Computer Vision Report

Dominic Azaris - Assignment 3

Introduction

In this assignment, a program was written to compare and match images based using kirsh filters and edge detection. The program was written in clojure.

Experiment

Design

The experiment being run is designed to test the effectiveness of the implementation, by taking three sets of 60 images (20 cars, 20 planes, 20 trains) and generating percentages on how well the algorithm matches them together.

The function that ran the experiment took each image in each set, and took the average differences between each set. For example, to compare the algorithms ability to detect cars in images, the function ran the difference between *car1 & car1, car1 & car2, ... , car1 & car20*. It will then run through and compare each car, with the entire cars set (*car2 & car1, car2 & car2 etc.).* Finally this will produce a single percentage comparing all of the cars with each other. The closer the percentage is to 100%, the better the algorithm.

Finally, the program will run the same comparisons on the car images versus the other image sets (trains and planes). In this case, the closer the end result is to 0, the better.

Execution

As this was a lot of combinations of comparisons, a simple cache system was implemented to massively reduce the runtime of the program. To do this, when the program generates the image descriptor, it will write it to a file unique to that image name. Then when it goes to regenerate the same descriptor, it will first check if a cache file exists, and if it does it will read the file instead of rerunning the filters and histogram generation.

Outcome

Expected

The expected outcome was for the car images to be close to 100% as possible to the other cars, and as close to 0% for the trains and planes sets.

Actual

The actual output was as follows:

Base Image Set	Compared To	Average Difference
Car	Car	**82.3%**
Car	Train	75.0%
Car	Plane	48.9%
Train	Car	**75.0%**
Train	Train	72.7%
Train	Plane	50.3%
Plane	Car	48.9%
Plane	Train	50.3%
Plane	Plane	**53.1%**

Summary

The results show that on average, the algorithm was fairly successful at identifying cars at an average difference of 82%. The algorithm was a lot less effective at identifying trains and planes. Trains were presented as closer matches to cars, at 75%, than other trains at 72%. It was also not good at identifying planes, as planes only got a 53% average difference with themselves.