RE Lee Research Report

Computer Vision is the study of a machine's ability to extract information from a set of natural images. Object classification is the task of determining whether or not a specific object, such as a plane or a cow, is visible in an image. We improved our object classification algorithm to compete in the PASCAL Visual Object Classes Challenge, which is an annual competition held to encourage development in this field.

Our particular approach to object recognition involves two broad phases, the training and testing phases. The training phase "trains" the computer to differentiate and recognize particular objects, producing a machine learning algorithm. This machine learning algorithm is then applied to the test images. The training phase consists of four steps: gathering SIFT features from the training images, clustering those features, pooling them, and generating the machine learning algorithm, which is called a Support Vector Machine (SVM). The test phase simply consists of applying the SVM to the test images. The steps in the training phase are very computationally intensive since our dataset is composed of thousands of images.

To shorten the length at which our pipeline identifies objects, we have parallelized the entire process at various levels. We used high-level parallelism to distribute work among multiple computers. We also used low-level parallelism to take advantage of the multiple processors available on each computer to further parallelize the computation.

We used Python to write scripts that sent parts of each step of the pipeline from a client computer to available host computers. The available computers would perform the necessary computation on its part of the step and return the result to the client computer. This process is performed on every step of the pipeline except the clustering; it is very difficult to divide the work for clustering among several computers, therefore to increase the computation speed we have sought to improve the efficiency of computation on a lower-level closer to the hardware.

On a lower-level, we parallelized our processes to take advantage of the multiple processors on each individual computer through POSIX threads (Pthreads) and Open Multi-processing (OpenMP). Since our algorithm is written in MATLAB, we needed to use MEX files (Matlab EXecutable files), which allow MATLAB to call on C code, to use Pthreads and OpenMP. Because C is relatively low level, the amount of computational overhead was also greatly reduced.

We also explored many ways to measure the similarity between images. Last year, the Earth Mover's Distance was implemented as the metric to measure similarity between representations of images. This year, we explored various other metrics, including the Earth Mover's distance, the Bhattacharyya distance, the Minkowski distance, the Jensen–Shannon divergence, the Kullback–Leibler divergence, the Chi-Squared Distance, Quadratic Form distance, the Quadratic-Chi distance, Euclidean Distance, and variations of these. We found the

Quadratic-Chi distance to be the most effective metric because it produced more accurate results when testing on sample images.

In addition to various levels of parallelism, we used Principal Component Analysis (PCA) to further increase the speed of the pipeline. PCA is a tool used in statistics that identifies the strongest trends in data, and thus reduces the number of dimensions that can approximate the data. The use of PCA reduced the number of dimensions that we use to accurately replicate the gathered SIFT features, allowing us to use the extra time to create more clusters, yielding more accurate results.