Capstone Proposal AbdElrhman Mohamed

Machine learning Nanodegree March 30th, 2020

Dog Breed Classifier

Domain Background

The breed classifier is one of the most famous problems in machine learning. The problem we want to solve is making a model who can identify the the dog and it's breed in a given image or human, this is a multi-class classification where we can use supervised machine learning to solve this problem as we have well labeled data, I would love to use this problem in web application.

Problem statement

The project goal is to build a machine learning model that can be used within web application to process real world images uploaded by the user to identify either:

- if a **dog** is detected in the image, return the predicted breed.
- if a **human** is detected in the image, return the resembling dog breed.
- if **neither human nor dog** is detected in the image, provide output that indicates an error.

Datasets and inputs

The data used in the project is provided by Udacity. The dataset has pictures of dogs and humans. Dog images dataset:

- There are 13233 total human images.
  - With 5750 sorted by human names
  - All the images are 250X250
  - Have different angels and different backgrounds.
- There are 8351 total dog images with 133 classes.
  - Train images 6,680
  - Test images 836
  - Valid images are 835

The data will be processed as the following:

- I've applied RandomResizedCrop & RandomHorizontalFlip to just train_data. This for doing both image augmentations and resizing jobs.
- Image augmentation randomize the dataset to prevent overfitting.
- I've Resized the images to be **256px**
- then center crop for images 224 X 224.

Solution Statement

Convolutional Neural Network (CNN) is the best in solving image multiclass detection to solve the problem.

The CNN deep learning, a convolutional neural network is a class of deep neural networks, most commonly applied to analyzing visual imagery. They are also known as shift invariant or space invariant artificial neural networks, based on their shared-weights architecture and translation invariance characteristics

Firstly, OpenCV provides many pre-trained face detectors, stored as XML files on github. implementation of same feature based cascade classifiers.

Secondly, to detect dog-images we will use a pretrained VGG16 model and resnet50 model.

Thirdly, use a pretrained VGG16 model and resnet50 model.

Fourthly, Create a CNN to Classify Dog Breeds (from Scratch) to detect the humans.

Finally, Use transfer learning to create a CNN to classify dog breed, we used . resnet50 and continue training it to get out but of 133 classes. so, the image is identified as dog/human, we can pass this image to an CNN which will process the image and predict the breed that matches the best out of 133 breeds.

Benchmark Model

- The CNN model created from scratch has 12% accuracy.
- But the CNN model created using transfer learning resnet50 has 83% accuracy.

Evaluation Metrics

I used the sample accuracy rule to measure the model accuracy which is:

$$\text{Accuracy} = \frac{true\ predicated\ images}{number\ of\ images\ in\ the\ data}$$

Project Design

- Step 0: Import Datasets
  - Import the necessary dataset and libraries, Pre-process the data and create train, test and validation dataset. Perform Image augmentation on training data.
- Step 1: Detect Humans
  - using OpenCV's implementation of Haar feature-based cascade classifiers
- Step 2: Detect Dogs
  - Created using VGG16 pretrained model.
- Step 3: Create a CNN to Classify Dog Breeds (from Scratch)
  - Creating train valid test datasets
  - The model has 3 convolutional layers. All convolutional layers has kernal size of 3 and stride 1. The first conv layer (conv1) have input of 3 and the final conv layer (conv3) output of 128.
  - I have also used ReLU as my activation function. using max pooling layer of (2,2) is used which will reduce the input size by 2. We have two fully connected layers that finally produces 133 output. with dropout of 0.3 is added to avoid overfitting.
- Step 4: Create a CNN to Classify Dog Breeds (using Transfer Learning)
  - Specify appropriate transforms, and batch_sizes
  - Using resnet model as pretrained model
- Step 5: Write Algorithm
  - Write an algorithm that accepts a file path to an image and first determines whether the image contains a human, dog, or neither
- Step 6: Test Algorithm
  - Test your algorithm at least six images