Project Proposal

Dog Breed Classifier

By Gaurav Pal

Domain Background

The origins of dogs date back thousands of years, having evolved as domesticated descendants of the wolf, whereas modern dog breeds date back to the late 19th century. Prior to the Victorian era, there were different types of dogs that were defined by their function. Many different terms were used to describe dogs, such as breed, strain, type, kind, and variety. Dogs are the most variable mammal on earth, with artificial selection producing around 450 globally recognized dog breeds. Breed classification remains a relevant problem for the dog owners who searches for a show-class puppy or for a working dog with specific traits. Deep learning/Computer vision algorithms are particularly useful in identifying the image pattern and thus can be used to classify the dog breeds from a picture after training the model with substantial amount of training images.

Problem Statement

The goal of this project is to build a CNN model that can be used to identify dog breeds from use supplied images. This project covers following tasks:

Dog Face Detector – Identify the dog breed from supplied dog image.

Human Face Detector – Identify the resembling dog breed from the human image.

Datasets and Inputs

Classifier uses images as an input. Dataset used in this project is provided by Udacity and it contains images of dogs and humans.

Dog images dataset: The dog image dataset has 8351 total images which are sorted into train (6,680 Images), test (836 Images) and valid (835 Images) directories.

Human images dataset: The human dataset contains 13233 total human images which are sorted by names of human (5750 folders). All images are of size 250x250 with different background and angles.

The dog dataset is used for training, validation and testing of the CNN classifier. The human

Dataset is used for testing of the dog breed classifier in such case that it predicts the resembling breed from the human image.

Solution Statement

Convolutional Neural Network (CNN) is used to address this multiclass image classification problem.

Convolutional Neural Network (CNN) is a type of deep learning algorithm which can take an image as input(tensors) and then differentiate the images from one another by assigning weights and biases to various aspects or objects in the image.

The solution has following steps to classify the provided images:

Step 1 - Detect human images, we can use existing algorithm like OpenCV's implementation of HAAR feature based cascade classifiers.

Step 2 - Detect dog-images we will use a pretrained VGG16 model.

Step 3 - Once the image is identified as dog/human, we can pass this image to an CNN which will process the image and predict the breed that matches the best out of 133 breeds.

Benchmark Model

The CNN model created from scratch must have accuracy of at least 10%. This can confirm that the model is working because a random guess will provide a correct answer roughly 1 in 133 times, so achieving a high accuracy using bare-CNN is not inherently possible.

The pre-trained model should have an accuracy of at least 60% so that it can be successfully used in a dog breed classifier app.

Evaluation Metrics

Accuracy metric is used as an evaluation metrics to gauge the performance of both models. Accuracy is calculated as number of correct predictions divided by total number of predictions and return a percentage value bounded between 0 and 100 and provides a good measure of the model performance.

Project Design

Following are the development steps for this implementation –

Step 1 - Import the necessary dataset and libraries, Pre-process the data and create train, test and validation dataset. Perform Image transformation on training data.

Step 2 - Detect human faces using OpenCV's implementation of Haar feature based cascade classifiers.

Step 3 - Create dog detector using pretrained VGG16 model.

Step 4 - Create a CNN model from scratch to classify dog breeds, train, validate and test the model.

Step 5 - Create a CNN to Classify Dog Breeds using Transfer Learning. Train and test the model.

Step 6 - Write an algorithm to combine Dog detector and human detector.

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

Step 7 - Test the model on sample images.

References

1. https://en.wikipedia.org/wiki/Dog_breed
2. Imagenet Documentation - https://github.com/pytorch/vision/blob/master/torchvision/models/vgg.py
3. PyTorch Documentation – https://pytorch.org/docs/stable/index.html