MACHINE LEARNING ENGINEER NANODEGREE

Capstone Proposal

Dog Breed Classifier with CNNs

Navjot Singh Bajaj

(8th August 2020)

1. Domain Background

As evident from history, Dogs play a major role in the life of human beings, be it hunting or guarding homes and farms. Faithfulness of dogs has made humans shower their love on them. So I am one of those dog lovers and always wanted to conserve them and spread awareness about dogs. There are about 350 breeds of dogs, some being very similar to each other's. In this course I have chosen this topic to fulfill my desire of knowing more about dogs and their breeds. The problem is to identify a breed of dog if dog image is given as input, if supplied an image of a human, we have to identify the resembling dog breed. The idea is to build a pipeline that can process real world user supplied images and identify an estimate of the canine's breed. This is a multi-class classification problem where we can use supervised machine learning to solve this problem. After completing this model, I am planning to build a web app where user can input an image and obtain prediction from this model. This project gives me an opportunity to build and deploy ML models on my one of the favorite topic.

2. Problem statement

The aim of the project is to build a pipeline to process real-world, user-supplied images. The main objective is to assemble an AI model that can be utilized inside web application to process genuine world, client provided pictures. The algorithm will identify an estimate of the dog's breed given an image. When the image is of a human, the algorithm will choose an estimate of a dog breed that resembles the human.

3. Datasets and Inputs

For this venture, the information design must be of picture type, since we need to include a picture and distinguish the type of the canine. The dataset for this undertaking is given by Udacity. The dataset has pictures of canines and people.

Canine pictures dataset

The canine picture dataset has 8351 all out pictures which are arranged into train (6,680 Images), test (836 Images) and substantial (835 Images) indexes. Each of this registry (train, test, legitimate) have 133 organizers relating to canine varieties. The pictures are of various sizes and various foundations; a few pictures are most certainly not full-sized. The information is not adjusted in light of the fact that the quantity of pictures accommodated each breed shifts. Few have 4 pictures while some have 8 pictures.

Human pictures dataset

The human dataset contains 13233 absolute human pictures which are arranged by names of human (5750 organizers). All pictures are of size 250x250. Pictures have distinctive foundation and various edges. The information isn't adjusted since we have 1 picture for certain individuals and numerous pictures for a few.

4. Solution Statement

For playing out this multiclass grouping, we can utilize Convolutional Neural System to take care of the issue. A Convolutional Neural Network (CNN) is a Deep Learning calculation which can take in an info picture, allocate significance (learnable loads and predispositions) to different viewpoints/objects in the picture and have the option to separate one from the other. The arrangement includes three stages. To begin with, to recognize human pictures, we can utilize existing calculation like OpenCV's execution of Haar highlight based course classifiers. Second, to distinguish canine pictures we will utilize a pretrained VGG16 model as it has 1000-way fully-connected layer with Softmax in the end and trained on ImageNet, a large popular dataset for image classification. At long last, after the picture is distinguished as canine/human, we can pass this picture to a CNN which will process the picture and foresee the variety that coordinates the best out of 133 varieties. The CCN is created using Resnet101 which is a 101 layers deep Neural network, capturing granular spatial information from the image. Resnet model significantly boost the accuracy of classification task as CNN provides better way of representation for each input image and residual blocks that generates the identity connections making gradient flow easy.

5. Benchmark Model

In the project we use the CNN model that will be created from scratch and must have accuracy of at least 10%. When the accuracy is greater than 10% then this can confirm that the model is working and It is working because a random guess will also providing a correct answer roughly 1 in 133 times, which is likely equal to an accuracy of less than 1%. The final CNN model which is created using transfer learning must have accuracy of 60% and above to make a good model.

6. Evaluation Metrics

For this multi class classification, Multi class log loss will be used to evaluate the model. Log loss takes into the account of uncertainty of prediction based on how much it fluctuates from actual label and this will help in evaluating the model. Accuracy will be the main metric used to test both the benchmark model and the solution model.

Accuracy = (TP+TN) / (TP+TN+FP+FN)

True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN)

7. Project Design

Step 1: Import the necessary dataset and libraries, Pre-process the data and create train, test and validation dataset. Perform Image augmentation 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 to classify dog breeds from scratch, train, validate and test the model.

Step 5: Create a CNN to Classify Dog Breeds using Transfer Learning with resnet101 architecture. Train and test the model.

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

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

References

1. Original repo for Project - GitHub: https://github.com/udacity/deep-learning-v2-pytorch/blob/master/project-dog-classification/
2. Resnet101: https://pytorch.org/docs/stable/_modules/torchvision/models/resnet.html#resnet101
3. Imagenet training in Pytorch: https://github.com/pytorch/examples/blob/97304e232807082c2e7b54c597615dc0ad8f6173/imagenet/main.py#L197-L198
4. Pytorch Documentation: https://pytorch.org/docs/master/
5. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neuralnetworks-the-eli5-way-3bd2b1164a53
6. http://wiki.fast.ai/index.php/Log_Loss