Udacity Machine Learning Engineer Nanodegree

Capstone Proposal

Pyae Phyo Hein

October 2, 2020

Dog Breed Classification

Domain Background

The brains behind the Industry 4.0 are surely Artificial Intelligence (AI). There are many different types of AI such as Deep Learning and Machine Learning. Machine Learning concept is introduced by Arthur Samuel in 1959 so it is not new, but till today it is still developing and more powerful. The main important thing for ML and DL is Data. "If you have Data, you could be billionaire". Data is a new kind of gold for industry 4.0.

Different ML products use different models. It only depends on the experience of the developer. This project is one of the famous project of Udacity Machine Learning Engineer Nanodegree. The main idea behind this project is using Convolutional Neural Network (CNN).

When I make this model I want to build WebApp and the main reason for choosing this project is because this is the best starter path to solve a more complex problems that I believe.

Problem Statement

The main goal of this project is to classify and detect on the input image of user. The problem statements on this projects are:

1. Detecting the human or dog, and
2. Classification of Dog Breed

The output will be the resulted image and print out the input image is dog or human and dog's breed, even if the detect image is human.

Dataset and Inputs

Human Datasets

Human pictures are sorted by name of each human. We have 13233 images of 5750 people with 250 * 250 pixels in all images. Some people have only one image but some are not. We want to detect human and going to use OpenCV library, I won't split it into train, valid and test.

Dog Breed Datasets

There are 8531 dog images of 133 breeds. Dog images have different images, different background, some dogs are in full sizes and some just ahead. Lighting is not the same. I prefer using 6680 images in train, 836 images in valid and 835 images in test. The major goal of this project is to classify dog breed, I'll create CNN both from scratch and transfer learning.

By looking this sample pictures, we can see that some images have only one human but some are not.

Solution Statement

We will use Convolutional Neural Networks (CNN) to make a model. CNN is a part of deep neural networks and it is great for analyzing images. To find if the picture is human or not we will use the OpenCV model. And to find if the dog is on a picture we will use a pre-trained VGG16 model using transfer learning. I am going to use 64, 128, 256 and 512 filters respectively in layers. I'll set kernel size 3 and padding 1 for convolutions layers. Then, set kernel 2 and stride 2 in maximum pooling layer to reduce image size in half. And for non-linear activation function, I used RELU. I created 3 fully connected layers and the final layer is yield predicted class. I also added dropout with p =0.5 to avoid overfitting. Finally, I'm going to use transfer learning to get better performance.

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

The problem we try to solve is a classification problem. Our data is an unbalanced, simple accuracy score is not very good here. The dataset is split up into train, validate, and test dataset. The dataset I got is already split up and so I don't need to do it manually. These splits used in both scratch and transfer learning. The evaluation metric is the accuracy. For getting the best trained model during training, I just compared the best loss with the validation loss and assumed as best trained model so far if validation loss is less than best loss.

Best loss = min (Best loss, Validation Loss)

Accuracy = corrected prediction / dataset size

Project Design

1. Import dog dataset and human dataset and preprocess the image and use augmentation techniques to prevent overfitting.
2. Detect human using OpenCV haarcascade classifiers
3. Detect human using pre-trained VGG16 model
4. Detect Dogs
5. Create a CNN model to Classify Dog Breeds (from scratch)
6. Create CNN model to Classify Dog Breeds (using Transfer Learning)
7. Write My Algorithm
   a. If detect human print Hello, You look like a …
   b. If detect dog print This dog is a …
   c. If don't detect human or dog print 'Sorry! We can't detect a human or a dog in this image.'

8. Testing my Algorithm

References

1. Learn CNN from Deep Learning Specialization course by Andrew Ng

   https://www.coursera.org/learn/convolutional-neural-networks

2. Restnet101

   https://www.kaggle.com/pytorch/resnet101

3. OpenCV Course

   https://www.alexsnowschool.org/courses/take/opencv-with-python

4. Artificial Neural Network by Zaw Min Khaing (book)