Dog Classifier Project Proposal

Domain Background

"Image classification is one of classical problems of concern in image processing. The goal of image classification is to predict the categories of the input image using its features. There are various approaches for solving this problem such as k nearest neighbor (KNN), Adaptive boost (Adaboosted), Artificial Neural Network (NN), Support Vector Machine (SVM)." (Thai, Hai, Thuy 2012)

For this project, I will use Convolutional Neural Network (CNN) to approach the dog classifier problem. The goal is to build a pipeline that would process real-world user supplied image and identify the dog breed in that image. In this project, I would use different models for different parts of the problem, including the algorithms to first identify if there's a dog or human existing in the image, and then the algorithm to identify the breed if applicable.

Since this project would mostly focus on CNN and I want more exposure to application of this field, I choose this project as my capstone project for my Udacity Nanodegree. I also think taking on this project would allow me to use similar models to process other more advanced problems.

Problem Statement

The problem for this project to develop an algorithm through which the users would be able to input an image. Then the algorithm would be able to process the image and identify if there's a dog or a human existing in this image. It would return the estimated dog breed if there's a dog in the image, and it would return the resembling dog breed of the person that existing in the image. If the algorithm determines neither exists in the input image, it return an error.

Datasets and Inputs

The datasets are provided by Udacity project. In the git repository, the project includes two zip files. One is the data for dog images, and the other is the data for human images.

For the dog data file, it includes a total of 8531 dog images, 20% of which would be used as the validation sets, and the other 10% would be used as the test set. We have a total of 133 dog breeds can functions as folders.

For the human data file, it includes a total of 13233 human images, 20% of which would be used as the validation sets, and the other 10% would be used as the test set. The human image has 5749 humans as the example, and it would be formatted as 250x250 image.

The data is not balanced, because there can be several images of the same person. The same applies to dog image. Therefore we need to take those into acounts.

In this project I would also import any python libraries such as cv2 , numpy, pandas or tqdm to facilitate the model building process.

Solution Statement

The solution contains an algorithm that would take an image, and return either “dog”, “human” or neither based on the model estimate. It would also output the dog breed or resembling dog breed. The solution includes using OpenCV on Haar Feature-based cascade classifier to detect the human face in the image. It then use another pre-trained VGG-16 model on ImageNet with trained weights to detect the dog in the image. Then it would include the third model, an CNN model which classifier dogs or human faces. The final algorithm put all three models together, takes an input image, and returns a predicted value.

Benchmark Model

The benchmark model would be a CNN model created from scratch with an test accuracy of 10%.

Evaluation Metrics

For CNN Models (without transfer learning), the test accuracy need to be more than 10%. For CNN models with transfer learning, the test accuracy should be more than 60%.

Besides, we would input several dog and human images into the model once we completed the algorithm, and we would expect an accuracy be at least around 60% (so not much less than the test accuracy).

Project Design

Step 1: To solve the problem stated in the problem statement section, I would first use algorithm like OpenCV to identify the human image. I would first process the image and transfer them into grayscale. Then I would implement the algorithm using Haar feature-based cascade classifier to detect human faces in the image.

Step 2. Use a pre-trained VGG-16 model to detect dog image. The model is being trained on ImageNet, with weighted already been set. We would use this model to detect dog image.

Step 3. Use CNN model (transfer learning) to classify the dog breed in the dog image. We would build this algorithm from scratch. Train the model, and select models with optimate hyperparameters and test the model on our test sets. The model would use the loss function to choose the best model to use. The goals is to predict the dog breed once this model receive any image of dog or human.

Step 4. After all models stated in step 1-3 are trained and tested, we would build an algorithm which would accept an image and return the dog if it determines the image contains a dog, human if it contains a human, and neither if it contains neither. It would also output the dog breed of the dog or the resembling dog breed of the human.

Reference

1. Thai, L., Hai, T., & Thuy, N, I.J. Information Technology and Computer Science, 2012, 5, 32-38, Published Online May 2012 in MECS (http://www.mecs-press.org/) DOI: 10.5815/ijitcs.2012.05.05