Volodymyr Mudryi
May 28th, 2020

Convolutional Neural Networks (CNN) project
Dog Breed Identification

Domain Background

Today a machine learning has a lot of hype around it. It can help to solve a complicated business tasked and make some fun application for community. So with an enough amount of data you can do a lot of interesting and useful things.

So I previous I think about a app that help with classification a dog breeds on a photo, it can be useful for some sort of quick check a breed of a dogs walking around or can show actually what the name of this dog breed. Therefore, I without a doubt choose this task.

Also this task is interesting for me to improve my Pytorch skill with CNN, because in previous I only used Keras framework.

Problem Statement

The main idea is to take a picture of any dog and app tell what is dogs breed, also app should check if there are any dogs on a picture and for some funny and interesting stuff, if you upload a picture of human as result you get a dogs breed that most related to your photo.

To solve this problem, I will build a supervised algorithm (because we have a lot of labeled data) especial a multi-class classificatory.

Datasets and Inputs

This task requires a following dataset: labeled images with dogs of different breeds and their breeds names (dog detection and breed classification) and photos with human (for human detection).

All the required data is provided by Udacity.

So in result for dogs pictures we got 8351 splited into a three folders: train (6,680 Images), test (836 Images) and valid (835 Images) which divided by folders by dog breeds (133 folders in all splits). All this pictures have different sizes, lightning, backgrounds and dog position so it will help to build more robust algorithm.

For human pictures we got 13233 images without any splits because we use it only for test pre-trained models and wouldn't train any human recognition algorithm. Therefore, all human pictures are the same size (250×250)

Solution Statement

To solve this task, I will use a CNN types of models (common approach for image task). So there will be a human detector which used a OpenCV approach, dog detector which use a pre-trained VGG model, a build-from-scratch CNN (based on AlexNet architecture) and if it will have a poor result I will use ResNet pre-trained model for transfer learning approach (because not enough train data).

Benchmark Model

So in this solution the first, build-from-scratch, model will be a benchmark, this model should perform better that random guess which are less than 1% accuracy.

Evaluation Metrics

This is a classification task with a lot of classes, so a common approach will be to use multi-class log loss metrics to model evaluation.

Project Design

Firstly, after loading data there will be OpenCV approach for human faces detection. I will use a Haar feature-based cascade classifier which take some preprocessing steps and return if any human faces are found.

Next, we detect dogs on images, for this task I will use a pre-trained VGG16 model, which running on GPU and after preprocessing step it return an index from ImageNet dataset, where is a lot of different dogs breeds, so in that case we should choose we will label as a dog all related classes (158-268).

After that we use a train, test, validation splits to train a benchmark model from scratch, based on CNN architecture. This model should achieve an accuracy score more that 10% on test data.

Next step, will be to improve algorithm with transfer learning, so I will use a ResNet101 model, where the final fully connected layer is replaced by layers with ours number of output classed (133 different dog breeds). After training we should achieve more that 60% of accuracy on same test set.

Finally, we add this all together to make a pipeline, where app after received a photo can check if it any human or dog on it, and return a dog breed of that dog or a dog breed which is the most related to human on picture.