Machine Learning Engineer Nanodegree Capstone Project

Dog Breed Classification

Burak Kılıçaslan

October 12th, 2020

Domain Background

In today's world, the subject of artificial intelligence is very popular and AI applications are followed by everyone. Researches and academic studies continue rapidly on this field to find new use-cases. One of the most popular AI algorithms is deep learning in recent years. With development of hardware technologies and deep learning algorithms, image and video processing applications become so popular and useful for many people and many companies. However, apply these algorithms and techniques are so challenging and complicated problems in real world.

One of these challenging applications is Dog breed classification. In this application, there are 133 different classes and all of them belong to dogs. Classifying different species such as dog and horse is relatively easy comparing to classifying same species breeds. For this reason working such this project and getting high performance is a tough problem. For image classifying or object detection applications, there are lots of different concepts and algorithms in literature to tackle with these problems. In this project, we will use Convolutional Neural Networks which is the state of art for image or video processing.

CNN provides better feature extraction for complicated problems. Its architecture is similar human brain. It contains lots of layers and lots of neurons in every layer. Every layer’s neurons are connected with next layer’s neurons and in training phase, this connection between neurons keep information about data or for our project images. This architecture allows to us build complicated image processing applications with high accuracy. On the other hand, this architecture contains lot of computational operation and this means that it needs high hardware performance. As we stated above, thanks to development on hardware technologies, GPU performances and cloud services, it is easy to provide requirements for this situation.

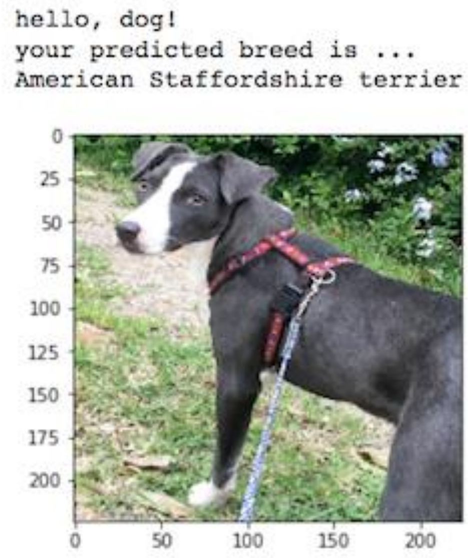
To sum up, in my Udacity Machine Learning Nanodegree Capstone Project, I follow the steps CNN architectures and try to create CNN model from scratch. This challenging project gives me chance to face difficulties on image classification applications and gain knowledge from these difficulties.

Problem Statements

In this project, we will build a pipeline to process real-world images. Provided an image by user, our algorithm will detect that, the image is human or dog. If image is dog than our algorithm will predict its breed. If the image is human than our algorithm will predict which breed of dog the human image looks like.

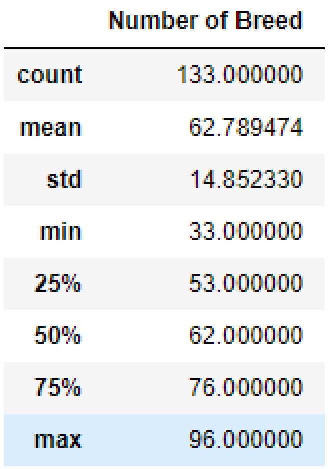
Dataset and Inputs

The dataset has been provided by Udacity. Dataset contains 13233 human images and 8351 dog images, which are separated as train, test and validation. Human images or lfw size is 180 MB. Dog image's size is 1.1 GB. Dog images have 133 different class and it is already labeled by Udacity. On the other hand, human dataset is ordered according to name. When we will train our model, images size will be cropped as 224x224 because of pytorch pre-trained model preparation and our CNN network. Our outputs will be firstly the image is human or not, if image is human, than identify the resembling dog breed, if image is dog than predict its breed. If image nor human or dog, system will be get an warning. You can see sample output as below.



```
hello, dog!
your predicted breed is ...
American Staffordshire terrier
```

As we said above, the dataset has 133 different classes. When we look into dataset more deeply, there are 62 number of image for every dataset on average. 'Norwegian Buhund' breed class has 33 number of images that are min value between all classes and 'Alaskan Malamute' has 96 number of images that are max value between all classes. You can see detailed information about distribution of the dataset as below.

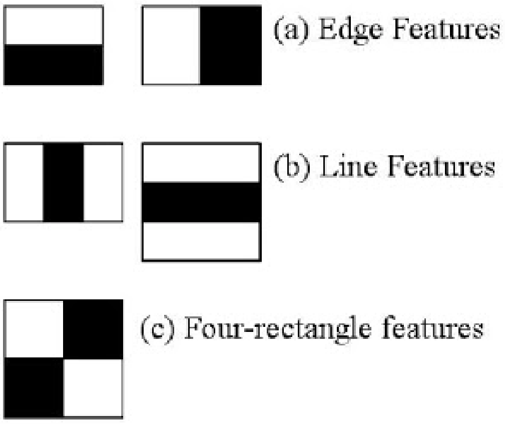


	Number of Breed
count	133.000000
mean	62.789474
std	14.852330
min	33.000000
25%	53.000000
50%	62.000000
75%	76.000000
max	96.000000

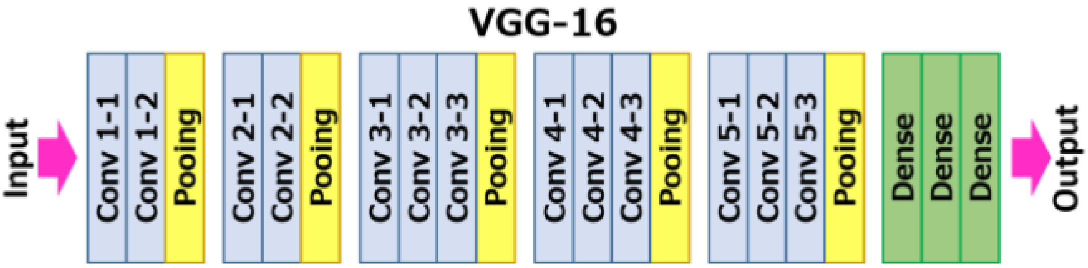
Solution Statements

In this project, firstly, we will detect the image is human or dog. We will use openCV's Haar feature-based cascade classifiers for human detection and pre-trained VGG-16 model for dog breeds classifying.

OpenCV haar cascade classifier and is proposed by Paul Viola and Micheal Jones in 2001. It is a machine learning method for object detection and used as effectively. It needs lots of true and false images to train. If we create a face detection than true images are images with face, false images are images without face. After that, feature extraction process starts from these images with sum of the pixels of the image location by location. Feature extraction processes based on three features which are line, edge and four-rectangle as you can see below clearly. Because calculating features for all possible locations need so much computation and there may be lots of irrelevant features on an image, Adaboost algorithm which is a learning algorithm to improve performance uses during feature extraction process. After feature extraction, best features are selected according to minimum error rate and classification process start. After every classification, based on the result weights are updated and this loop continue until model converge with optimum metric.



VGG-16 model is a simpler CNN architecture. It contains 16 layer which last three of them fully connected (dense) layer. It contains about 138 million parameter computation and gives output from dense layer about 1000 classes which is calculated with softmax activation function. Softmax activation function gives output between 0 and 1, so it uses as very popular for classification problems. Inputs of this model 224x224 pixel. You can see the model architecture clearly as below.



In this project, we will create a CNN model from scratch and train our model. The expected accuracy result for this model should be at least %10. Actually %10 accuracy is very low however image classification is challenging and hard task. It needs so many data and very complex model with long training sessions. We can see that it is so complicated to get high accuracy results immediately with creating basic CNN models.

After our CNN model, we will use transfer learning to improve our CNN. The improved model test accuracy should be at least %60.

At the end of our project, we will write an algorithm which takes sample image and gives the image is human or dog and if the image is dog, gives the breed of dog.

Benchmark Model

When we will create CNN model from scratch, we will try lot so of different hyper parameter and different model architecture. This trials will be shown of the final project. Besides, we will compare our CNN from scratch results and transfer learning using Res-Net results.

Evaluation Metrics

Accuracy metric will be used on all steps of this project. Besides, Udacity reviewers evaluate my project according to accuracy metric as mentioned above. Because our dataset is balanced fairly, accuracy metric can be used for evaluation.

Project Design

We show our project steps as below.

- Step 0: Import Datasets
    - Because we will use udacity workspace, we will not need to download dataset in our local machine. Dataset has been already uploaded by Udacity in the workspace. We just need to load and read data from workspace folder.
- Step 1: Detect Humans with OpenCV Haar feature-based cascade classifier

  - We will use the OpenCV Haar feature-based cascade classifier which is explained detailed on solution statement section. We will import this classifier using OpenCV and give outputs on human images.
- Step 2: Detect Dogs with pre-trained VGG-16 model
  - Firstly, we will use VGG-16 pre-trained model to detect dog breeds. We just load the model to our workspace and using the model, test our images.
- Step 3: Create a CNN to Classify Dog Breeds (from Scratch)
  - This is one of the most challenging part in the project. CNN models are so complicated and we will try to build a CNN model from scratch and train this model with our dataset. We will try to reach at least %10 accuracy and to do that we should tune our hyper parameter and model architecture.
- Step 4: Create a CNN to Classify Dog Breeds (using Transfer Learning)
  - Because getting high accuracy results with creating and training a CNN model from scratch is really tough problem, we will use transfer learning which is that training the model with using a pre-trained model weights and features. So, we can increase our accuracy rate.
- Step 5: Write final Algorithm
  - We will write an algorithm which takes the images from users and gives the results. This algorithm contains everything that we need. It will read and load the image that user upload, calls required functions to identify objects, based on the result decide what it prints.
- Step 6: Test final Algorithm and results
  - Testing final algorithm and whole project codes.

References

1. https://github.com/udacity/machine-learning/blob/master/projects/capstone/report-example-1.pdf
2. https://github.com/udacity/deep-learning-v2-pytorch/tree/master/project-dog-classification
3. Hsu, David. “Using Convolutional Neural Networks to Classify Dog Breeds”, Stanford University. http://cs231n.stanford.edu/reports/2015/pdfs/fcdh_FinalReport.pdf
4. Karen Simonyan, Andrew Zisserman “Very Deep Convolutional Networks for Large-Scale Image Recognition” Cornell University https://arxiv.org/abs/1409.1556
5. LFW Dataset, Udacity https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/lfw.zip
6. Dog Image Dataset, Udacity https://s3-us-west-1.amazonaws.com/udacity-aind/dogproject/dogImages.zip
7. https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html
8. https://medium.com/@ayyucekizrak/derine-daha-derine-evrişimli-sinir-ağları-2813a2c8b2a9