UDACITY

Machine Learning Engineer Nanodegree

Capstone Project Proposal

Dog Breed Classifier (CNN)

Vishwajeet Ghatage

June 2nd, 2020

Vishwajeet Ghatage
https://github.com/cloud-VG

Domain Background:

The Dog Breed Identification is popular project in ML/AI community. “Kaggle” also held a competition for this problem. The goal was to build a model capable of doing breed classification of a dog by just “looking” into its image. This problem can be solved by using frameworks like TensorFlow, PyTorch or using a high level approach with Keras.

Here, the capstone project I selected for Machine Learning Engineer Nanodegree is of the same type, and I will be solving it by building two models of Convolutional Neural Network (CNN) in PyTorch, one from scratch and one by Transfer Learning approach using pre trained model.

Problem Statement:

The goal of this project is to create a Convolutional Neural Network, which can be accessed by an endpoint or API to process real-time user provided images and complete following two tasks,

1. **Dog breed identification**: If a dog is present in the image, identify its breed.
2. **Human identification**: Identify human if present and predict the resembling dog breed.

Datasets and Inputs:

Input data for training will be an image either containing dog or human. The data is provided by Udacity containing both dog and human images.

Images of Dogs are arranged in three sub directories each for training, validation and testing. Where there are total 8,351 images, 6,680 for training, 835 for validation and 836 for testing. There are total 133 dog breeds provided in dataset, with images having different dimensions. The data is imbalanced as the distribution for each breed is not same.

Similarly, there are 13,233 images of 5,750 people. These images are of size 250 x 250, and these are also not balanced as there are more images of some people comparing to others.

Solution Statement:

The solution include following steps:

1. **Human detection:** We'll use OpenCV's Haar features based cascade classifier.
2. **Dog detection:** For detecting a dog image, we will use a pre trained VGG-16 model.
3. **Breed Identification:** Finally we will pass the image to a CNN built from scratch, to identify the breed.

Benchmark Model:

1. We will build a CNN with trial and error approach, tuning its hyper parameters step by step. The model needs to have accuracy in range at least 10-20% after some epochs of training.
2. The models, created by transfer learning needs to have 70% or more accuracy.

Evaluation Metrics:

As our training data is imbalanced, it will be better to go with "Categorical Cross Entropy" or "Multi class Log Loss". As it will comparing actual and predicted labels for loss calculation, it'll be a good metric.

$$F = -1/N \sum_i^n \sum_j^m y_{ij} * \ln(p_{ij})$$

Project Design:

Step 1: Import required libraries, load and preprocess data, create train, validation and test data sets.

Step 2: Detect if the image contains human using OpenCV's cascaded classifier.

Step 3: Detect if the image contains dog using pre trained VGG-16 model.

Step 4: Creating a CNN from scratch, training and evaluating it to identify dog breeds.

Step 5: Creating a CNN using ResNet101 pre trained model with transfer learning approach for breed identification.

Step 6: Finally combining these models, to accomplish following tasks,

1. If dog is in the image, return its breed.
2. If human is in the image, return resembling dog breed.
3. If image doesn't contain dog/human the output will indicate error.

References:

1. **Udacity's GitHub repository:**
   https://github.com/udacity/deep-learning-v2-pytorch/tree/master/project-dog-classification

2. **PyTorch documentation:**
   https://pytorch.org/docs/master/

3. **Multi class log loss:**
   https://stats.stackexchange.com/questions/113301/multi-class-logarithmic-loss-function-per-class

4. **Article on solution to similar problem:**
   https://towardsdatascience.com/dog-breed-classification-hands-on-approach-b5e4f88c333e