Capstone Project Proposal

Dog Breed Detector App

Author: Sean Black

Background

People with dogs for pets often get asked the same question by everyone that meets their furry pet: “What kind of dog is he/she?”. If you have a dog, chances are you have at least a little knowledge as to what type of breed the dog is. You can even make inferences about the type of breed just by looking at the dog itself.

The goal of this project is to automate these types of inferences using convolutional neural networks, a type of machine learning algorithm that is structured similarly to human neural networks. These types of algorithms are often used for image classification problems, as they can scale and interpret large and complex datasets of naturally occurring data elements. In this case, the neural network will take in an image of a dog or human and transform the image’s pixel information into a grid of quantitative data points. Each feature of the grid (pixel) will serve as a “node” in our network and will be trained on a collection of dog breeds. When the application is complete, users will be able to upload their images and the application will be able to determine what kind of dog breed the image most likely resembles if the image contains a dog or a person.

Problem Statement

Can this application detect the correct dog breed for pictures of dogs and make accurate inferences about dog breed resemblance for pictures of people?

Datasets and Input

Dog Dataset:

- Provided by Udacity
- Split into train, test, and validation directories, each containing folders of images separated by breed

Human Dataset:

- Provided by Udacity
- Folders of images separated by the individual’s name

Input:

- File path of an image that the user wants to assess

Solution Statement

Using a Haar feature-based classifier and a pre-trained neural network, detection methods will determine whether the image contains a human, a dog, or neither. Once a dog or human is detected, our custom convolutional neural network will infer what kind of dog breed the image most likely resembles.

Capstone Project Proposal

Dog Breed Detector App

Author: Sean Black

Benchmark Model

Per the instructions on the Udacity workbook, the custom neural network from scratch will need to pass a benchmark score of 10% accuracy or better on the testing data. The custom model that uses transfer learning will need an accuracy score of 60% or better.

Evaluation Metrics

The metric that will first be used to determine the model's accuracy will be PyTorch's negative log likelihood loss function also known as NLLLoss. The input of this loss function is a tensor that contains information about the log-probabilities of the nodes in the final output layer. When defining the network, a LogSoftmax layer will determine the probabilities in these nodes. The loss function will take in default parameters, meaning that mean log loss on a per batch basis will be calculated.[1]

Project Design

Images will first be uploaded and preprocessed before they are fed into the machine learning algorithms. Image transformation via cropping and color scaling will be applied so that images are optimized for training.

The project will use pre-trained models to create the dog and human face detectors for preliminary image analysis. These detectors will return true or false and dictate the text output of what the application returns to the user.

The dog breed classifier will use an optimized pre-trained model to determine what kind of dog breed the image most likely resembles. This classifier will be optimized for maximum accuracy and will be saved as a checkpoint which will be loaded when the application calls the predict function. A GPU will be used for training and inference if available. The following components will be fine-tuned to for optimization:

- Pre-trained model architecture (densenet, vgg, etc…)
- Hidden layer architecture of classifier
- Dropout rate
- Learning rate
- Loss function used (currently starting with NLLLoss)
- Training epochs

Prediction output will be passed to the user interface with the name of the breed, a picture of the original image, and a picture of the first image within the training dataset folder that belongs to a particular breed.

1. https://pytorch.org/docs/stable/nn.html#nllloss