Capstone Project Kushank Sriraj

Machine Learning Engineer Nanodegree October 30th, 2020

# CNN Project: Dog Breed Classifier

## Project Overview

The dog breed classification is a very popular problem in the computer vision domain. The task is to classify a dog breed from 133 different breeds. This can be done by supervised learning, a type of machine learning that deals with labeled data.
In this project, a pipeline has been built to process real-world, user-supplied images. Given an image of a dog, the algorithm will identify an estimate of the canine's breed. If supplied an image of a human, the code will identify the resembling dog breed.

This project uses Convolutional Neural Network(CNN) to perform multi-class image classification. Plenty of research has been done in the field of image classification and CNN is the go-to model for this task.
Further, technologies like OpenCV for face detection and transfer learning have been used which are described later in this report.
The datasets have been provided by Udacity for the project.

## Problem Statement

Problems:

1. To predict the dog's breed given an image of the dog. This is possible by supervised learning.

2. If supplied an image of a human, predict a resembling dog breed. For human face detection, OpenCV's haar cascade classifier is used.

This project leverages the power of transfer learning to achieve better accuracy even from a small dataset. The strategy is to use a pretrained model like VGG-16, ResNet, etc. and modify them according to the requirements of the project.

## Metrics

The metrics used to evaluate the model is accuracy and log loss.
Accuracy is defined as the *number of correct predictions divided by the total number of predictions, multiplied by 100*. It is the most commonly used metric and the simplest of all.

$$\text{Accuracy} = \frac{\textit{correct predictions}}{\textit{all predictions}}$$

Image source : Google Images

The other metric, log loss or cross entropy loss is used because this is a problem of multiclass classification.

$$L = -\frac{1}{m}\sum_{i=1}^{m}(y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i))$$

Image source[1]

The test and validation datasets are used to compute the metrics. The final model was able to attain an accuracy of 83%.

## Data Exploration

The datasets : dog dataset and human dataset required for this project is provided by Udacity. Since this is a project on image classification, the input required is of the image type.

Details of the datasets :

1. **Dog Image dataset** : It contains images of various dog breeds. The dataset is not balanced. It contains a total of 8351 images of dogs of different breeds.

   The dataset is divided into 3 directories: train, test and valid. These directories contain images in unequal numbers :

   - The *train* folder contains 6680 images.
   - The *test* folder contains 836 images.
   - The *valid* folder contains 835 images.

   All the three directories have 133 folders corresponding to 133 dog breeds, the images are present inside these folders.

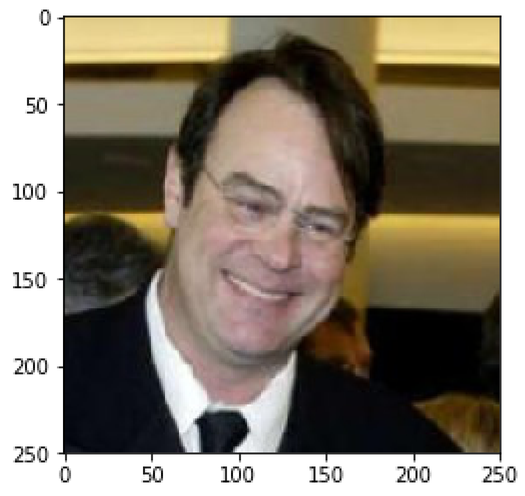
   The images are of different sizes(image resolution) and the number of images vary for each breed.

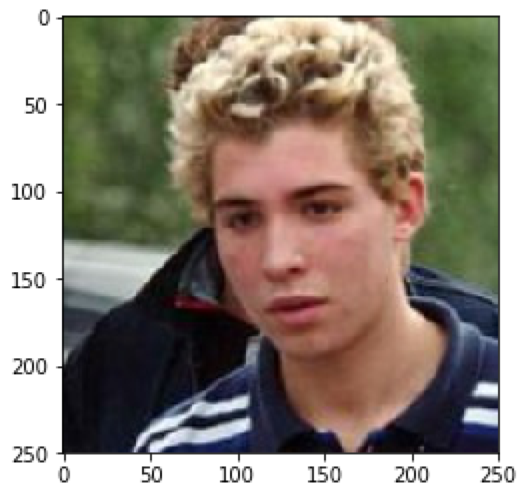
   Some sample images :

2. **Human Image dataset** : It contains images of human faces sorted by name. This dataset is also not balanced.

    - It contains a total of 13233 images.
    - All the images are of 250x250 resolution.
    - The dataset contains a total of 5750 folders which are named after human names in a sorted manner. Each of these folders contain an unequal number of images.
    - Some folders contain only one image while others have many.

Some sample images :

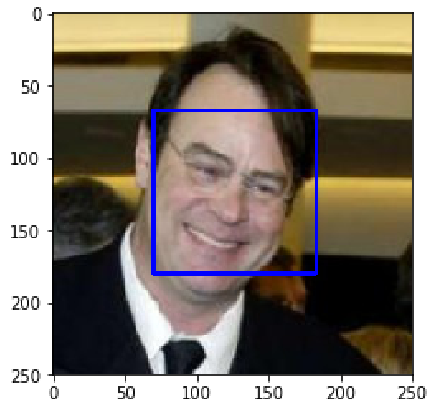




## Exploratory Visualization

The dog and human face datasets are interesting and are visualized when necessary.

Example : Human face detection using OpenCV's haar cascade classifier



## Algorithms and Techniques

Since the problem is of image classification, the state of the art algorithm, best suited for this task is **Convolutional Neural Network(CNN)**.

**CNN** is one kind of a feedforward neural network**.** It is an efficient image recognition algorithm which is widely used in pattern recognition and image processing. It has many features such as simple structure, less training parameters and adaptability.
The main advantage of CNN compared other models is that it automatically detects the important features without any human supervision.

**Transfer learning** is a machine learning method where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems.
Pretrained models such as VGG-16 and ResNet are used in this project to do multiclass classification of dog breeds. Transfer learning is used for this purpose.

**OpenCV** (Open Source Computer Vision Library) is an open-source BSD-licensed library that includes several hundreds of computer vision algorithms.
For face detection, OpenCV's **Haar feature-based cascade classifier** is used.

**Pytorch** is used to build the neural network from scratch. It is a popular machine learning library for python. It is fast and efficient for experimentations.

## Benchmark

**The Model built from scratch** must attain an accuracy of 10%. This will establish that the model is working because a random guess will provide a correct answer roughly 1 in 133 times, which corresponds to an accuracy of less than 1%.

**The model built from transfer learning** must attain at least 60% accuracy on the test set. The pretrained model(ResNet50) is already proved to be good at image classification and hence the model made from transfer learning must attain this minimum threshold to prove that it is working

## Data Preprocessing

The images in the datasets have been preprocessed to meet the requirements of the model.

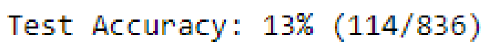
1. All the images have been resized to 224x244 resolution as this is the default size required for the VGG-16 model

2. The images in the train set have been augmented with random horizontal flip to reduce overfitting and improve accuracy.

3. The images have been converted to tensors and normalized.

## Implementation

Firstly, the model was made from scratch with 3 convolutional layers and a dropout layer to prevent overfitting. The model was trained for 15 epochs and the loss function used was CrossEntropyLoss().
This model resulted in 13% accuracy.



```
Test Accuracy: 13% (114/836)
```

Then, the pretrained model ResNet50 was employed and using transfer learning, this model was modified to produce output of the required dimension.

```
model_transfer.fc = nn.Linear(2048, 133, bias=True)
```

The loss function used was CrossEntropyLoss() and Stochastic Gradient Descent(SGD) optimizer was used.

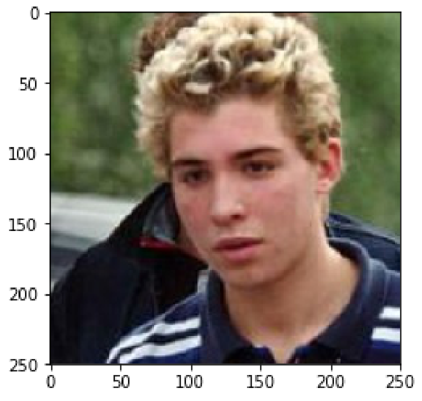
After training for 5 epochs, this model resulted in accuracy of 83%.

```
Test Loss: 0.674245


Test Accuracy: 83% (701/836)
```

After this, the model was used to predict dog breeds resembling human faces.

```
Human here :)
Similar dog breed : American water spaniel
```



.

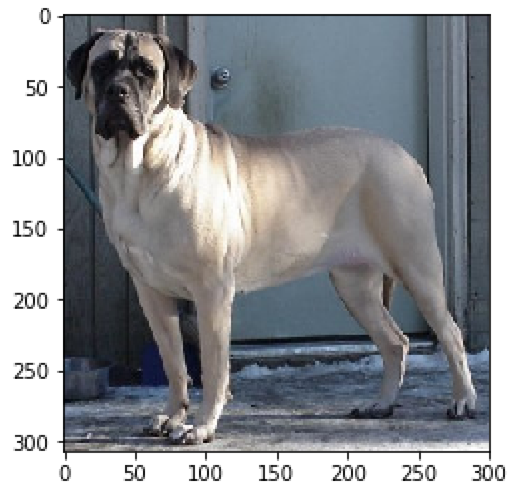
## Refinement

The CNN model created from scratch had an accuracy of 13% only.

After using transfer learning with the pretrained model, the accuracy went up to 83% within 5 epochs. This significant increase in accuracy is because of using the ResNet50 model.

Sample prediction:

```
Dog here :)
Dog's breed : Mastiff
```



## Model Evaluation and Validation

**The model built from scratch** was able to attain an accuracy of 13%. This is acceptable compared to the benchmark accuracy of 10%.
This confirms that the model built from scratch is working.

**The model built from transfer learning** scored an accuracy of 83% after just 5 epochs. This is significantly better than the benchmark of 60%.
This proves that the model is working. An accuracy above 80% is very impressive.

Evaluation metrics of other algorithms :

1. Human Face Detector :
   - 98% of the first 100 images in human_files have a detected human face.
   - 17% of the first 100 images in dog_files have a detected human face.

2. Dog Detector :
   - 100% of the images in dog_files_short have a detected dog.
   - 0% images in human_files_short have a detected dog.

## Justification

The model is performing very well compared to the benchmark model. The accuracy of 83% shows that the model is generalized and can also perform well on images not in the dataset.

However, the model can be further improved by :

1. Using a deeper pretrained model like ResNet101, ResNet152
2. Doing more image augmentations
3. Using models of different architecture like MobileNetv2

### References

1. Links to datasets :
   https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/dogImages.zip
   http://vis-www.cs.umass.edu/lfw/lfw.tgz
2. Udacity's github repo :
   https://github.com/udacity/deep-learning-v2-pytorch/tree/master/project-dog-classification
3. ResNet : https://pytorch.org/hub/pytorch_vision_resnet/
4. CNN : http://deeplearning.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/
5. Transfer Learning : https://machinelearningmastery.com/transfer-learning-for-deep-learning/
6. OpenCV and haar cascade classifier: http://opencv.org/, https://docs.opencv.org/4.5.0/db/d28/tutorial_cascade_classifier.html
7. Image source[1] : https://levelup.gitconnected.com/grokking-the-cross-entropy-loss-cda6eb9ec307