Project Report

Detection of Dog Breeds

A ML and Deep Learning Approach

Project Report submitted on the fulfillment of the requirements of Post Graduate Diploma in Big data Analytics

Authors:		Co-ordinators:
Ms. Varsha Patil	(220960925032)	Ms. Roopa Panicker
Mr. Ketan Saptasagare	(220960925041)	Ms. Divya Das
Mr. Sarvesh Mayekar	(220960925042)	Ms. Soorya M.
Mr. Sajid Shaikh	(220960925043)	
Mr. Abrar Shaikh	(220960925044)	

STDC

CDAC Thiruvananthapuram

Trivandrum, Kerala 695581

Table of Contents

I. Abstract 3
II. Introduction 4
III. Literature Survey 5
IV. Proposed System 9
A. Dataset pre-processing 10
B. Feature Extraction 11
C. Model used 12
V. Discussion and Results 13
A. Output obtained 14
B. Evaluation measures used 14
VI. Conclusion 16
VII. References 17
Dataset Reference: 17

I. Abstract

With growing concerns over climate change and global warming, it falls on us humans to take care of other species. So, it is important to identify different breeds particularly to have a deeper understanding of the concerns over health, natural instinct and their behavior. Keeping this in mind, we are going to create a model to identify dog breeds as accurately as possible. This will also contribute to the Characteristic registry.

II. Introduction

Machine learning (ML) is a subset of artificial intelligence that involves teaching computers to learn from data without being explicitly programmed. The goal of machine learning is to develop algorithms that can identify patterns in data and make predictions or decisions based on those patterns.

Deep learning is a subset of machine learning that involves training neural networks with many layers to recognize patterns in data. These networks are modeled after the structure of the human brain, with layers of interconnected nodes that learn to recognize different features of the input data. Deep learning has shown great success in a wide range of applications, including computer vision, natural language processing, and speech recognition.

Image recognition and type have efficaciously implemented in various domain names, inclusive of face recognition and scene information for self reliant driving. At present, human face identity is efficiently used for authentication and security purposes in lots of packages. Consequently, there are efforts to increase studies from human to animal reputation. Specifically, dogs are one of the most common animals. On account that there are more than 180 dog breeds, dog breed recognition may be an important task for you to offer the right training and health remedies.

For the purpose of our project, we decided to choose the Stanford Dogs Dataset. The Stanford Dogs dataset contains images of 120 breeds of dogs from around the world. This dataset has been built using images and annotations from ImageNet for the task of fine-grained image categorization. Contents of this dataset include 120 breeds of dogs and about 20,580 images in total. There are approximately 150 images per breed.

III. Literature Survey

The main principle is to implement Image classification with Deep Learning and Convolution Neural Networks using Tensor flow. The main focus is on a machine learning model which classifies the breed of the dog from an image. The methods used to solve this problem would also help identify breeds of cats and horses as well as species of birds and plants - or even models of cars. Any set of classes with relatively small variation within it can be solved as a fine-grained classification problem. Since there are more than 180 dog breeds, dog breed recognition can be an essential task in order to provide proper training and health treatment. Previously, dog breed recognition was done by human experts. However, some dog breeds might be challenging to evaluate due to the lack of experts and the difficulty of breeds' patterns themselves.

In this experiment we are using Convolutional Neural Network: A special type of Neural Networks that works in the same way of a regular neural network except that it has a convolution layer at the beginning. Instead of feeding the entire image as an array of numbers, the image is broken up into a number of tiles, the machine then tries to predict what each tile is. Finally, the computer tries to predict what's in the picture based on the prediction of all the tiles. This allows the computer to parallelize the operations and detect the object regardless of where it is located in the image.

The image classification task is mainly divided into the following 3 steps:

1. Image Pre-processing
2. Feature extraction and training
3. Classification of the Object

Image Pre-processing:

Before being used for model training and inference, pictures must first undergo image preprocessing. Data preprocessing is a step in the data mining and data analysis process that takes raw data and transforms it into a format that can be understood and analyzed by computers and machine learning. Machines like to process nice and tidy information – they read data as 1s and 0s. So calculating structured data, like whole numbers and percentages is easy. However, unstructured data, in the form of text and images must first be cleaned and formatted before analysis.

The phrase that is quite common among ML professionals is “garbage in, garbage out”. This means that if you use bad or “dirty” data to train your model, you’ll end up with a bad, improperly trained model that won’t actually be relevant to your analysis.

Figure 1 Data that is not pre-processed

To prepare picture data for model input, preprocessing is necessary. For instance, convolutional neural networks' fully connected layers demanded that all the images be in arrays of the same size. Additionally, model preprocessing may shorten model training time and speed up model inference.

Feature Extraction:

The most crucial part of our dog breed detection is the feature extraction process. With over decades of research available on the internet, there are a few major algorithms that are frequently used. SIFT (*Scale Invariant Feature Transform*), RGB histogram, Colour Centers Histogram are some of the examples.

SIFT, or Scale Invariant Feature Transform, is a feature detection algorithm in Computer Vision. SIFT helps locate the local features in an image, commonly known as the ‘key points ‘of the image. These key points are scale & rotation invariants that can be used for various computer vision applications, like image matching, object detection, scene detection, etc. We can also use the key points generated using SIFT as features for the image during model training. The major advantage of SIFT features, over-edge features, or hog features, is that they are not affected by the size or orientation of the image.

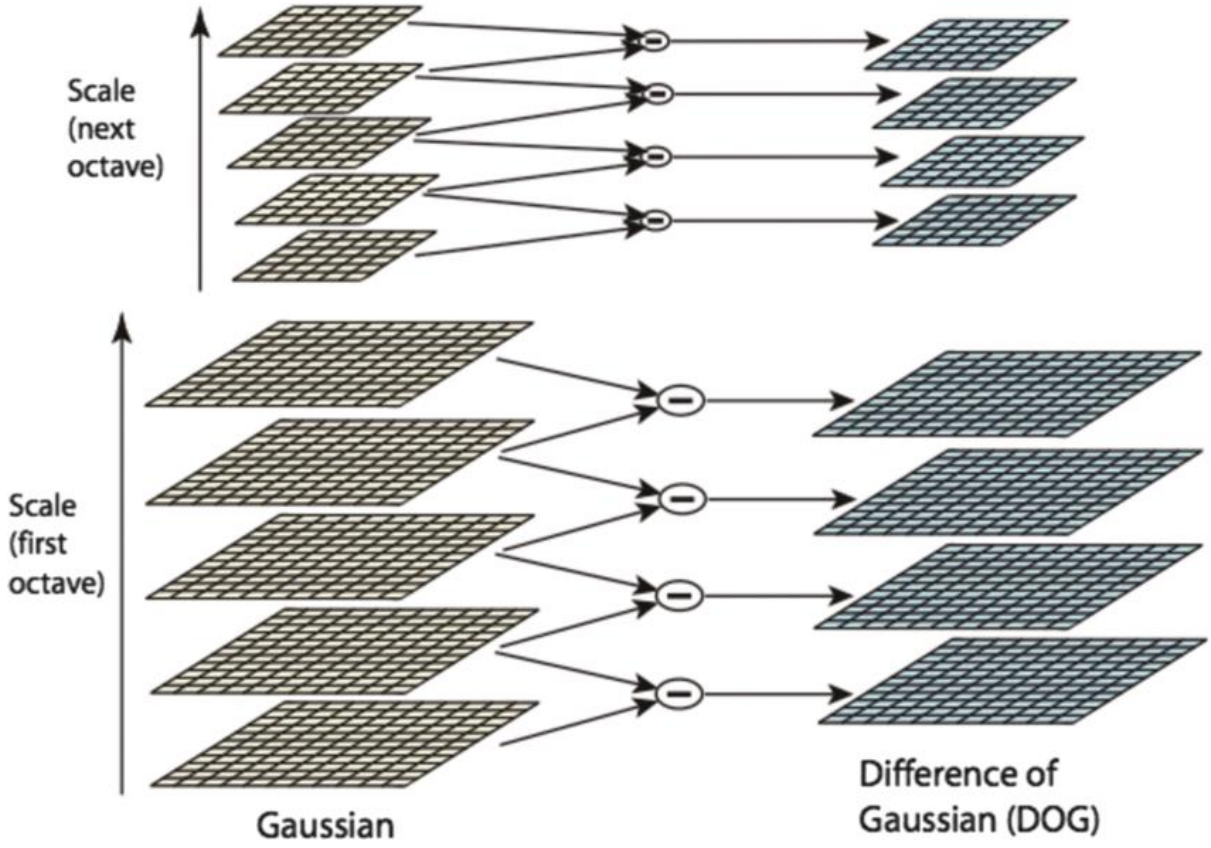


Figure 2 SIFT Scaling

The entire process can be divided into 4 parts:

- Constructing a Scale Space: To make sure that features are scale-independent
- Key point Localization: Identifying the suitable features or key points
- Orientation Assignment: Ensure the key points are rotation invariant
- Key point Descriptor: Assign a unique fingerprint to each key point

The RGB histogram is a compact representation of the color distribution of an image that can be used for a wide range of computer vision tasks such as object recognition, image retrieval, and content-based image retrieval. The image is first divided into a set of non-overlapping blocks or regions. The pixels in each block are then grouped into bins based on their color values. The number of bins can vary, but it is typically between 8 and 64 for each color channel. For each color channel, the number of pixels in each bin is counted to create a histogram of the color distribution in that channel. The three color channel histograms are then concatenated to form the final RGB histogram.

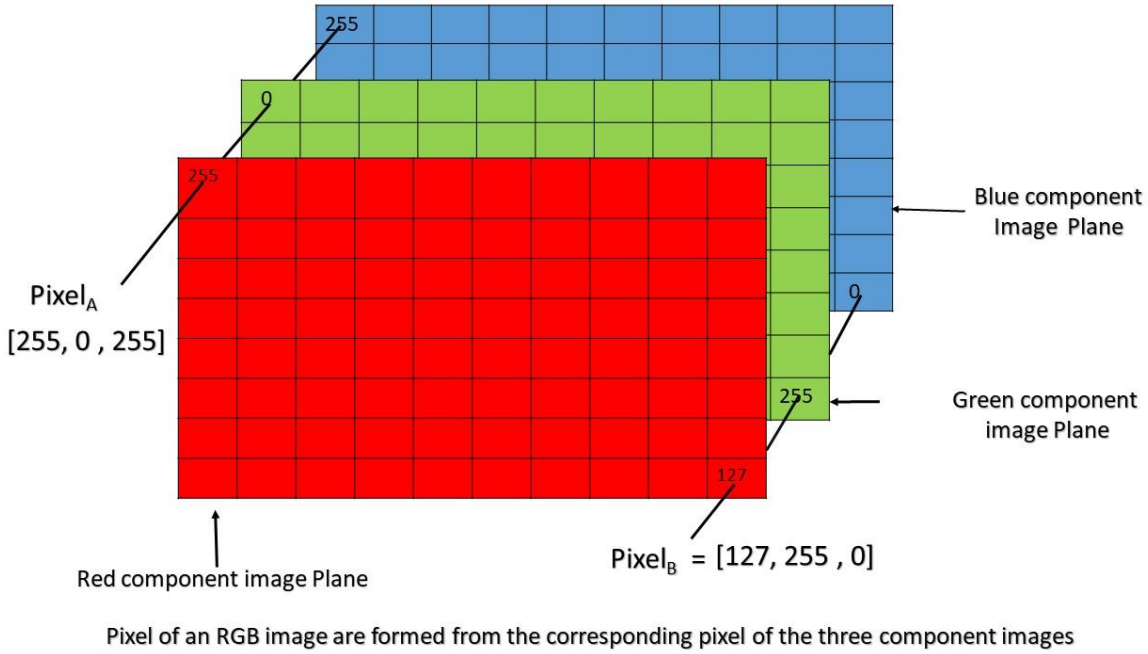


Figure 3 RGB Histogram

For computing color centers, we are having images in RGB color spaces, each pixel in the image is then treated as a point in the color space. The pixels are clustered together based on their color values using a clustering algorithm such as k-means clustering or mean shift clustering. Each cluster represents a group of pixels that have similar color values. Here the k-means algorithm is used to cluster the pixel values of all the training images into 32 clusters, which represent the 32 color centers. Each image in the training set is then represented by a histogram that counts the number of pixels in each cluster. This histogram represents the color distribution of the image. The centroid of each cluster is then computed as the mean color value of all the pixels in the cluster. The resulting color centers can be used as a feature descriptor.

The last step in the image classification task is identification of the object. With the help of extracted features we can train our model to understand them and use it to further classify unknown images.

IV. Proposed System

Developing a CNN from scratch and then training it to fit a large dataset like ours requires heavy resources. And even with the availability of such resources, time is a luxury. So, we propose using a pre-trained model like ResNet50 to maximize our accuracy and minimize the processing time. This method is also known as transfer learning.

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems.

ResNet50 is a deep convolutional neural network architecture that was proposed by Microsoft Research in 2015. The architecture of ResNet50 is based on the concept of residual learning, which allows the network to learn complex mappings between the input and output by using skip connections or shortcuts to bypass one or more layers in the network.

The ResNet50 architecture consists of 50 layers, including convolutional layers, max pooling layers, and fully connected layers. The input image is first passed through a convolutional layer, followed by max pooling layers that reduce the spatial dimension of the feature maps. The convolutional layers are organized into blocks, each of which consists of several convolutional layers, followed by batch normalization and a rectified linear activation function (ReLU).

The key innovation of the ResNet50 architecture is the use of residual blocks, which allow the network to learn residual functions with respect to the input. Each residual block consists of two or more convolutional layers, with a shortcut connection that adds the input to the output of the final convolutional layer. The shortcut connection allows the network to learn the residual function by minimizing the difference between the input and output, rather than directly learning the desired output.

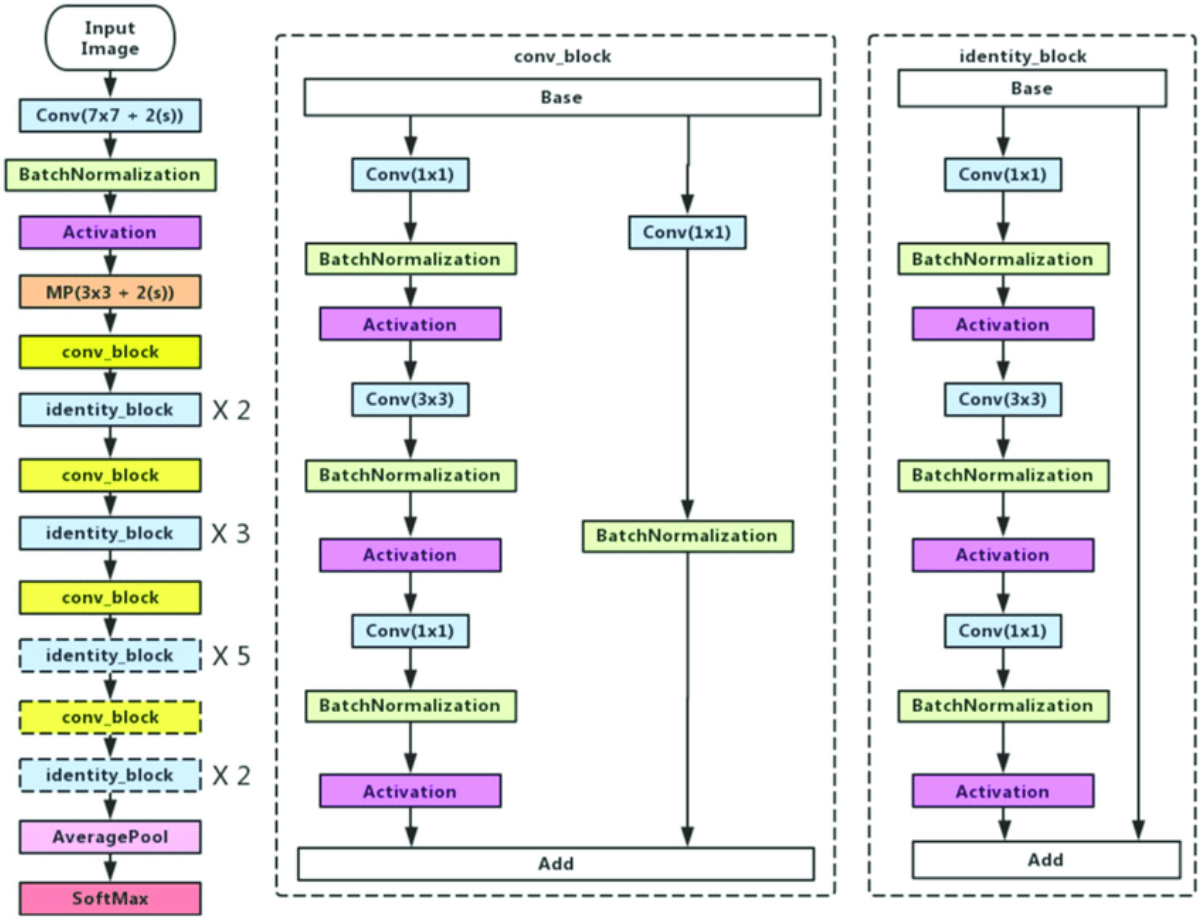


Figure 4 ResNet50 Architecture

The final layers of the network consist of global average pooling, followed by fully connected layers and a softmax activation function that produces the final output probabilities for the classification task. The ResNet50 architecture has been shown to achieve state-of-the-art performance on a range of image classification tasks.

A. Dataset pre-processing

Good data-driven decision making requires good, prepared data. In ResNet50, we use the *preprocess_input* from the tensorflow keras library. It preprocesses a tensor or Numpy array encoding a batch of images.

```
tf.keras.applications.resnet50.preprocess_input(
    x, data_format=None
)
```

Here, *x* is a floating point *numpy.array* or a *tf.Tensor*, 3D or 4D with 3 color channels, with values in the range [0, 255] and *data_format* is the optional data format of the image tensor/array.

We also use *to_categorical* from tensorflow keras library to convert a class vector (integers) to a binary class matrix.

```
tf.keras.utils.to_categorical(
    y, num_classes=None, dtype='float32'
)
```

Here, *y* is an array-like with class values to be converted into a matrix (integers from 0 to *num_classes* - 1). The *num_classes* is the total number of classes. If None, this would be inferred as *max(y) + 1*.

B. Feature Extraction

In a CNN architecture like ResNet50, feature extraction is performed by the convolutional layers. These layers consist of a set of filters that are convolved with the input image to produce feature maps. Each filter responds to a specific pattern or feature in the input image, such as edges, corners, and other visual elements.

ResNet50 has a deep architecture, with 50 layers, including residual connections. The residual connections allow the network to learn residual features, which are the differences between the input and the output of a block of layers. By learning residual features, the network can avoid the problem of vanishing gradients, which can occur in very deep architectures.

In ResNet50, the convolutional layers are organized into blocks, each of which contains a set of convolutional layers, batch normalization layers, and activation layers. The residual connections connect the input of the block to the output of the block, allowing the network to learn residual features.

Overall, the CNN architecture in ResNet50 helps to extract features from images by using a set of convolutional layers with residual connections, which allow the network to learn residual features and avoid the problem of vanishing gradients.

C. Model used

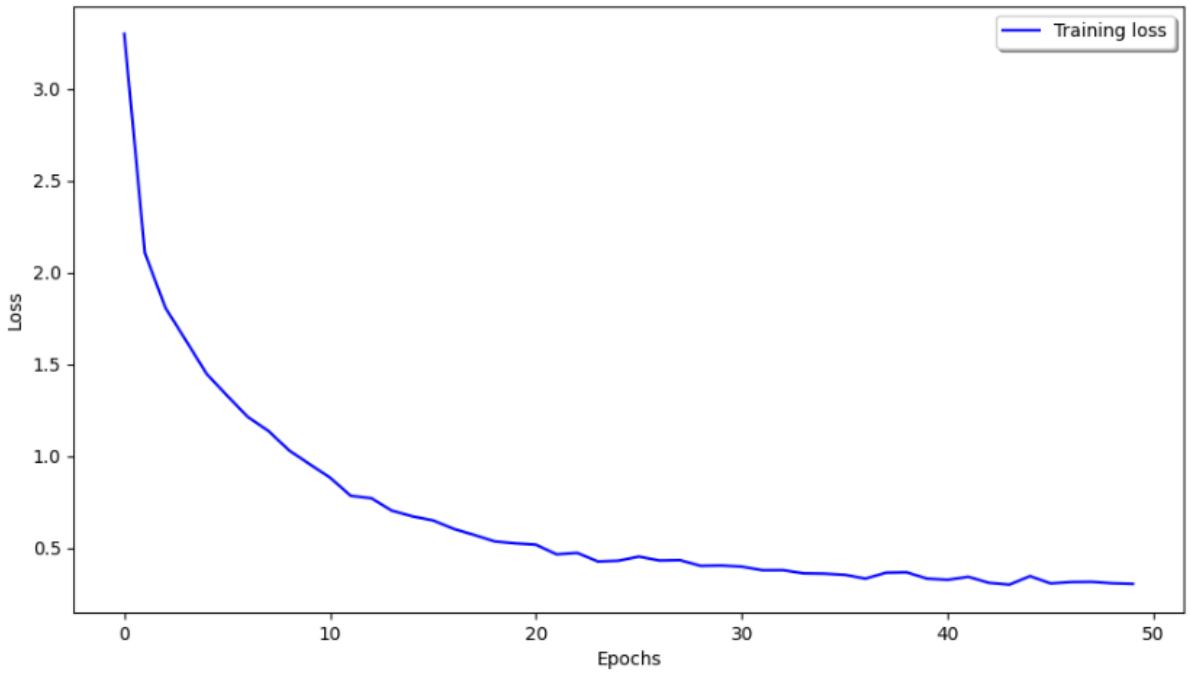
Using various types of layers available to us in tensorflow, we proposed the following model. It has 5 layers including the input and predictions layer. As discussed, we used ResNet50 on the *base_model*. The weights assigned were *imagenet* with the input shape fixed uniformly to 128x128.

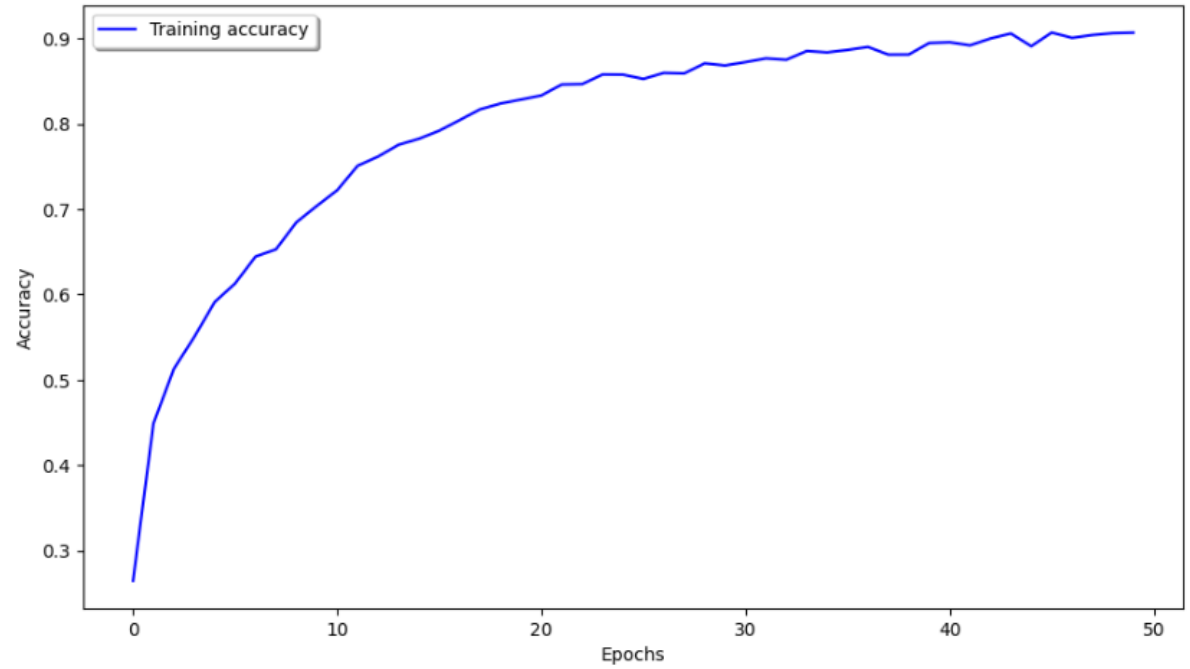
```
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
x = Dropout(0.5)(x)
predictions = Dense(n_classes, activation='softmax')(x)
```

Using *adam* optimizer and *categorical_crossentropy* loss method, the model was trained on 50 epochs and a batch size of 64 images.

V. Discussion and Results

It is normal for a model to have some training loss. While computing our model, we came across a maximum accuracy of 90%. For a model trained on 50 epochs and batch size of 64 images, the plot is as follows:

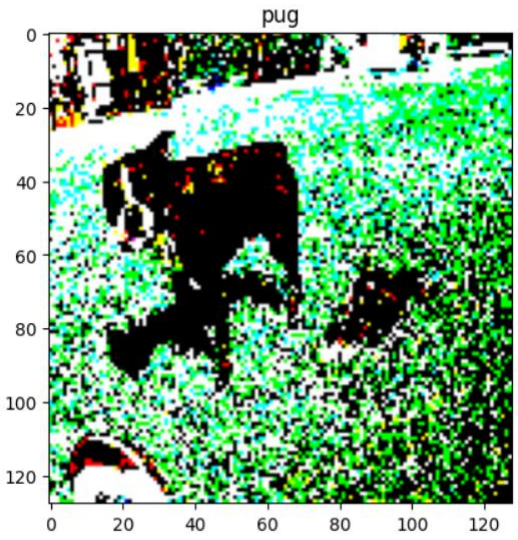


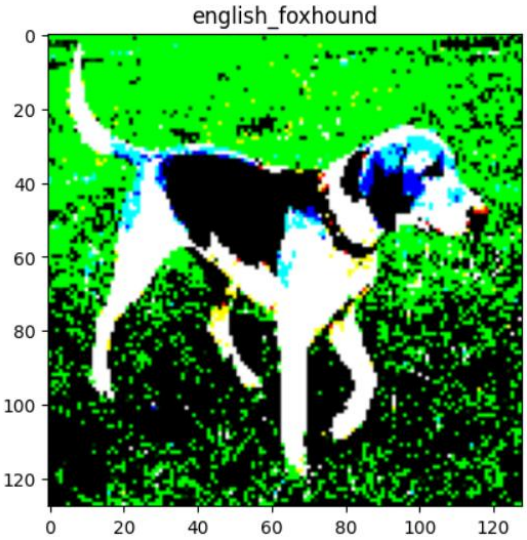


We can clearly see that after 35 epochs, the training loss was stabilized and the accuracy was stable.

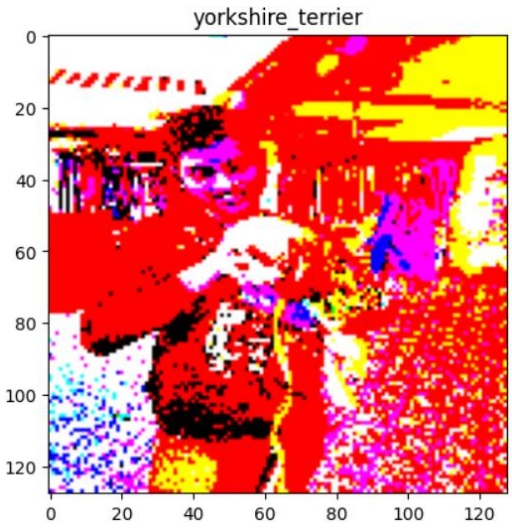
A. Output obtained

Some examples of output are as follows:





Predicting a photo that was uploaded by us:

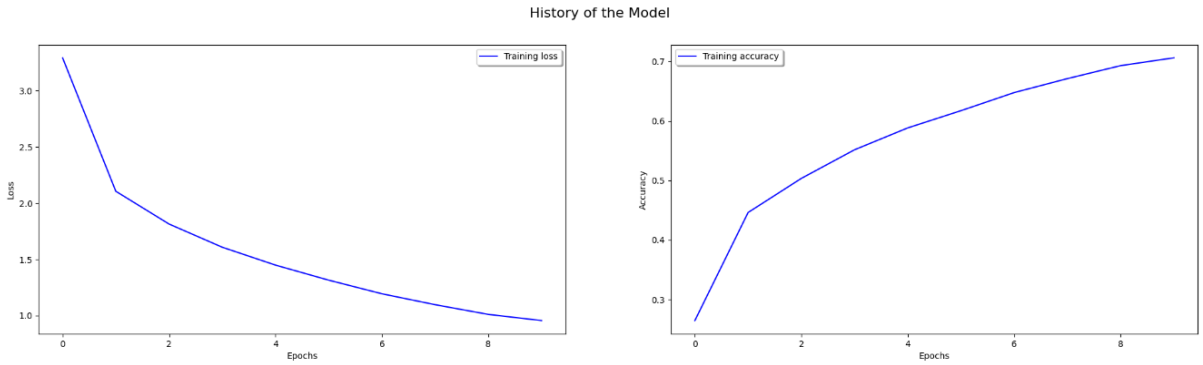


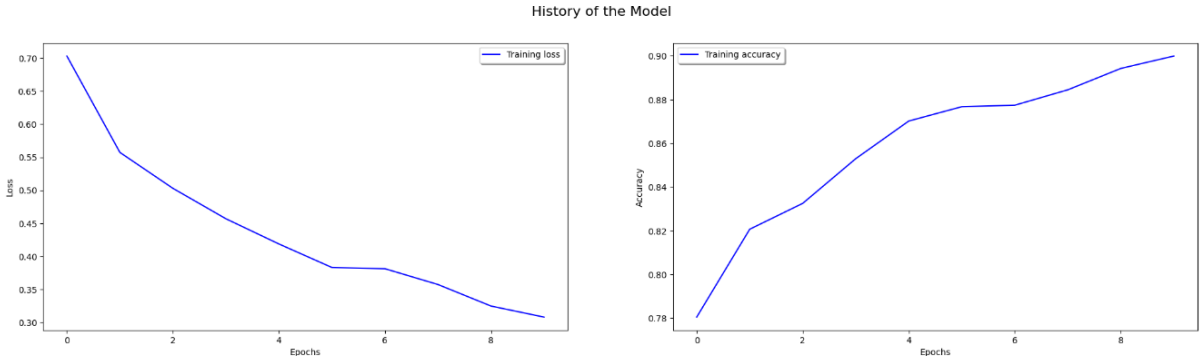
The breed of the dog was correctly predicted by the model.

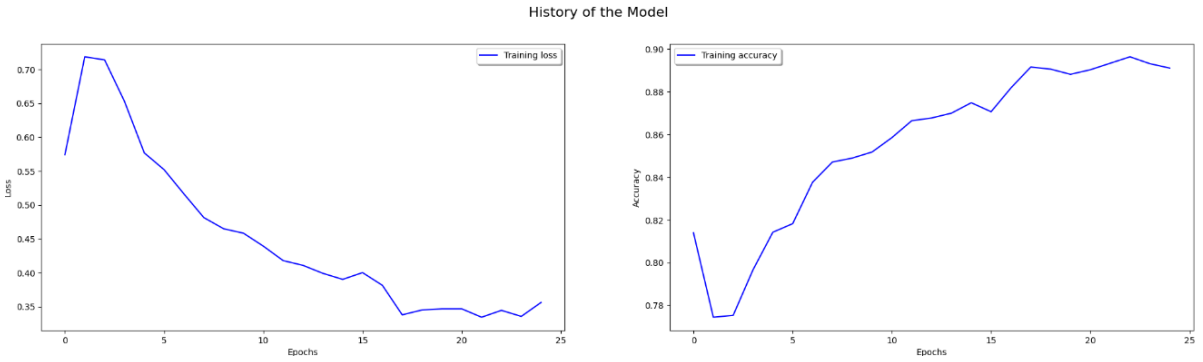
B. Evaluation measures used

For comparison, we also trained a model with different epochs and batch sizes. We used the following 3 variations:

1. 10 Epochs, 64 batch size
2. 10 Epochs, 128 batch size
3. 25 Epochs, 64 batch size







As it is clearly visible, the model with more epochs gives us a better output and better accuracy. When fewer epochs are used, the accuracy is not stabilized.

VI. Conclusion

We can say that the results were a success overall. The goal that we set out to accomplish was carried out successfully to an accuracy of 90%. This model can be used further with other technologies, to improve dog breed detection.

VII. References

1. “Knowing Your Dog Breed: Identifying a Dog Breed with Deep Learning”, Punyanuch Borwarnginn, Worapan Kusakunniran, Sarattha Karnjanapreechakorn, Kittikhun Thongkanchorn. Faculty of Information and Communication Technology, Mahidol University, Nakhon Pathom 73170, Thailand
2. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.
3. https://web.stanford.edu/class/cs231a/prev_projects_2016/output%20(1).pdf
4. https://www.tensorflow.org/api_docs/python/tf
5. Parthesh Haswar, Aditya Iyer, Prachiti Godhane, Amisha Jadhav, Dr. Ankita Malhotra, “Dog Breed Identification and Age Detection using Neural Networks”, MCT Rajiv Gandhi Institute of Technology, Mumbai
6. ResNet50 is a popular convolutional neural network architecture proposed by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun in their paper "Deep Residual Learning for Image Recognition". The paper was presented at the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) and can be accessed here: https://arxiv.org/abs/1512.03385
7. Review Paper on “Dog Breed Classification Using Convolutional Neural Network” Suyash S. B., Rishikesh P. P., Rohit P.W., Kaustubh P. J., Prof. Balaji. Bodke BE, Department of Computer Engineering Modern Education Society, College of Engineering, Pune, India

Dataset Reference:

1. Will Cukierski. (2017). Dog Breed Identification. Kaggle. https://kaggle.com/competitions/dog-breed-identification