Capstone Report

Project Overview

This project uses Convolutional Neural Networks (CNNs)! In this project, you will learn how to build a pipeline to process real-world, user-supplied images. Given an image of a dog, your algorithm will identify an estimate of the canine's breed. If supplied an image of a human, the code will identify the resembling dog breed.

Problem Statement

Write an algorithm that accepts a file path to an image and first determines whether the image contains a human, dog, or neither. Then,

- If a dog is detected in the image, return the predicted breed.
- If a human is detected in the image, return the resembling dog breed.
- If neither is detected in the image, provide output that indicates an error.

Then, deploy algorithm in a flask web application where user load in a dog photo can the application will provide the prediction outcome using the algorithm.

Metrics

Metrics use in the evaluation of the models are accuracy and validation loss. Accuracy of model is defined by the ratio of number correct predictions to total prediction made. A loss function, categorical crossentropy is used in the training which tracks the errors made during the training and learns to minimise the loss function value. The accuracy and loss of the trained models are plotted to monitor for the training process.

Data Exploration

The dataset consists of 8351 dog images with 133 dog categories. The dataset is split into 6680 training images, 835 validation images and 836 test images.

Data Pre-processing

When using TensorFlow as backend, Keras CNNs require a 4D array (which we'll also refer to as a 4D tensor) as input, with shape (samples_size, rows, columns, channels) where corresponds to the total number of images (or samples), and rows, columns and channels correspond to the number of rows, columns, and channels for each image, respectively. Each images in the dataset will need to convert to a 4D tensor suitable for supplying to a Keras CNN. The data will first be resized to 224 X 224 pixels reshaping to shape (1, 224, 224, 3).

Implementation

Implementation steps as follows:

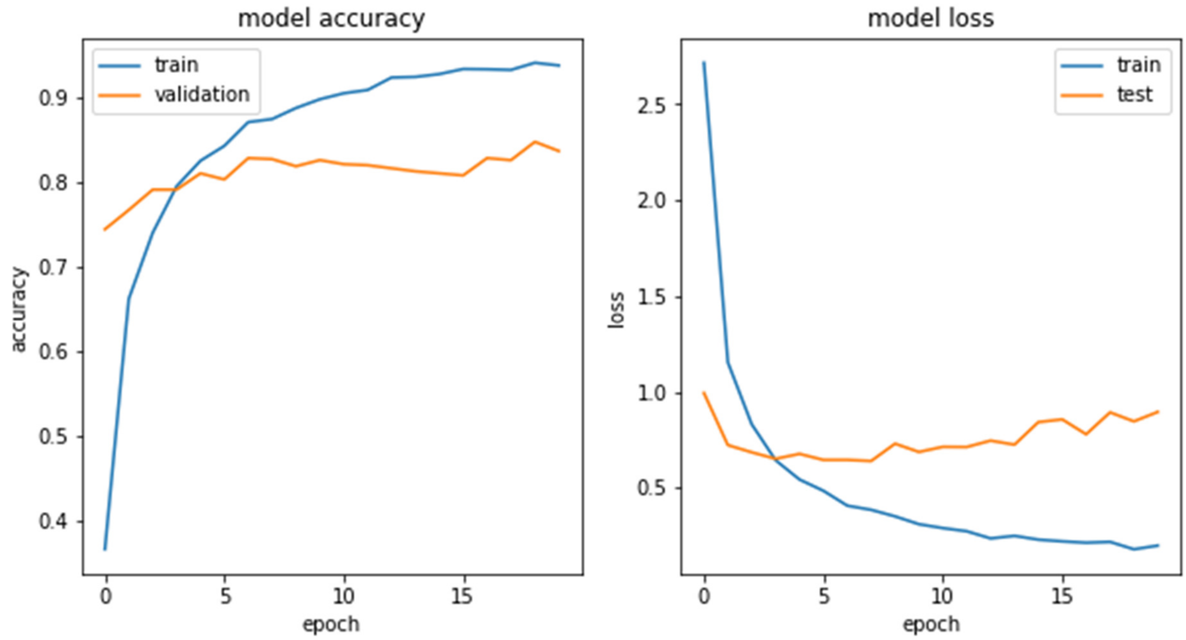
1. Load a face detector from opencv's Haar feature-based cascade classifiers to detect human faces in images.
2. Load a dog detector using pre-trained ResNet-50 model.
3. Use transfer learning to train a pre-trained CNN model to identify dog breed from images to achieve at least 60% accuracy on the test set.
4. VGG19, ResNet-50 and Inception CNN pre-trained networks are evaluated.
5. Then select the better accuracy for development of algorithm model. Construct convolutional model architecture:
    a. Uses the pre-trained model as fixed feature extractor
    b. Add fully connected layer with 256 neurons, using 'relu' activation function.
    c. Add a dropout for regularization
    d. Add fully connected layer with 133 neurons with a softmax function.
6. Compile the model using loss function (categorical_crossentropy), optimizer (rmsprop).
7. Train the model and using model checkpointing to save the model that attains the best validation loss.
8. Plot the model accuracy/loss against the train epochs to evaluate result.
9. Make improvement in the convolutional network to improve the model.
10. Test the model to achieve accuracy of greater than 60%.
11. Write a function to take in an image path and return the dog breed following the steps:
    a. Feature extract the image using the pre-trained CNN model.
    b. Input the features into the function to return the predicted vector and argmax index
    c. Return the predicted dog breed.
12. Finally use the function to write an algorithm to accept an image then,
    a. If a dog is detected in the image, return the predicted breed.

b. If a human is detected in the image, return the resembling dog breed.
c. If neither is detected in the image, provide output that indicates an error.

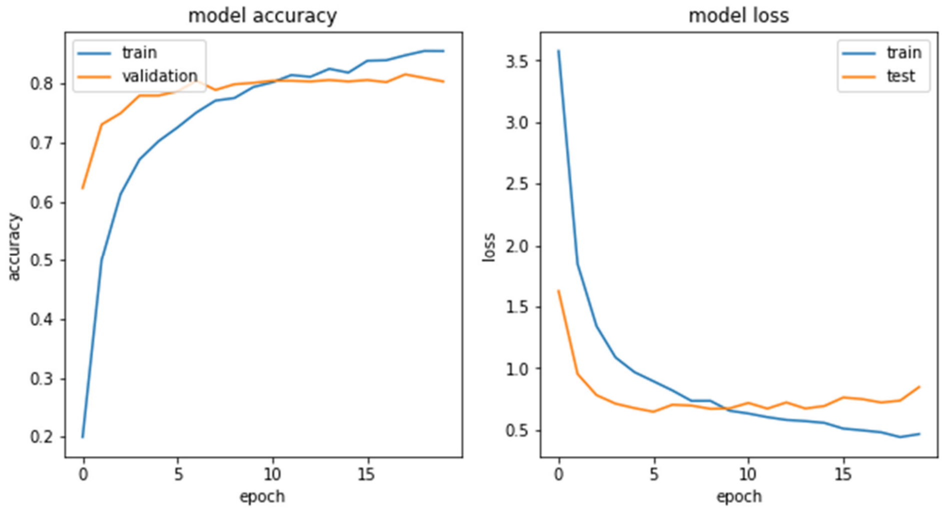
Refinement

Then initial convolutional network was found to be over-fitting. The model architecture was refined by reduce the complexity of the dense layer and also introduce a dropout rate to improve the model.

Before refinement:



After refinement:



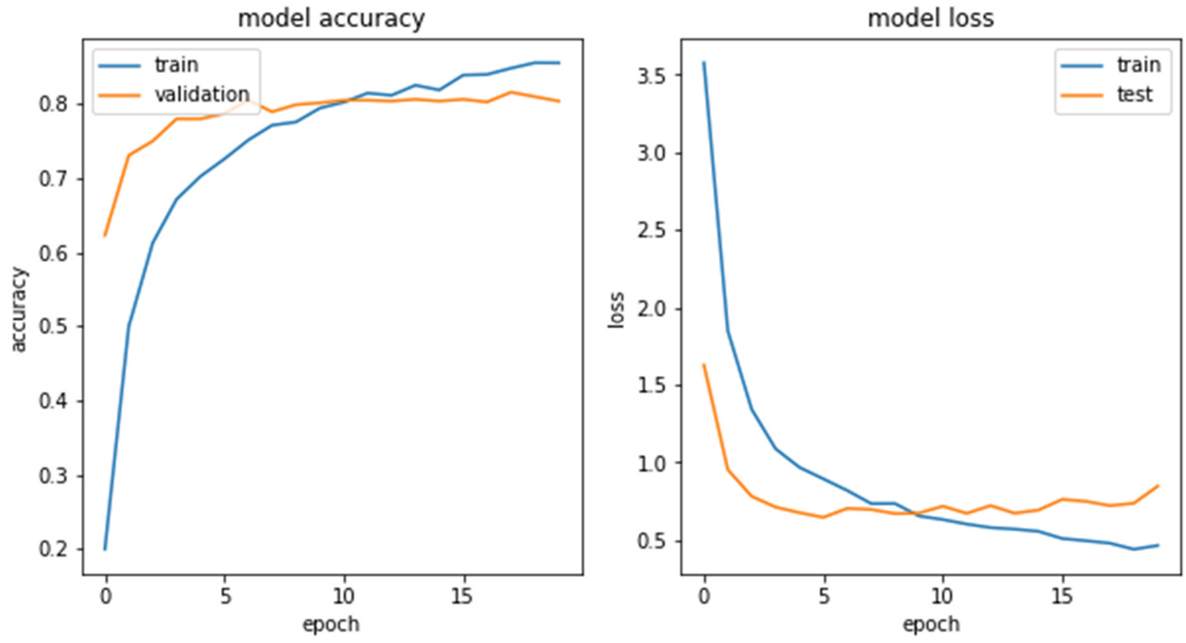
Model Evaluation and Validation

Preliminary Model Evaluation of pre-trained model

- VGG19: 45.5742%
- Inception: 79.4258%
- ResNet50: 81.6986%

RestNet50 network was selected for further development.

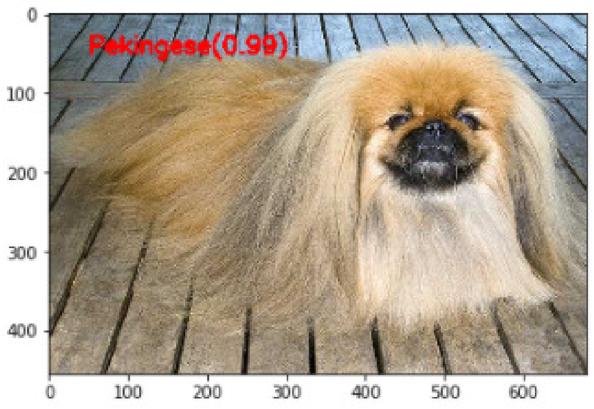
Final model accuracy of 78.3% was achieved with the following model accuracy/ loss plot.



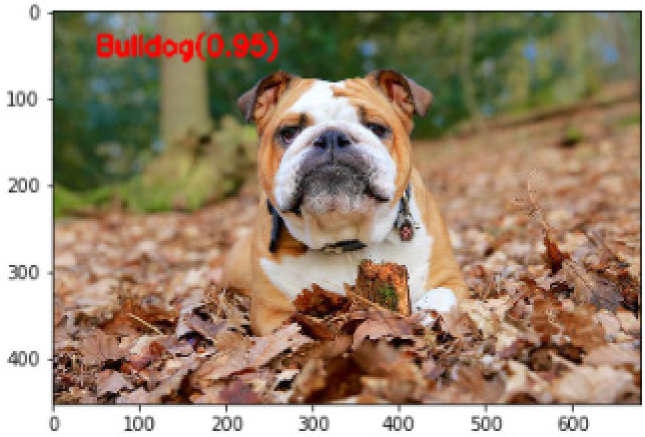
The model was validated on sample dog images to return the predicted dog breed with confidence rate.

1. Prediction return with dog images:

```
This photo looks like a dog of breed Pekingese.
```

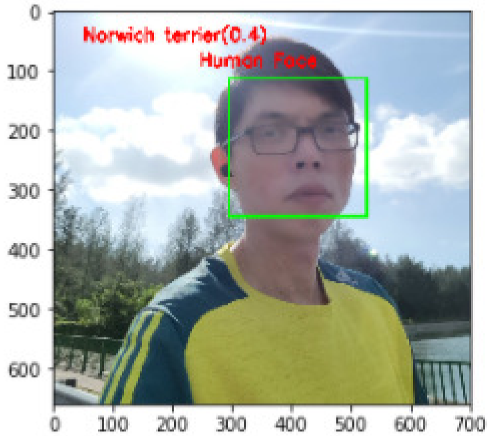


```
This photo looks like a dog of breed Bulldog.
```



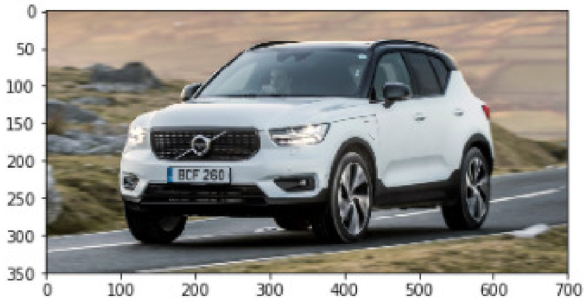
2. Prediction return with human image:

```
This photo looks like a human and resembles a Norwich terrier.
```



3. Prediction return with neither human nor dog:

```
Opps this photo looks like neither human nor dog.
```



Justifications

1. The algorithm output the result of the prediction as well as an output image with printed dog breed with its confidence rate for better user interaction and allow user have more insight of the confidence level of the prediction.
2. One of the dog image is Shih Tzu (not in the trained classes) and is misinterpreted as a Maltese.
3. The algorithm is improved by trying to filter predicted dog breed of low confidence rate that is likely to be misclassified. It was done by setting a threshold to the confidence rate of prediction and if a dog image is identified has a low confidence rate in predicting the breed of the dog, the algorithm will return unknown breed of the dog.

After modification output in point 2 to rectify missing classes.

```
This photo looks a dog but we do not know the breed.
```

Conclusion & Improvement

1. The algorithm is performing as expected to the accuracy of the model of about 78%. Out of the 8 tested dog images, 6 were predicted correctly with confidence rate of more than 70%.
2. One of the dog image is Shih Tzu (not in the trained classes) and is misinterpreted as a Maltese. Also a Chihuahua is misclassified as a German pinscher.
3. For the case of misclassification of Shih Tzu, it would improve the algorithm with more training data of more dog breeds.
4. For the case of misclassification of pinscher could be due to dogs which looks similar but of different size. Would improve the algorithm if there is a way to interpret dog size from the image.