13] https://github.com/aleju/imgaug

[14] https://imgaug.readthedocs.io/en/latest/source/jupyter_notebooks.html