Joost Zegers

CNN Dog Breed Classifier

Machine Learning Engineer Nanodegree - Udacity

PROJECT OVERVIEW

For the Capstone project, I have elected the Dog Breed Classifier project provided by Udacity. The project focuses on implementing Convolutional Neural Network (CNN) algorithms in order to identify the breed of a dog from an image.

My reason for choosing this project is simple: I would like to learn more about image classification algorithms. At university, I spend a lot of time learning about algorithms for time series forecasting and clustering. Image classification was not part of the curriculum. This Capstone project is a great opportunity to learn about image classification!

DOMAIN BACKGROUND

In the lessons about CNNs, we learned that they are used to obtain state-of-the-art results in the field of Computer Vision[1]. The image classification models that achieve highest accuracy on the Imagenet dataset all use CNNs[2]. Hence, it makes sense that we use CNNs to build a dog breed classifier.

PROBLEM STATEMENT

The goal of the project is to build a pipeline that processes real-world images and is able to:

1. Classify whether an image contains a human face, a dog or neither
2. If a dog is detected: estimate the dog's breed (with >60% accuracy)
3. If a human face is detected: estimate the most-resembling dog breed

Once completed, the pipeline should be able to be used as part of a mobile or web-based app for dog breed identification.

---

[1] Machine Learning Engineer Nanodegree > Extracurricular > 2. Additional Materials: Convolutional Neural Networks > Convolutional Neural Networks > 2. Applications of CNNs
https://classroom.udacity.com/nanodegrees/nd009t/parts/e8261bb7-c4bb-4c9c-8eae-2347c5aa92d2/modules/19a75d10-547d-4497-ae68-609ca1a235fc/lessons/807590ea-abd5-4581-b91d-9eede9a0aad2/concepts/e7190f8c-c824-4936-89ff-db6230fd3d12

[2] https://paperswithcode.com/sota/image-classification-on-imagenet

DATASET

The *Stanford Dogs* dataset[3] is used to develop algorithms for recognizing whether an image contains a dog and classifying the dog's breed. The *Labeled Faces in the Wild* (LFW) dataset[4] is used to implement a human face detector algorithm. Finally, both datasets are used to test the final algorithm, which should be able to perform the three tasks described in the problem statement.

Both the *Stanford Dogs* dataset and the LFW dataset have been pre-processed by the project creators at Udacity. The resulting *dog image* dataset contains 8,351 labeled images of 133 dog breeds, already split into train, validation and test data sets. The LFW dataset contains 13,233 human face images.

PROPOSED SOLUTION

1. Detect human faces

A pre-trained face detector from OpenCV is used to detect human faces in images. The selected implementation uses Haar Cascades[5].

2. Detect dogs

A pre-trained VGG-16 model is used to detect dogs in images. This model has been trained on ImageNet - a very large dataset used for image classification models.

3. Classify dog breed - CNN build from scratch

Here, we build a dog breed classifier CNN from scratch without using transfer learning. The model must attain a test accuracy greater than 10%. We design the model architecture, loss function and optimizer. The train, validation and test split has been made for us by the project creators at Udacity.

4. Classify dog breed - CNN build using transfer learning

Now, we may use transfer learning to create a CNN that classifies dog breeds from images. This CNN must attain a test accuracy greater than 60%. Similar to (3), we design the model

[3] http://vision.stanford.edu/aditya86/ImageNetDogs/
[4] http://vis-www.cs.umass.edu/lfw/
[5] https://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html

architecture, loss function and optimizer. We use the provided dataset split for training, validation and testing.

5. Algorithm that combines proposed solution 1, 2 and 4

Finally, we write an algorithm which combines solutions 1, 2 and 4 to meet all goals set out in the Problem statement. This algorithm could be used as part of a dog breed identification app.

BENCHMARK

We regard the CNN without transfer learning as our benchmark model. In doing so, we can assess the benefit of using a transfer learning approach more clearly.

Also, the project creators have set an accuracy benchmark for both the CNN without transfer learning (>10%) as well as the CNN that uses transfer learning (>60%). These respective accuracies must be achieved on the *dog image* test set.

EVALUATION METRICS

The prediction accuracy on the *dog image* test set is used to evaluate both CNNs (with and without transfer learning).

PROJECT DESIGN

1. Download datasets provided by Udacity
2. Detect human faces using OpenCV model
3. Detect dogs using VGG-16 model
4. Build dog breed classifier CNN from scratch (without transfer learning)
    a. Investigate potential data augmentation
    b. Find model architecture, loss function and optimizer
    c. Test prediction accuracy
5. Build dog breed classifier CNN using transfer learning
    a. Investigate transfer learning
    b. Find model architecture, loss function and optimizer
    c. Test prediction accuracy
6. Write algorithm for dog breed identification app
7. Test algorithm for dog breed identification app