

1. Introduction

The purpose of the fourth milestone was to evaluate the performance of the classifier designed in milestone 3. Since the goal of the project is automated hand gesture recognition of live video, I will evaluate both the accuracy of the classifier and the speed of the entire program.

Image segmentation in the program is accomplished through a Gaussian mixture modeling (GMM) based approach, described in detail in milestone 1. Features extracted from the image (milestone 2) include the estimated fingertip and finger-web locations, which are transformed to polar coordinates. In milestone 3, I described the classifier design: a three-classifier majority voting algorithm that attempts to deliver accuracy (a “strong” classifier) through the use of multiple “weaker” classifiers. The three classifiers selected were a linear discriminant classifier, a linear SVM, and a weighted kNN classifier. The four hand gestures the program was designed to recognize are listed below in figure 1 (these will be referenced as gestures A, B, C, and D respectively).

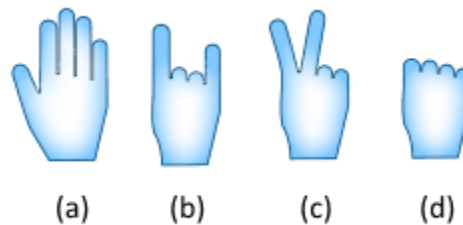


Figure 1: Hand Gestures

2. Speed Analysis

The execution of the `analyzeimages.m` program on each image in the ‘eval/subject1’ dataset (103 total) was timed in chunks corresponding to the major stages of the algorithm: segmentation, feature extraction and classification. For the purposes of this test, image capture and plot creation time were ignored, in order to evaluate the algorithm itself, rather than the visualization code. The results of the speed test are shown below in table 1.

Table 1: Speed Measurement

Process	Mean (ms)	St. Dev. (ms)	Min (ms)	Max (ms)
Segmentation	855	529	149	2510
Feature Extraction	118	15.7	90.5	173
Classification	29.6	6.50	22.7	50.5

Segmentation is the longest stage of the algorithm by far. This is unsurprising due to the iterative nature of the MATLAB `gmfit` function—an image might go through as many as 100 iterations

before a suitable model is found. Additionally, segmentation has the highest time variability of any stage, with times on the given sample ranging from 0.1 to 2.5 seconds. This is due again to the iterative procedure of the Gaussian fitting process—an image with “weak” Gaussian clusters will take more iterations to find an accurate model, while an image with strong clusters will take much less time. Adding up the average time for each step, the average time from image capture to the completion of image classification is approximately 1.002 seconds.

3. Accuracy Analysis

For the first test, a sample of 103 images of my hand was used (the ‘eval/subject1’ dataset). Using the `analyzeimages.m` program, the gesture recognition algorithm was run on each image and the accuracy of the overall classification was recorded. The results are shown below by gesture in table 2.

Table 2: Accuracy Measurement

Gesture	Total Images	Images Correct	Accuracy
A	21	17	81%
B	23	20	87%
C	35	18	51%
D	24	18	75%
Total	103	73	71%

Overall the algorithm achieves an accuracy rate of 71% on average. Clearly, the gesture C classification is the weakest of the four, but gesture B recognition is quite good. One way to improve the classifiers overall would be to train them on additional training data. For the training of these classifiers, I used only a small 60 image dataset of my hand. With additional training data, especially from a variety of different people, the classifier would be much more accurate. To improve the gesture C classification in particular (after gathering more data), it would most likely be worthwhile to add another classifier to the pool that is particularly good at correctly identifying gesture C (even if it is not effective at identifying the other gestures). Then, if this new classifier predicts that the gesture is gesture C, its vote would be given a greater weight.

In order to test the “big-picture” effectiveness of the classifier, I gathered a second data set from a second test subject and re-measured the accuracy of the classifier. The results are shown below in table 2. As shown, the classifier performs approximately equally well on a person foreign to the system, a desirable characteristic for a hand gesture recognition system.

Table 3: Accuracy Measurement

Gesture	Total Images	Images Correct	Accuracy
A	11	9	82%
B	10	6	60%
C	5	3	60%
D	7	6	86%
Total	33	24	73%

4. Conclusion

Overall, the algorithm achieved moderate accuracy and speed performance under most conditions. Additionally, it appears mostly insensitive to differences between individual hands, focusing more on the overall shape rather than unique features of each hand. The algorithm, while not perfect, has achieved its goal of classifying the hand gestures from a live video feed.