UDACITY
MACHINE LEARNING ENGINEERING NANODEGREE

Capstone Proposal

DOG BREED CLASSIFIER WITH PYTORCH

EROL ŞEN

June 20th 2021
GERMANY

DOMAIN BACKGROUND

Dog breed classification dataset is one of the great challenge of machine learning area.
This project aims to classify dog breeds using PyTorch CNN Model. After I train a model, it can be used as a web application, and I am planing to make it alive.

This idea gives us easy way to recognize all dog breeds when we see them in the street or somewhere else.

PROBLEM STATEMENT

In this case, the problem is classifying the dog breeds. For that I will use pre-trained CNN model which is called ResNet50.

The trained model tries to detect human face. If human is detected, I will also make algorithm to search for dog breeds. And same thing will be for dog breeds. If dog breed is classified, I will make algorithm to search for human face.
After that, the algorithm returns like "*I detect a human who look like Löwchen and I detect a Borzoi*"

DATASETS AND INPUTS

The dataset contains 13233 human images and 8351 dog images. Train dataset has 6680 images, valid dataset has 835 images and test dataset has 836 images. Which I can assure that dataset is reasonable. A model can be trained effectively.

On the other hand, I should use ***CrossEntropyLoss*** as a Loss Function due to that we have more than one classes.

SOLUTION STATEMENT

The problem is computer vision problem. Best way to obtain effective classifier is choosing pre-trained model. For instance "VGG16, ResNet50, Inception, etc.." If model detects a dog, it will return the breed.

As I mention in ***Problem Statement*** section, I will make algorithm to search for dog breed in human images. And also, I will apply same technique for dog images. So that, the model can detect human and dog in same images. I believe that it will work.

BENCHMARK MODEL

In the first step, I used VGG16 model that has ImageNet weights. This pre-trained model will be my benchmark model.

The specified model which will be used in this problem is ResNet50. I believe that after some hyperparameter tuning, this ResNet50 model gives me best result in this problem.

The notebook provided also stated that at least 10% test accuracy for CNN model that I create and 60% accuracy must be achieved on test set for Transfer Learning model to classify to dog breeds.

EVALUATION METRICS

The main purpose is comparing the performance of the model. I will be comparing the benchmark model and ResNet model, so that I will use ***accuracy*** as an evaluation metric.

PROJECT DESIGN

Steps are listed by provided notebook:

Step 0: Import Datasets
Step 1: Detect Humans
Step 2: Detect Dogs
Step 3: Create a CNN to Classify Dog Breeds (from Scratch)
Step 4: Create a CNN to Classify Dog Breeds (using Transfer Learning)
Step 5: Write your Algorithm
Step 6: Test Your Algorithm

REFERENCES

1. ***Deep Residual Learning for Image Recognition***
https://arxiv.org/abs/1512.03385
2. ***PyTorch Pre-Trained Models***
https://pytorch.org/docs/0.3.0/torchvision/models.html
3. ***PyTorch Transfer Learning Tutorial***
https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html
4. ***PyTorch ResNet Tutorial***
https://pytorch.org/hub/pytorch_vision_resnet/
5. ***Original Udacity Project***
https://github.com/udacity/deep-learning-v2-pytorch/tree/master/project-dog-classification
6. ***Udacity Transfer Learning GitHub Project***
https://github.com/udacity/deep-learning-v2-pytorch/blob/master/intro-to-pytorch/Part%208%20-%20Transfer%20Learning%20(Solution).ipynb