Machine Learning Nanodegree Capstone Proposal : Dog Breed Classifier using CNN

Ji-woong Choi
September 16th, 2020

Domain Background

Starting with AlexNet, which first appeared in 2012, the Convolutional Neural Network has grown very steeply, and now it is showing more than an expert beyond the ability to distinguish between ordinary people.
As the research progressed repeatedly, networks that showed quite good performance in the classification field appeared, and after that, the data set itself became the criterion for determining the quality rather than the development of the network. In this proposal, we propose a model that classifies breeds by CNN using the Dog Face and Human Face datasets.

Problem Statement

The goal of project is to build a pipeline to process real-world, user-supplied images. The algorithm will identify an estimate of the dog’s breed given an image. It has to perform two tasks:

Dog Face Detector : Given an image of a dog, the algorithm will identify an estimate of the canine’s breed.

Human Face Detector : If supplied an image of a human, the code will identify the resembling dog breed.

If neither of the two is detected, then an error message will be issued.

Datasets and Inputs

All the datasets used to train, test, and validate the CNN model are provided by Udacity. The dataset consists of human and dog faces.

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. Each of this directory (train, test, valid) have 133 folders corresponding to dog breeds. The images are of different sizes and different backgrounds, some images are not full-sized. The data is not balanced because the number of images provided for each breed varies. Few have 4 images while some have 8 images.

Human Images Dataset : The human dataset contains 13,233 images.

Solution Statement

The solution is to design a CNN model that is able to estimate as much as possible the breed of a dog that is included in a picture. To do this, it must be recognized beforehand whether a person or a dog is included in a picture. If a person is present, the resembling dog breed is identified. If a dog is present, an estimate of the dog breed is given.

Benchmark Model

For our benchmark model, we will use the Convolutional Neural Networks (CNN) model created from scratch with an accuracy of more than 10%. This should be enough to confirm that our model is working because random guess would be 1 in 133 breeds which are less than 1% if we don't consider unbalanced data for our dog images.

Evaluation Metrics

On Kaggle they use multi-class log loss metrics to evaluate models. We'll also use this metric so we can compare it to results on Kaggle. We will also use F1 score testing because it considers precision and recall.

$$F_1 = \frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{\mathrm{t}p}{\mathrm{tp} + \frac{1}{2}(\mathrm{fp} + \mathrm{fn})}.$$

Project Design

The following strategy is what we choose for the project.

- Step 0 : Import Datasets
  There are 2 datasets (Human & Dog datasets). They are provided by Udacity.

- Step 1 : Detect Humans
  An algorithm will be developed that is able to recognize the faces of humans by using OpenCV and the integrated HaarCascade.

- Step 2 : Detect Dogs
  The pre-trained VGG-16 model, which comes with the weights that were trained on the ImageNet dataset, can be used here to design a dog detector.

- Step 3 : Create a CNN to classify Dog Breeds from Scratch
  Here I will develop a CNN from scratch to classify dogs. This will be implemented using PyTorch.

- Step 4 : Create a CNN to classify Dog Breeds using Transfer Learning
  Here I will use Transfer Learning to create a CNN to classify dogs. This will be also implemented with PyTorch. I will use the pre-trained ResNext101 model. This result should be better than the network developed under Step 3.

- Step 5 : Write My Algorithm.
  In this step I will put everything together. With a new image as input, I will first determine whether the image is a human or a dog. After that, I will return an estimate of the dog breed using the model from Step 4.

- Step 6 : Test My Algorithm.
  In this section, I will take my new algorithm for a spin.