Dog Breed Classifier using CNN

Domain Background

Image classification has always been the top priority of machine learning. Creating a good image classification algorithm was difficult because of the lack of sophisticated techniques. With the advent of CNN, the entire game changed. CNN are robust algorithms that help create deep models that understand multiple aspects of images such as edges, shapes, colors etc.

The project is based on creating a classifier to classify different breeds of dogs using deep learning techniques such as a Convolutional Neural Network (CNN). The project can be used to classify stray dogs and help with NGOs who work with Animal adoption and animal shelter to help people with the adoption and donation process.

Problem Statement

Aim of the project is to build a supervised learning ML model that can be deployed and used within a web app to process real world, user-supplied images.

The tasks of our final model are as below:-

- Human Face detector - If given an image, model outputs True if Human face is detected, else returns False
- Dog detector - If given an image, model outputs True if dog is detected else outputs False.
- Dog breed detector - If given an image, model predicts the dog breed associated with that that image. If it detects dog, it will predict dog breed, if human face is detected, it predicts the dog breed that resembles with the human face.

Datasets and Inputs

The dataset is provided to us by Udacity. It includes images of dogs and humans and their corresponding labels.

- 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 images are of different sizes and different backgrounds, some images are not full-sized. The data is not balanced because the number of images provided for each breed varies. Few have 4 images while some have 8 images.
- 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. The data is not balanced because we have 1 image for some people and many images for some.

Solution Statement

The solution is provided by a deep learning algorithm called Convolutional Neural Network (CNN) that understands various features of an image and can classify the images with capabilities of multiclass classification. The solution is divided into three steps:

1. Understand the difference between dogs and human faces
2. Create a benchmark model with a pretrained model such as VGG16
3. Create my own model with hyperparameter tuning

Benchmark Model

For our benchmark model, we will use the Convolutional Neural Networks (CNN) model created from scratch with an accuracy of more than 10%. This should be enough to confirm that our model is working because random guess would be 1 in 133 breeds which are less than 1% if we don't consider unbalanced data for our dog images.

Evaluation Metrics

With 133 different classes to classify one image into, there are high chances of class imbalance. The metric that we will use to address this will be F1 score as accuracy will provide poor results in such cases. F1 score is calculated with Precision and Recall from the confusion matrix as below:

	Actual -- True/False	
Predicted -- Positive/Negative	True Positive	False Positive (Type I)
	False Negative (Type II)	True Negative

Precision: TruePositives / (TruePositives + FalsePositives)

Recall: TruePositives / (TruePositives + FalseNegatives)

F1 Score: 2*Precision*Recall/(Precision + Recall)

Project Design

Step 1: Import the necessary dataset and libraries, Pre-process the data and create train, test and validation dataset. Perform Image augmentation on training data.

Step 2: Detect human faces using OpenCV's implementation of Haar feature based cascade classifiers.

Step 3: Create dog detector using pretrained VGG16 model.

Step 4: Create a CNN to classify dog breeds from scratch, train, validate and test the model.

Step 5: Create a CNN to Classify Dog Breeds using Transfer Learning with resnet101 architecture. Train and test the model.

Step 6: Write an algorithm to combine Dog detector and human detector.

References

1. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

2. https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2