Machine Learning Engineer Nanodegree Shuyi Huo

Capstone project report June 13th, 2020

CNN project: Dog Breed Classifier

Project Overciew:

Dog bread classifier is a popular image classification problem in the computer vision domain. There are two major tasks in this project. First one is to identify the image given from input is a dog face or human face. Second one is to identify the breed of dog if the input image is dog or identify the resembling dog breed if the input image is human. The main vision is to build a Machine learning pipeline that will take the user-supplied images from real world and output the result of classification. We will use supervised learning in this project to solve the problem. Further, building a web app would be feasible for users to input the image an then get the predicted output.

Problem statement:

The goal of project is building a machine learning model to predict the dog breed or resembling dog breed giving a human face. Two main tasks are human face detector and dog breed detector.

Datasets and inputs:

Dataset is given by Udacity and there are two datasets: human image and dog image. The human images are distributed in 5749 folders named like human name. There are total 13233 human face image in human image folder but images are not evenly distributed in the folders. The dog image contains three folders: train, test and valid. The train folder contains 6680 images for training divided in 133 dog breed classes. The valid and test folder contain 835 and 836 image for validation and testing. Thus, there are total 8351 dog images in dog image folder. The input of project would be an image. For example, the two images below are from human files and dog image file.

Solution statement:

The problem is a typical multi-class classification in visual recognition and could be solve using Convolutional Neural Networks(CNN) which is a deep learning algorithm. CNN could extract the feature efficiently from the images based on the filters of weight and biases. For human face detection, I used the existing algorithm: OpenCV's implementation of Haar feature-based cascade classifiers to detect human faces in images. For the dog detector, I used the pre-trained VGG-16 model to detect dogs in images. For Breed classification, I created a CNN model and also used pre-trained Resnet-101 model to process the images. Then comparing the accuracy and loss of both models, choose the best model as the final model to predict the images.

Benchmark model:

The VGG-16 model used in transfer learning for dog detector should predict dog with at least 80% precision in image because it makes sure the dog detector method is well-trained. The custom CNN method have to get some accuracy higher than at least 10% precision to cinfirm the model is work. If working, it is able to output the only one dog breed.

Data Processing:

All the images are resized to 224*224, then normalization is applied to all images includes training, validation and test datasets. For the training dataset, image are randomly rotated and randomly horizontal flipped. Then all the images are converted into tensor before sending them to the model.

Evaluation metrics:

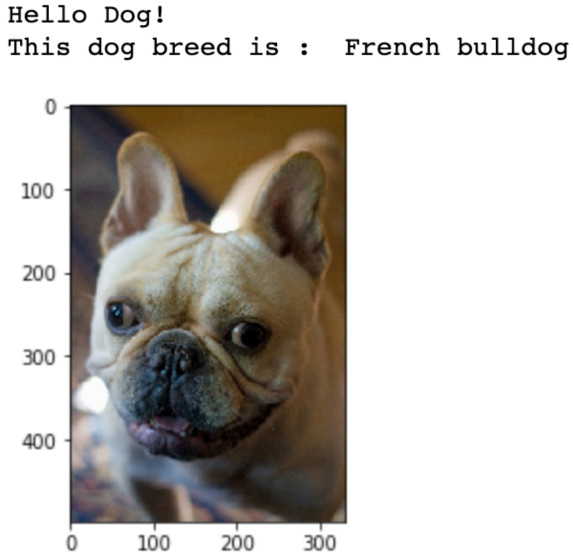
I used accuracy to evaluate the model.

- accuracy = number of items correctly classified /all classified items.

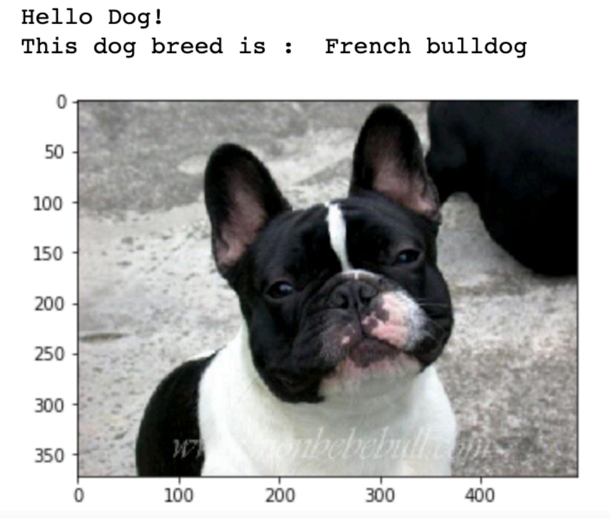
Then we compare the test dataset prediction with validation dataset and compute the multi-class log loss for evaluating metric to find best performing model..

Refinement:

The custom CNN model form scratch has accuracy of 18% (151/836) and the loss is 3.4669. Although this model satisfy the benchmarking, the model can be improved more by using transfer learning. To create CNN from transfer learning, I used the Resnet101 which has 101 layers deep and pertained on ImageNet dataset. The performance of model is extremely better then custom CNN from scratch because it has 84% accuracy and 0.5704 loss.



```
Hello Dog!
This dog breed is :  French bulldog
```



```
Hello Dog!
This dog breed is :  French bulldog
```

Model evaluation:

The human face detector function was created using OpenCV's implementation of haar feature cascade classifier. The percentage of the first 500 images in human file have a detected human face is 98.6%. The percentage of the first 500 images in dog file have a detected dog face is 10.8%. The dog breed detector function was created using pre-trained VGG-16 model.The percentage of the image in human_files_files have a detected dog is 0.0. The percentage of the image in dog_files_files have a detected dog is 0.97. The CNN model from transfer learning was created using ResNet101 model with 5 epochs. The accuracy is 84% and the loss is 0.5704. The model correctly predicted the 709 dog images out of 836 total images.

Improvement:

The model could be improved by adding more training dataset and test dataset and applied more powerful and efficient models and then boosting them to choose the best model for detection. Hyperparameter tuning is also a good practice to improve the performance of model.

Reference:

1. Original repo for project: https://github.com/udacity/deep-learning-v2-pytorch/tree/master/project-dog-classification
2. Resnet101: https://pytorch.org/hub/pytorch_vision_resnet/
3. PyTorch document: https://pytorch.org/docs/stable/data.html#torch.utils.data.DataLoader