Machine Learning Engineer
Nanodegree

Capstone Proposal

Dog Breed Classifier with CNNs

Andreas Häuser

20th January 2020

Proposal

Contents

1 Domain Background

If dog owners are unsure about their dog's breed, a dog-breed classifier can help them identify their dog's breed. This would be especially helpful for inexperienced dog owners. If a dog owner have doubts about the breed before buying a dog, i.e. there is a discrepancy between what the seller says and what you remember about the breed, you can quickly confirm your doubts with a dog breed classifier or get rid of the world.
The same problem hit me two years ago when I got my first dog called JJ. The dog breeder claimed it was a Chihuahua and I assumed that for a few months. Only shortly before I wanted to fly from Mexico to Germany with JJ and the customs papers had already been issued for a Chihuahua, a veterinarian informed me that he was actually a miniature pinscher. The upshot was that JJ had to spend two weeks in a dog hotel in Mexico.

These people and also me can be helped by developing a picture classifier that appreciates dog breeds. This is possible via the image classification in the area of deep learning, which is part of machine learning. Image classification refers to a process in computer vision that can classify an image according to its visual content. A convolutional neural network, CNN for short, is preferably used here.
CNN are a specialized type of neural network for processing data that have the structure of a grid. Image classification with CNN saw the light of day in 1994 with Yann LeCun's LeNet5 and received a huge boost in 2012 with AlexNet by Alex Krizhevsky who won the ImageNet competition.[1]

2 Problem Statement

In this project it is important to build a data processing pipeline to classify real-world, user-supplied images. On the basis of a image of a dog, an algorithm is supposed to give an estimate of the breed of the dog. If the image of a person is given, the algorithm should reproduce the most similar dog breed. In addition, an accuracy of 60 percent or greater should be achieved.

The following therefore applies:

- It must be recognized on an image whether a person or a dog is present. If neither of the two is the case, an error message is issued
- The breed of the dog must be estimated
- An accuracy of 60 percent or greater should be achieved

3 Datasets and Inputs

The following datasets provided by Udacity will be used:

- The dog dataset[2] with 8351 dog images
- The human dataset[3] with 5750 human images

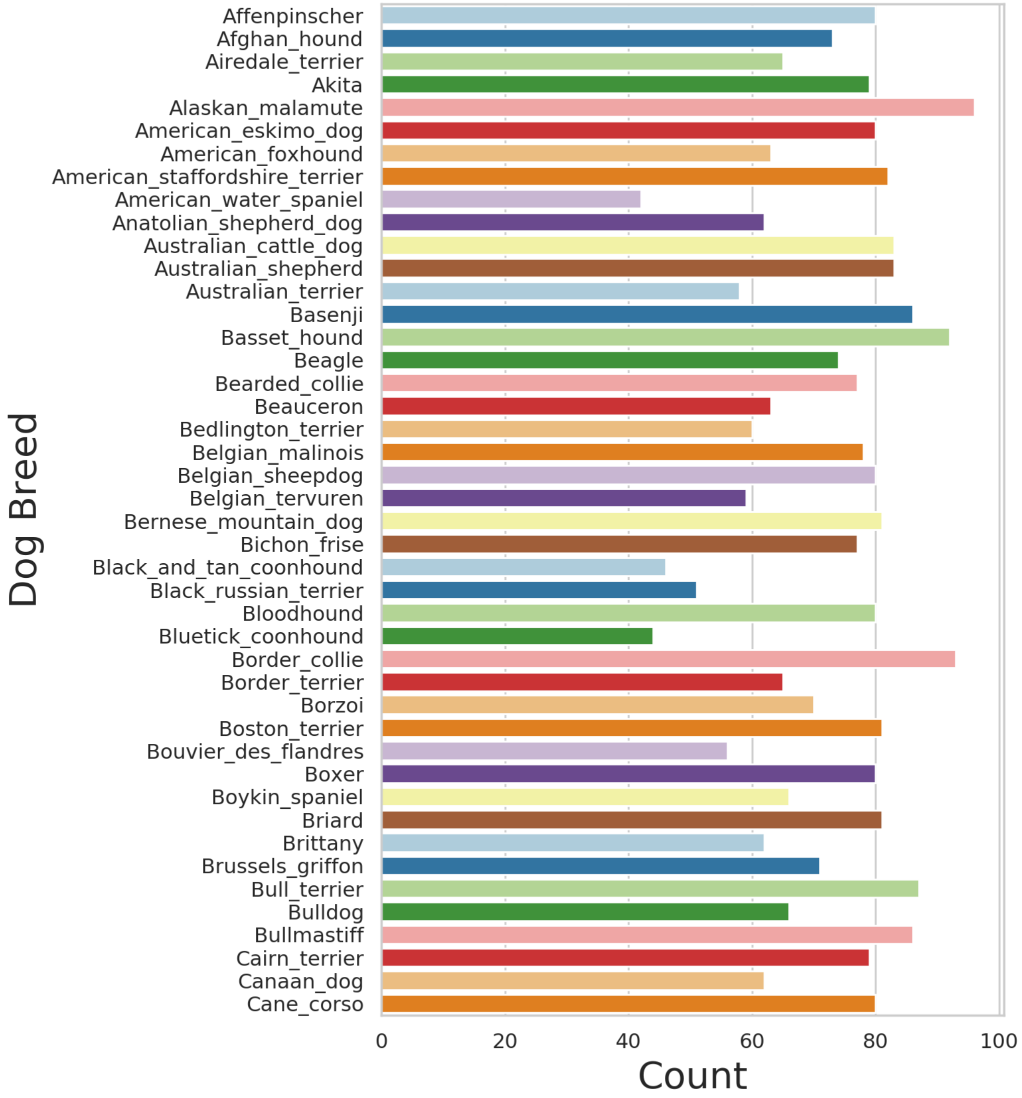
[1] Eugenio Culurciello. *The History of Neural Networks.* dataconomy.com. Apr. 19, 2017. URL: `https://dataconomy.com/2017/04/history-neural-networks/` (visited on 01/11/2020).

[2] Udacity. *The dog dataset.* URL: `https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/dogImages.zip` (visited on 01/10/2020).

[3] Udacity. *The human dataset.* URL: `https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/lfw.zip` (visited on 01/10/2020).

3.1 The dog dataset

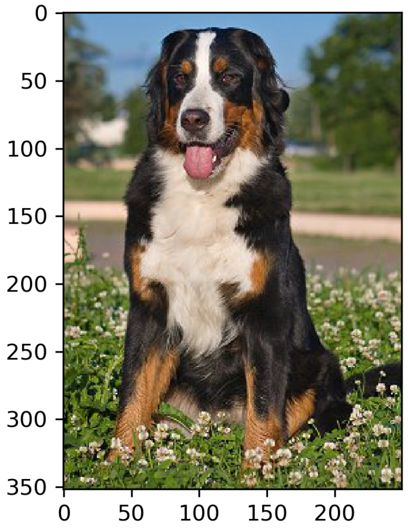
The data set contains 8351 dog pictures, which are subdivided into training, test and validation data. So we have 6,680 images to train the model, 836 images for testing and 835 images for validation. The entire data set contains 133 different breeds from the Afghan dog to the Yorkshire Terrier.
So that you can get an idea of what the dataset contains, i will show you a part of the distribution of the dog breeds over the complete dataset:



Dog Breed Distribution

Here are two examples of very nice dogs:

Chihuahua



Bernese Mountain Dog

3.2 The human dataset

This dataset contains 5750 ordinary pictures of more or less ordinary people. These images are used to test the performance of the Human Face Detector.

4 Solution Statement

The solution is to design a CNN model that is able to estimate as much as possible the breed of a dog that is included in a picture. To do this, it must be recognized beforehand whether a person or a dog is included in a picture. If a person is present, the resembling dog breed is identified. If a dog is present, an estimate of the dog breed is given.

5 Benchmark Model

I found an interesting benchmark model on the Internet. It can be viewed here[4]
On page 4 in Figure 5 the accuracy of different state of the art models is compared. So I choose the ResNet-50 + FT + DA shown in the table on page 4 as my benchmark model.

6 Evaluation Metrics

The goal here is to compare the performance of my model with that of the benchmark model. Therefore, I would use accuracy as an evaluation metric. Also because the benchmark model only specifies the accuracy.

7 Project Design

I choose the following strategy for the design project:

- Step 0 - Import datasets
  The data described under point 3 'Datasets and Inputs' are imported.

- Step 1 - Detect humans
  An algorithm will be developed that is able to recognize the faces of humans by using OpenCV and the integrated HaarCascade.

- Step 2 - Detect dogs
  The pre-trained VGG-16 model, which comes with the weights that were trained on the ImageNet dataset, can be used here to design a dog detector. With the Human Detector described in Step 1, I am already able to differentiate between humans and dogs.

- Step 3 - Create a CNN to Classify Dog Breeds from Scratch
  Here I will develop a CNN from scratch to classify dogs. This will be implemented using PyTorch.

[4] Aydin Ayanzadeh and Sahand Vahidnia. *Modified Deep Neural Networks for Dog Breeds Identification.* May 2018. URL: `https://www.researchgate.net/profile/Aydin_Ayanzadeh2/publication/325384896_Modified_Deep_Neural_Networks_for_Dog_Breeds_Identification/links/5cd0345ea6fdccc9dd90690c/Modified-Deep-Neural-Networks-for-Dog-Breeds-Identification.pdf` (visited on 01/20/2020).

- Step 4 - Create a CNN to Classify Dog Breeds using Transfer Learning
  Here I will use Transfer Learning to create a CNN to classify dogs. This will be also implemented with PyTorch. I will use the pre-trained ResNet50 model as a basis and by fine-tuning i will train a model with the dog pictures from our dataset. The resulting CNN should be better than the network developed under Step3 from scratch.

- Step 5 - Write my algorithm
  In this step I will put everything together. With a new image as input, I will first determine whether the image is a human or a dog. After that, I will return an estimate of the dog breed using the model from Step 4.

- Step 6 - Test my Algorithm
  In this section, I'm going to test my algorithm from step five with my own images.

References

Ayanzadeh, Aydin and Sahand Vahidnia. *Modified Deep Neural Networks for Dog Breeds Identification.* May 2018. URL: `https://www.researchgate.net/profile/Aydin_Ayanzadeh2/publication/325384896_Modified_Deep_Neural_Networks_for_Dog_Breeds_Identification/links/5cd0345ea6fdccc9dd90690c/Modified-Deep-Neural-Networks-for-Dog-Breeds-Identification.pdf` (visited on 01/20/2020).

Culurciello, Eugenio. *The History of Neural Networks.* dataconomy.com. Apr. 19, 2017. URL: `https://dataconomy.com/2017/04/history-neural-networks/` (visited on 01/11/2020).

Udacity. *The dog dataset.* URL: `https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/dogImages.zip` (visited on 01/10/2020).

— *The human dataset.* URL: `https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/lfw.zip` (visited on 01/10/2020).