Capstone Project – CNN Project: Dog Breed Classifier

Udacity – Machine Learning Engineer Nanodegree

Buu Son, Dinh
Stuttgart, Germany
May 25, 2020

Abstract

This project uses computer vision technique to predict dog breeds from images. The involved Convolutional Neural Networks (CNN) model is implemented to bridge the gap of traditional Machine Learning techniques in localization and classification in the real-world. Besides, the project also aims to deploy a REST API along with web or mobile app to process user-uploaded images.

Keywords: CNN, REST API

1. Domain Background

Dog Breed Classifier is one of elective final project in Udacity Machine Learning Engineer Course. Some of breeds share the same body features and structure, so identifying breed in an image is a challenging task. Even with our human eye, guessing correct dog breed is also impossible. For example, Labrador retrievers can be yellow, black or brown, which can confuse our intuition. Besides, noise such as background color can result in a misleading conclusion.

One of the popular works in fine-grained classification was plant species classification by Belhumeur et. al. by segmenting a leaf to define its shape in order to determine the species. Additionally, Whitney et. al proposed the state-of-the-art method of dog breed identification by introducing traditional Machine Learning Classification Model beside CNN. Dog facial key points were gained from CNN were then fed in Support Vector Machine classifier to predict the breed of the dog shown in the image.

The complete project includes a web or mobile app, takes image URL from user's input and produces the breed of the dog. If a human is involved in the image, the system will provide a look-a-like method for dog breed guessing.

In the future, this model can be extended to other fields, classifying motorbike for example. Motorcycle modification is prohibited in some countries, by identifying which model of the motorcycle, the officers can have a clear insight about its stock components and put a fine on the suspector if he or she have unauthorized components.

2. Problem Statement

The purpose of the project is to create a model pipeline to predict the species of the dog in the user-provided image. The facing problems are:

- Image has more than one dog: this can be solved by implementing another model to detect each dog independently, then crop a bounding box around the dog to feed in our prediction model to find which breed is
- Noise in the image: cropping can also help
- Dogs overlap each other in the image: this is an extremely challenging task, and hopefully can be solved in the future
- Different species have minor differences in features, which might not be distinguished efficiently right away.

The project is a basic version of Whitney's, a CNN is built and deployed so as to provide real-time prediction in a web or mobile app. By designing a CNN from scratch, the students gain more knowledge and insights to further apply to the transfer learning technique. Since feature extraction is complicated, a pre-trained model with modifications is used for prediction.

3. Datasets and Inputs

There are two data files associated with this project:

- /dogImages: Dog images, consists of 8351 images in total, where train, test, and validation sets include 6680, 836 and 835 images respectively.
- /lfw: human images, consists of 13233 images.

The classifier will take one out of 133 categories of dog.

4. Solution Statement

The project starts with face_detector (based on OpenCV) and dog_detector (pre-trained Visual Geometry Group – VGG16), then the output is passed through a transfer learning CNN model to increase the accuracy in classifying look-a-like dog breed even for human or dog.

5. Benchmark Model

CNN model is one of the most popular methods to classify images, surpassing traditional machine learning model. In this context, we will restrict the epochs of CNN model to 100 with the help of transfer learning technique. The result then will be evaluated with accuracy score and human-eye observation.

6. Evaluation Metrics

The model is evaluated on the provided shuffled test dataset over accuracy score and human-eye observation. In this way, the performance of this model can be guaranteed and improved, for example, data cleansing before training, add or remove layer to prevent overfitting.

7. Project Design

The guideline for this project was provided in dog_app.ipynb as follows:

- Step 0: Import Datasets

Download and load the dataset of dog and human in to the project.

- Step 1: Detect Humans:

By using OpenCV's implementation of Haar feature-based cascade classifiers, human faces are promised to be detect with high probability. In addition, not only human but dog datasets are tested with this detector, in order to provide clear insight if dog can be misclassified as human and vice versa.

- Step 2: Detect Dogs:

In this step, the pre-trained VGG-16 model, along with ImageNet dataset containing over 10 million URLs of 1000 categories is introduced. Dog detector is measured over the dog and human dataset as well.

- Step 3: Create a CNN to classify dog breeds (from Scratch)

With the restriction of over 10% accuracy, the scratch CNN model needs to be carefully observed and measured, because the chance that one breed can be correctly classified is 1/133 which is less than 1%.

- Step 4: Create a CNN to Classify Dog Breeds (using Transfer Learning)

By meet the requirement of Step 3, achieving basic foundation of CNN, we will be capable of optimizing the elective pre-trained model to fit the outcome of the project, which is attaining at least 60% accuracy on the test set.

- Step 5: Write your Algorithm:

After completing the model pipeline, at this step, we have to write our own function implementing the model at step 4 to take an image path and produce prediction as follows:

- if a **dog** is detected in the image, return the predicted breed.
- if a **human** is detected in the image, return the resembling dog breed.
- if **neither** is detected in the image, provide output that indicates an error.

- Step 6: Test your algorithm

At this step, when everything is setup, sample images are using to evaluate the model and give conclusion.

References

https://web.stanford.edu/class/cs231a/prev_projects_2016/output%20(1).pdf

https://pytorch.org

P. N. B. et al. Searching the world’s herbaria: A system for visual identification of plant species. 2008.

https://github.com/udacity/deep-learning-v2-pytorch/tree/master/project-dog-classification

https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148

https://flask.palletsprojects.com/en/1.1.x/