Capstone Project: Machine Learning Engineer Nanodegree

Dog Breed Classifier using CNN

Domain Background

Image classification is one of the fundamental problems in field of Machine Learning and Deep Learning. This forms core of several computer vision and real-world image processing problems. The objective is to accurately predict a breed of dog given an image as an input. If input is an image of a human being then the classifier to predict the label that closely resembles a canine breed of the dog. This model helps me to create classifier that can be deployed and integrated any web or mobile application to classify real world images.

Problem Statement

The primary goal of this project is to create a multi-class classifier using CNN. The trained model will predict a score that indicates the probability of an input image belonging to a specific breed of a dog. The project includes 2 main tasks:

1. Create a Dog detector that will predict the estimator of a canine's breed.
2. Create a human face detector that predicts resembling breed of the dog.

Data Exploration

There are 2 datasets used in the project which are as follows:

Dog images dataset: The dog image dataset has 8351 total images which are sorted into train (6,680 Images), test (836 Images) and valid (835 Images) directories. Each of this directory (train, test, valid) have 133 folders corresponding to dog breeds. The data is not balanced because the number of images provided for each breed varies.

Human images dataset: The human dataset contains 13233 total human images which are sorted by names of human (5750 folders). All images are of size 250x250. Images have different background and different angles. This dataset is also not balanced.

Data Pre-processing

Following are the steps performed involved in data pre-processing:

1. Resize the images in the dataset to fixed size 224*224 pixels.
2. Perform augmentation of the images to include variation in the dataset.
3. Normalization the pixel values of the images.
4. Convert the images into appropriate tensor the feed in the model for training.

Algorithms and Techniques

For solving this multiclass problem, Convolutional Neural Network (CNN) is used which is a state-of-the-art algorithm for image classification problem. The approach include following includes 3 steps:

1. Detect human image using OpenCV which in turn uses a haar cascade classifier.
2. Detect dog image using a pretrained CNN model.
3. Train custom model to predict breed of the dog.
4. Finally, train a CNN model using transfer and compare the results.

Implementation

The architecture of custom CNN model is as follows:

- First Layer is the Convolutional layer which takes image as the input.
- Then there two 2 convolutional layers followed by maxpooling layer.
- Each layer uses Relu as the activation function.
- Then there are two fully connected layer.

Further, dropout hyperparameter is used to avoid overfitting. However, this custom CNN model performs poorly, and accuracy obtained is just 18%. To improve performance transfer learning technique is used.

Refinement

For transfer learning, Resnet101 architecture is used and initial layers are fined tuned and extra two fully connected layers are added on top of this architecture. The, this model was trained for 5 epochs and there is significant improvement in the result. The accuracy obtained is 80%. This really shows that if we have small dataset it is better to used transfer learning technique than training the custom CNN model.

Model Evaluation

For training the model cross entropy loss function and stochastic gradient descent is used. Further, 3 models are used in this project which are as follows:

1. Human Face Detector: Its used OpenCV's haar cascade classifier to detect face of human from an input image. The accuracy obtained for human dataset was 98% and for dog dataset it was 17%.
2. Dog Classifier: A pretrained VGG16 model was used to classify the input image as dog or not. The accuracy of this model was 100% for dog dataset and 1% for human dataset. This clearly shows that this model is much more robust than human face detector.
3. Resnet101 CNN model: Transfer learning technique was used to create a custom dog breed classifier and the accuracy obtained was 80% which far better than training a CNN model from scratch.

Future Work and Improvements

The final model had an accuracy of 80%. For a small dataset, the results are pretty good. But there is still room for improvement. More training data is needed to learn the complex features that can be used in differentiating breed of the dogs. Variation can also be included by increasing the rate of augmentation. Moreover, results of several CNN architecture can be compared to select the best model.

References

1. Original repo for Project - GitHub: https://github.com/udacity/deep-learning-v2pytorch/blob/master/project-dog-classification/

2. Resnet101:
https://pytorch.org/docs/stable/_modules/torchvision/models/resnet.html#resnet101

3. Imagenet training in Pytorch:
https://github.com/pytorch/examples/blob/97304e232807082c2e7b54c597615dc0ad8f6173/imagenet/main.py#L197-L198

4. Pytorch Documentation: https://pytorch.org/docs/master/

5. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neuralnetworks-the-eli5-way-3bd2b1164a53

6. http://wiki.fast.ai/index.php/Log_Loss