Resnet50 Similarity Search

George Strauch
Ruben Suarez
Avery Wylin

INTRO TO COMPUTER VISION CS 4379C

11/29/2021

Table of Contents

Objective

The objective of this project is to search for images of related dog breeds based on the Euclidean Distance between the feature vectors collected from the average pooling layer from the pretrained network Resnet50.

By using a best-fit search on the distances between the extracted feature vectors, similar images can be displayed along with their labels.

Implementation

Resnet50 is a Convolutional Neural Network (CNN) that uses 50 layers. It is pretrained from the ImageNet database using about 14 million images of various objects and animals cropped to a 244-pixel square. From these images, Resnet50 is able to create a vector of features that we utilize for image similarity. For this project, we will be focusing on dogs even though Resnet50 is capable of handling other features.

Training the network is extremely computationally expensive, but once trained, the network can be utilized with significantly less cost. By using the pre-trained network available in MATLAB's Deep Learning library, more accurate results can be derived without the expense of training the network locally.

A set of images modeling specific dog breeds are run through the CNN and their feature vectors are then extracted from the average pooling layer. This will be used as the example set for classifying subsequent images.

New images being classified are then passed through the CNN and their features are compared against the example set using the Euclidean distance between the two vectors. The smallest distances are then matched and then used for classification.

Matlab Functions

main.m:	Loads the images to test from the test_dogs_dir, then for each image it creates a vector of the distance between the given test image and the set of pre-calculated images. Using the mink() function, the top results are then displayed in a grid view along with the original image.
dist_calc.m:	Calculates the Euclidean distance between two feature vectors and returns the vector of the distances. Can also calculate multiple test vectors against the pre-calculated dataset.
feature_extract.m:	Takes a single image and calculates its feature vector using Resnet50. The 'avg_pool' layer is extracted. The image is resized to a 244-pixel square and given to the neural network. The result is then returned.
create_dataset.m:	Not called by the user, computes the dataset feature vectors, computationally expensive.
accuracy.m:	finds the accuracy of determining dog breed when deep features are used as a lazy classifier.

Observations

Because the features extracted are generalized and not dog-specific, other factors can affect which images are returned as similar. Images that contain elements such as people, similar backgrounds, similar poses, and various other similar features such as shape and color can all be prioritized over the actual breed. However, the purpose of this program is more broad than simple classification.

Accuracy Metrics

(When Used as a Classifier)

Using 50 test samples and each sample predicting 20 result images from the dataset, the accuracy of breed detection is 56.53%. This accuracy is not that great because of the number of features that are present in the image and the fact that these features are extracted from whole, uncropped images. Ideally, if we were using this as a classifier, we would implement some form of a machine learning model that learns which features to weigh more heavily over others for classification. This is the Idea behind transfer learning where we could build our own classification layer and train it to categorize dogs with the features the network learned to classify other types of images with. Using a nearest neighbor approach does not yield good results because all features are weighted the same however considering there are 120 different categories of dogs available in the set we are using, 56% accuracy with nearest neighbor is very good.