Develop a Dog Breed Classifier Using CNN

Domain Background

Dog breed identification is a challenging problem as the differences among breeds can be subtle. A decade ago, the common pipeline for this type of questions was to extract features from an image and fed them to a classifier. These features were often extracted at generic locations within the image, with the hope that these locations will reveal some patterns about the class. A breakthrough in building models for image classification came with the discovery that a convolutional neural network (CNN) could be used to extract high-level representations of the image content without the need of feature engineering. In this project, we will use CNN to build a pipeline that is able to identify a breed of dog if given a photo or image as input.

Problem Statement

The goal of this project is to build a pipeline that can be used within a web or mobile app to process real-world, user-supplied images. Given an image of a dog, the web app will identify an estimate of the dog's breed. If supplied an image of a human, the code will identify the resembling dog breed.

Datasets and Inputs

Both the human (LFW) dataset and dog dataset are used as inputs for classification.

- Human (LFW) dataset: 18,982 images from 5,749 persons are included for testing the classifier.
- Dog dataset: 8,753 images are used as training (n = 6,813), testing (n = 969), and validation set (n = 968). 133 different dog breeds with varying counts and size of images are ready for use.

Solution Statement

Before building the CNN model to make the predictions, we firstly try some pre-trained models provided by OpenCV and ImageNet. For example, we can use Haar Cascades from OpenCV for human face detection. We can also obtain pre-trained VGG-16 model based on images from ImageNet to detect dogs in images. After evaluating the performance of those pre-trained models, we can move forward to create a CNN from scratch or via transfer learning for assigning classes to a dog or human image.

Benchmark Model

- The CNN created from scratch for dog breed classification must attain a test accuracy of at least 10%. A random guess will provide a correct answer roughly 1 in 133 times, which corresponds to an accuracy of less than 1%.
- The CNN created via transfer learning for dog breed classification must attain a test accuracy of at least 60%

Evaluation Metrics
We will use the following metrics to evaluate the model performance. Since the dog dataset is slightly imbalanced, we prefer F1 score and AUC.

- Micro-averaged and macro-averaged precision: For macro-averaged precision, calculate precision for all classes individually and then average them. For micro-averaged precision, calculate class wise true positive and false positive and then use that to calculate overall precision.
- Micro-averaged and macro-averaged recall
- Micro-averaged and macro-averaged F1 score (F1)
- AUC

Project Design

1. Import Datasets: Download the required human and dog datasets
2. Detect Humans: write and assess a Human Face Detector using a pre-trained model from OpenCV
3. Detect Dogs: write and assess a dog Detector using a pre-trained VGG-16 Model
4. Create a CNN to classify Dog Breeds (from scratch)
5. Use a CNN to classify Dog Breeds (using Transfer Learning)
6. Write and test an algorithm that accepts a file path to an image and first determines whether the image contains a human, dog, or neither. Then, 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 is detected in the image, provide output that indicates an error.

References

1. Dog Breed Classification Using Part Localization: https://link.springer.com/chapter/10.1007/978-3-642-33718-5_13

2. Dog Breed Classifier for Facial Recognition using Convolutional Neural Networks: https://ieeexplore.ieee.org/abstract/document/9315871

3. Evaluation Metrics For Multi-class Classification: https://www.kaggle.com/nkitgupta/evaluation-metrics-for-multi-class-classification

4. CNN: https://developers.google.com/machine-learning/practica/image-classification/convolutional-neural-networks