

Machine Learning Engineer Nanodegree

Capstone Proposal

Dog Breed Classifier with CNNs

"Kirolos Wahba Moheeb"

October 31st, 2020

Domain Background

The Dog breed classifier is a well-known problem in Machine Learning. The problem is to identify a breed of dog if dog image is given as input. So if dog owners are unsure about their dog's breed, a dog-breed classifier can help them identify their dog's breed. This would be especially helpful for inexperienced dog owners.

This is a multi-class classification problem where we can use supervised machine learning to solve this problem. After completing this model, A web app can be developed where user can input an image and obtain prediction from this model.

The personal motivation about this project comes from my interest in computer vision field and the knowledge learnt from the nanodegree to solve these problems with cloud services tools.

Problem Statement

The goal of the project is to build a machine learning model that can be used within web app or any operating system app to process real-world, user-supplied images. I will be using CNNs and pretrained models in solving this problem.

The models used have to perform two tasks:

- **Dog face detector:** Given an image of a dog, the model will identify an estimate of the canine's breed.
- **Human face detector:** If supplied an image of a human, the model will identify the resembling dog breed.
- If neither of the two is the case, an error message is issued.

Datasets and Inputs

For this project, the input format must be of image type, because we want to input an image and identify the breed of the dog.

The following datasets provided by Udacity will be used:

- **The dog dataset** with 8351 dog 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.
- **The human dataset** with 13233 human images which are structured in folders by celebrities' name (5750 folders). All images are of size 250x250. Images have different background and different angles. The data is not balanced because we have 1 image for some people and many images for some.

Solution Statement

For performing this multiclass classification, I will use Convolutional Neural Network to solve the problem.

The solution involves three steps:

1. Detect human images, we can use existing algorithm like OpenCV's implementation of Haar feature based cascade classifiers.
2. Detect dog-images where we will use a pretrained VGG16 model.
3. After 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.

The accuracy of the predicted breed of dogs' images will be measurable, because we have a labelled data that we can compare the predictions against. The accuracy of the predicted breed for humans' images will not be measurable.

Benchmark Model

I found an interesting benchmark model in this paper (2). On page 4 in Figure 5 the accuracy of different state of the art models is compared. So I choose the ResNet-50+FT+DA with test accuracy of 89.66% as my benchmark model.

Evaluation Metrics

I would use test accuracy as an evaluation metric to compare my models with the benchmark model. Additionally, the performance will be compared between the following convolutional neural networks: dog breed's classifier trained from scratch and dog breed's classifier trained with transfer learning.

Project Design

The following steps will be followed to go through the project:

1. Import the necessary dataset and libraries, Pre-process the data and create train, test and validation dataset. Perform Image augmentation on training data.
2. Detect human faces using OpenCV's implementation of Haar feature based cascade classifiers.
3. Create dog detector using pretrained VGG16 model.
4. Create a CNN to classify dog breeds from scratch, train, validate and test the model. I will implement this using PyTorch.
5. Create a CNN to Classify Dog Breeds using Transfer Learning with resnet-101 architecture. Train and test the model. I will implement this too using PyTorch.

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.
7. Test my algorithm.

References

1. Original repo for Project - GitHub: <https://github.com/udacity/deep-learning-v2-pytorch/blob/master/project-dog-classification/>
2. Ayanzadeh, A.; Vahidnia, S. Modified Deep Neural Networks for Dog Breeds Identification. Preprints 2018, 2018120232 (doi:10.20944/preprints201812.0232.v1).