UDACITY CAPSTONE PROJECT

ARJUN MAINI

Machine Learning Engineer Nanodegree May, 2020

CNN Project : DOG BREED CLASSIFIER

PROJECT OVERVIEW

Since childhood I have always been fascinated with the different breeds of dogs. Using Convolutional Neural Network (CNN) with Transfer Learning, we will be building a pipeline of layers that can process a real world user supplied image (human/dog) and tell us the breed of dog in that input image. Object Identification and image classification are the two crucial steps of this multi-class classification project

PROBLEM STATEMENT

We'll be given an image of either a dog or a human. If it is a dog's image, we need to identify the breed of the dog and if it is a human's image, we need to identify which dog breed does the human resemble to. We'll be using supervised learning in this classification problem and transfer learning to speed up the training process and accelerate the rate at which we produce results.

METRICS

I have chosen accuracy as the metric for evaluating my model.

Accuracy = Correctly predicted items / Total no. of classified models

Observed accuracy of CNN (From scratch) : 16%

DATA EXPLORATION

All the data required for this project is provided to us by Udacity. The dataset for dogs is already divided into train, test and valid sets with 133 folders in each set directory of 133 different breeds of dogs. The dataset for humans is also provided sorted by the names. We'll be downloading the data set, unzipping it and then converting the images to NumPy arrays of required pixels.

Dog dataset has a total of 8351 images with different backgrounds, from various angles and lighting which is good for the accuracy of our model since we don't know how the provided user input will be. It is also non-uniform i.e. the number of images of different breeds in the folders isn't the same. This dataset has different image sizes so we would resize all to the same size (say 256 x 256).

Dog train dataset : 6680 images
Dog test dataset : 836 images
Dog valid dataset : 835 images

Human dataset has a total of 13233 images (5749 folders) all of the same size but non uniformity i.e. different number of images of different humans. All human images are of same dimensions i.e. 250 x 250.

The **input** data should be of image type and we want user to supply us the image which would further be sent to our model to obtain the required prediction.

ALGORITHMS & TECHNIQUES

- For the purpose of solving this multi-class classification problem, first I am using pre trained detectors for detecting whether it's a dog or a human.(VGG-16 for dog detection and OpenCV Haar-cascade for human detection)

- I'll be first creating a Convolutional Neural Networks from scratch comprising of convolutional layers followed by max pooling 2D layers (to reduce the dimensional complexity) which will then be converted to scalar and then the scalar will be used to output us the dog breed through SoftMax.

- After that, I will be creating a CNN using transfer learning with ResNet-50 and fine tuning it for obtaining predictions with an aim to get an improved accuracy.

BENCHMARK

We first create a CNN model from scratch. An accuracy of at least 10% would ensure a decent benchmark for the purpose of breed detection given the amount of variability in the data provided. We need to predict 1 out of 133 breeds which is <1% so a 10% accuracy is a good benchmark for the model.

DATA PREPROCESSING

- I chose **224*224 size for input tensor** as it is the default input size for VGG models and used in ImageNet.

- I have applied Random-Resize-Crop, Horizontal Flip and Random-Rotation to the training dataset to bring in the randomness which improves accuracy of the model. So, I have applied Data Augmentation to training dataset

- For valid/test dataset, I haven't used any Data augmentation apart from resize and crop transforms.

- All the training, testing and validating datasets were converted to tensors before passing them to the model.

IMPLEMENTATION

Firstly I created a CNN model from scratch. It had the following architecture :

- **3 convolution (conv2d) layers** with kernel size = 3 and stride = 1.
- First conv layer takes an input of size 224*224 and third conv layer produced an output of 128
- **Max pooling layers** of pool size =2 to reduce input size by 2 times.
- **2 Fully connected layers** followed by soft max that produced the final 133 output neurons
- A **Dropout** of 0.30 to prevent overfitting.
- A total of 133 output neurons in the final soft max layer to predict the breed.

REFINEMENT

The CNN model created from scratch had an accuracy of 16 %. I used 20 epochs for this model.

To refine this accuracy metric, we further used transfer learning with **ResNet50** model to create another CNN model for out project.

ResNet 50 is a pre trained model used extensively for image classification problems and has everything starting from convolutional layers to full connected layers pre built. So it takes care of everything about feature extraction too.

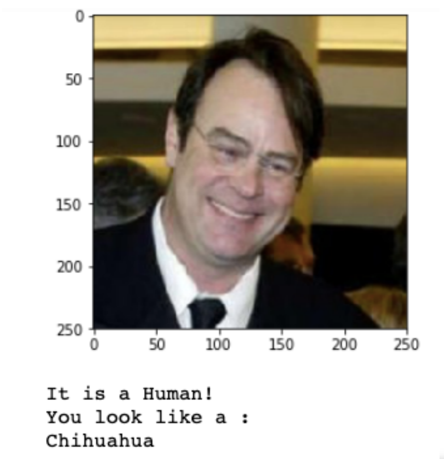
I only had to change the final fully connected layer gives an output of 64 neurons by default. So, I had to fine tune it to give output of 133 (breeds) neurons. I used 20 epochs.

We noticed that the accuracy improved significantly. **The CNN model created using ResNet50 gave an accuracy of 83% which is a big improvement.**

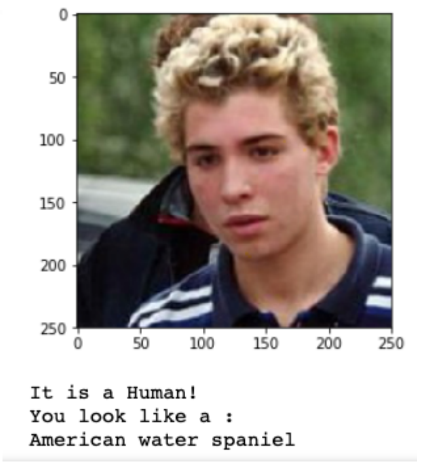
MODEL EVALUATION & VALIDATION

The output produced by our model looks like this :

For human images supplied to our model :

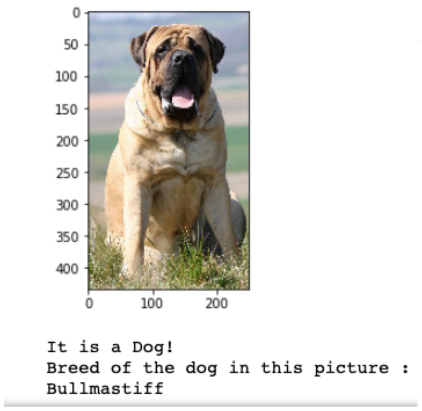


```
It is a Human!
You look like a :
Chihuahua
```

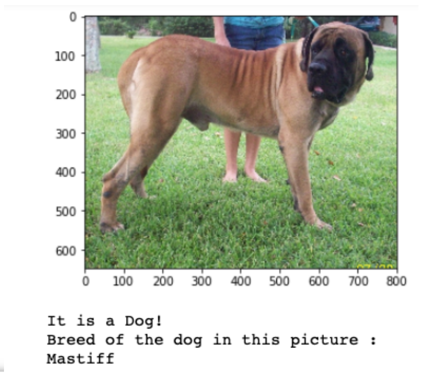


```
It is a Human!
You look like a :
American water spaniel
```

For dog images supplied to our model :



```
It is a Dog!
Breed of the dog in this picture :
Bullmastiff
```



```
It is a Dog!
Breed of the dog in this picture :
Mastiff
```

Human face detector : Made using open CV's implementation of Haar feature based cascade classifier. 98.00% of the first 100 images in human files detected human face and 17.00% of the first 100 images in dog files detected a human face

Dog detector : Made using a pre trained VGG 16 model on ImageNet database. 0.00% of the images in human files short had a detected dog. 96.00% of the images in dog files short had a detected dog.

CNN using scratch : Made using 3 conv layers, max pool layers, 2 fully connected layers and a dropout layer. I used 20 epochs to train it. It gave an accuracy of 16%.

Accuracy = 134/836. (Correctly predicted for 134 images from test dataset)

CNN using Transfer Learning : Made using pre trained ResNet50 model and trained on 20 epochs. It gave an accuracy of 83%.

Accuracy = 694/836. (Correctly predicted for 694 images from test dataset)

JUSTIFICATION

Compared to the 10% benchmark set up by me, the model performed significantly well as it gave an accuracy of 16%.

IMPROVEMENT

Yes, the output from the mode using transfer learning is much better and it gave an accuracy of 83%. We can improve our model even more by incorporating a larger data set for training, more data augmentation applied on training dataset so that any image could give a positive result and even use hyper-parameters for tuning our model.

REFERENCES

GitHub repository
https://github.com/udacity/deep-learning-v2-pytorch/tree/master/project-dog-classification

ResNet50
https://www.mathworks.com/help/deeplearning/ref/resnet50.html

CNN Guide
https://www.analyticsvidhya.com/blog/2018/12/guide-convolutional-neural-network-cnn/