Machine Learning engineering nanodegree

Capstone project report

DOG BREED CLASSIFIER

Xin Huang

1. Definition

1.1 Project overview

Images are very important for interacting with human life. With the rapid development of smartphones and cameras, more and more images can be captured and stored. It's easy for humans to recognize the meaning of the images while it's just pixels for computers to recognize. Deep learning technologies are used to recognize the image with higher accuracy and dog image classification is a classical problem, defined as predicting the dog breed for a user-supplied image. Different dogs have their special traits, which help define the particular breed. Developing a dog breed classifier enables owners to identify the dog's breed.

In this project, we use the convolutional neural network(CNN) to classify dog breeds. The data to be used for training our model is provided by Udacity, which includes 8351 dog images across 133 breeds and 13233 human images.

1.2 Problem statement

The purpose of the project is to build a machine learning model that can be used to process real-world, user-supplied images. Given an image of a dog, the algorithm will identify the dog's breed. If supplied an image of a human, the algorithm will choose an estimate of a dog breed that resembles the human. If neither a dog or a human is detected, the algorithm will return an error message. With the use of transfer learning, the classifier should perform an accuracy higher than 60%, which is set by Udacity.

1.3 Metrics

Accuracy will be the main metric used to test both the benchmark model and the solution model. In the equation shown below, TP, FN, FP and TN represent the number of true positives, false negatives, false positives and true negatives, respectively.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

During model training, we compare the test data prediction with validation dataset and calculate multi class log loss to find the best performing model. Log loss takes into account the uncertainty of prediction based on how much it varies from actual label and this will help in evaluating the model.

2. Analysis

2.1 Dog images

Dog image datasets contain 8351 dog images, including 6680 images(80%) used for training, 835 images(10%) for validation and 836 images(10%) for testing. For each training set, it has 133 dog breeds. Within a dog breed class, the maximum image count is 77 and the minimum count is 26. There are couples of dog images in the dog dataset shown in Figure 1.

Figure 1. Sample images from dog dataset

2.2 Human images

In Fig. 2, it shows the human images in the human dataset. There are 13,233 human images, containing 5749 distinct human names. The average number of images in each class is 2.3.

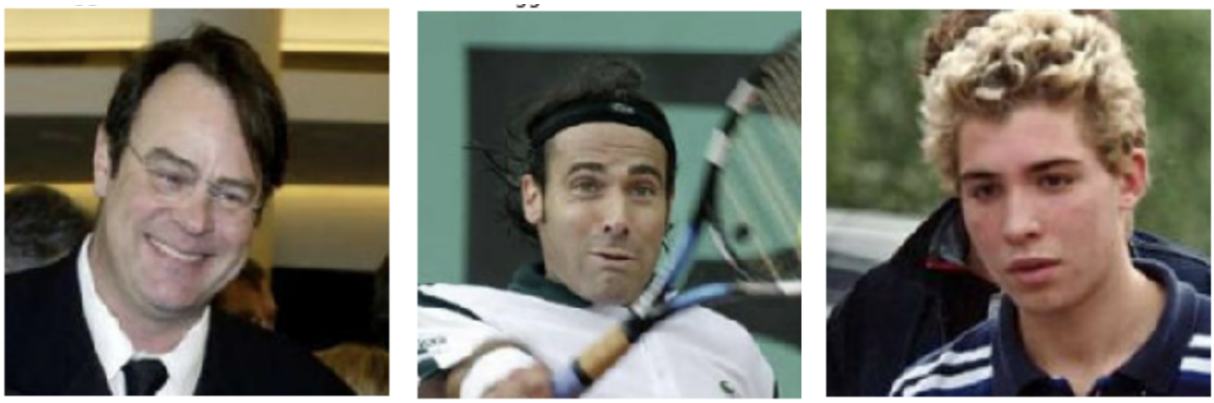

Figure 2. Sample images from human dataset

2.3 Algorithms and techniques

For performing this multiclass classification, Convolutional neural network (CNN) is applied to solve the problem. The solution involves four steps. First, we use existing algorithms such as OpenCV's implementation of Haar feature-based cascade classifiers to detect human faces. Second, a pretrained VGG-16 model is applied to detect dogs in user-supplied images. Then after

the image is identified as dog/human, we can use CNN model to classify dog breeds. This model should perform a test accuracy higher than 10%, which is set by Udacity. Finally, transfer learning will be used with a ResNet50 model to significantly boost the accuracy of the CNN model. It is required to achieve a test accuracy higher than 60% set by Udacity.

2.4 Benchmark

The CNN model created from scratch must have accuracy of at least 10%. The CNN pre-trained model with transfer learning should attain significantly enhanced accuracy. The CNN model created using transfer learning must have accuracy of 60% and above.

3. Methodology

3.1 Data preprocessing

There are two types of images in our dataset, human images and dog images. And we have different classifiers for them. For human images, we don't process the images for training purposes. However, we use the images to test the performance of our human face. The only preprocessing step is to convert the image to grayscale.

For dog classifiers and detectors, we need to preprocess the images before putting them into training models. First of all, I applied Random Resized Crop to resize the image to 224x224 pixels for the default size of model VGG 16. Then the images are cropped to the center. Finally we normalize the image with a mean of (0.485, 0.456, 0.406) and a standard deviation of (0.229, 0.224, 0.225).

3.2 Implementation

In this section, I will explain the algorithm and techniques that I have used for dog breed classification problems. There are mainly four parts, human face detector, dog detector, dog breed classifier from scratch as well as the final dog breed classifier.

3.2.1 Human face detector

OpenCV's implementation of the Haar feature-based cascade classifier is used to detect human faces. We get pre-trained face detectors provided by OpenCV. And then we convert the image into grayscale. To find the number of faces in the image, the method detectMultiScale() was used. This returns the number of human faces that the classifier has detected.

3.2.2 The dog detector

A pre-trained VGG-16 model is used to detect the dog images. We first download the VGG-16 model as well as its pre-trained weights. The transformation is described in the "Data Preprocessing" step and then feeds the image into the model. If the predicted index is between 151 and 268, this image is predicted to contain a dog.

3.2.3 Dog breed classifier from scratch

The dog classifier uses the ResNet-152 model and data processing is similar to the method in the dog detector. The pre-trained weights of the model from the convolutional layer have also been used for the dog breed classifier model. We trained our layer which takes 2048 input features, 512 nodes in the hidden layer and 133 output classes in the final layer. We use Stochastic Gradient Descent to train our model with a learning rate of 0.001. The model was trained for 20 epochs and we use log loss error measuring its performance.

3.2.4 The final dog breed classifier

The final dog breed classifier integrates the above steps. We check the image of either humans or dogs. Like the previous method, we take the image, load it and get its predicted breed using the predict_breed_transfer() function.

3.3 Refinement

The CNN model from scratch has an accuracy of 12% and it can be further improved by applying transfer learning. By using 20 epochs, the CNN model with transfer learning has an accuracy of 86%.

4. Results

4.1 Human Detector

The model detects 98 human faces from the first 100 human images and 17 human faces from the first 100 dog images.

4.2 Dog Detector

The model detects 98 human faces from the first 100 human images and 17 human faces from the first 100 dog images.

4.3 CNN with transfer learning

The accuracy on test image sets is 86%, which means it correctly predicts the dog breeds of 727 images out of 836 images.

5. Conclusions

In this section, I would like to show some output examples of our model.

Hi hooman. Your doggo version would be Black russian terrier.

Hi hooman. Your doggo version would be Cocker spaniel.

Hello Black russian terrier.

Hello Canaan dog.

Hello Black russian terrier.

Hello Mastiff.

6. Improvement

This model can be further improved in the future according to the following points:

- First of all, the current model only works for a single dog image and it can be achieved in the future for multiple dog recognition.

- The number of breeds classified by the dog can be increased from 133 to a greater number.