CNN Project: Dog Breed Classification

Domain Background

There are different canine breeds in the world humans wish to identify. With a large range of variety of breeds, it is difficult to point out what the breed of the dog. With machine looking and finding pattern of images of dogs and finding the commonality of the different breeds there are, we will be able to identify the breed having to input an image of a dog. Deep learning is a known technology in machine learning when finding patterns, especially on images. Face detection for instance is a trend on smart phones and security. Convolution Neural Network is a deep learning algorithm which assigns importance to various aspects in the images to differentiate it to another image with enough training ConvNets can learn the characteristics images. ConvNets architecture is patterned to the neurons on human brain. Each neuron corresponds to a stimulus which is a restricted region of the visual field. A collection of visual fields overlaps to cover the entire visual area or the image. Which is why CNN is effective on image classification.

Problem Statement

The goal is for to build an algorithm will identify an estimate of the canine's breed. Assigning breeds to dogs is challenging that even humans have a hard time distinguishing dogs on the same family. For instance, a Brittany and a Welsh Springer Spaniel.

Brittany	Welsh Springer Spaniel
	

These dogs that have a minimal inter-class variation is difficult to identify but we want to train our model to be able to identify the difference. Moreover, Labradors for example have different colors yellow, black, or brown.

Datasets and Inputs

- There are 13233 total human images and 8351 total dog images provided by Udacity.
- The input of the algorithm is an image.
- 1/10 of the total dog images are to be used for testing, another 1/10 to be used for valid data, and 8/10 to be used for training.
- All training datasets will be transformed to be randomly rotated and resized.
- All datasets are resized to fit the model.

Solution Statement

The proposed solution to this problem is to build a model using Convolutional Neural Network (CNN). CNN or ConvNet is an artificial neural network that is popular for classifying images. This algorithm has a special feature that can detect pattern and make sense of these patterns. Its hidden layer called convolutional layer receives input and transforms these to the output layer called convolutional operation.

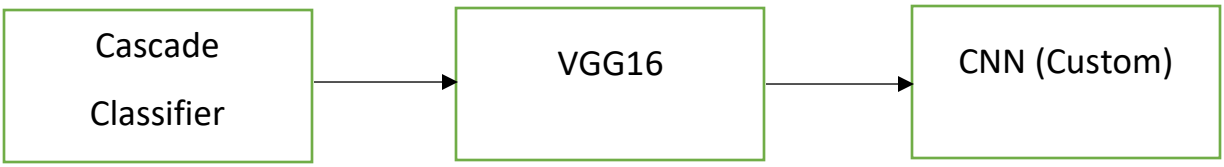

Figure A: Algorithm

Figure A shows the steps the data will go through. First is to detect humans on the picture using the OpenCV's implementation of Haar feature-based cascade classifiers. Then we check if there exists a dog in the image using Pre-trained VGG-16 Model. Lastly, the image will be processed by CNN to identify the breed of the dog.

Benchmark Model

1. Cascade Classifier
    a. An approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.
2. VGG-16
    a. A CNN model that makes an improvement of AlexNet by replacing large kernel sized filters with multiple 3x3 kernel sized filters one after another.

Evaluation Metrics

The metrics to be used to check if the model performed well or not is Cross Entropy Loss which is the uncertainty of prediction based on how much it varies from the actual label. Since there is an imbalance of output classes for this data, so using accuracy score is not a good idea. Although accuracy is shown in the model evaluation which is computed as follow:

$$accuracy = \frac{number\ of\ correctly\ classified}{number\ of\ classified\ items}$$

Instead using F-score to measure the model performance which is the weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal. It can be computed as follow:

$$F1\ Score = 2 * \frac{precision * recall}{precision + recall}$$

Project Design

Step 0: Import Datasets

Import the dataset to be used in the project

Step 1: Detect Humans

Build the model for detecting humans on the image

Step 2: Detect Dogs

Build the model for detecting dogs on the image

Step 3: Create a CNN to Classify Dog Breeds (from Scratch)

Build a model for classifying dog breeds from scratch

Step 4: Create a CNN to Classify Dog Breeds (using Transfer Learning)

Build a model for classifying dog breeds using transfer learning

Step 5: Write your Algorithm

Write an algorithm that accepts a file path of an image and will determine the breed of the dog. The condition are as follows:

1. if a **dog** is detected in the image, return the predicted breed.
2. if a **human** is detected in the image, return the resembling dog breed.
3. if **neither** is detected in the image, provide output that indicates an error.

Step 6: Test Your Algorithm

Testing the algorithm to at least 6 images with at least two human and two dog images.

References

1. https://docs.opencv.org/trunk/db/d28/tutorial_cascade_classifier.html
2. https://neurohive.io/en/popular-networks/vgg16/
3. https://www.youtube.com/watch?v=YRhxdVk_sIs
4. https://towardsdatascience.com/benchmarking-simple-machine-learning-models-with-feature-extraction-against-modern-black-box-80af734b31cc
5. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
6. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html