Department of Systems Design Engineering
University of Waterloo

Lab 3

Image Classification

Pattern Recognition
SYDE 372

Dan Hendry, 20207096.
Jenny Lu, 20172834

April 3, 2009

Introduction

The goal of this lab is to provide some context for classification in real-world situations. Image classification is chosen as an example of an important and interesting application of classification. Specifically, texture analysis is an important aspect of computer vision that depends on image classification. In this lab, image classification is performed on a set of texture images using both labeled classification (MICD) and unlabelled clustering (K-means). A standard set of texture images, the Brodatz images, are used.

This lab is done using Matlab.

Discussion

Feature Analysis

The simple feature vector for each block measures the average variability of each image in the horizontal and vertical directions. Therefore, images that exhibit similar average variability in both directions would be more easily confused. In this case, the texture of cork would be most likely confused with the texture of grass, because both are high contrast images and also share similar granularity of variability in both directions. Hence the average variability in both directions would be similarly high for any block size for the two textures.

The texture of wood and the texture of raiffa both exhibit a great degree and granularity of variability in the horizontal direction, but a low granularity of variability in the vertical direction. Specifically, the texture of wood is close to uniform in the vertical direction, while the texture of raiffa exhibits high contrast in the vertical direction, but with a large granularity. Therefore the two images would be more likely confused for smaller block sizes.

The texture of face would be least likely confused with other textures because it exhibits the lowest degree and granularity of variability compared to other textures. This is particularly true for smaller block sizes, since they are less likely to contain edges in the image.

Labelled Classification

The misclassification rate for $n = 2$ is found to be $P(\varepsilon) = 0.7064$, and classification performance is only slightly better than chance. The confusion matrix and breakdown of misclassification by image for this block size is shown in Table 1. The poor performance is due to the high degree of variability between features of different blocks, and the poor characterization of textures given the small block size used. Since patterns in the textures span greater than areas of 4 pixels, the chosen block size spans only a small fraction of a pattern repetition, and each block is arbitrarily placed at an area of high or low contrast, resulting in the high degree of variability. Furthermore, since the blocks do not encompass at least one full pattern recognition, the features do a poor job of characterizing the average variability of each texture.

As expected, the classifier performs relatively well for the textures of cotton and paper, which are significantly finer-grained compared to other textures. This means that the small block still covers a significant portion of a pattern repetition, since the patterns span a smaller area and repeat much more frequently. The classifier also performs somewhat well for the texture of wood, which is fine-grained with high contrast in the horizontal direction, but highly invariable in the vertical direction. The high degree of contrast between horizontal and vertical features, combined with the finer grain in the horizontal direction, results in better performance of the classifier. The classifier also performs relatively well in classifying the texture of stone, because the texture is much smoother and coarse-grained, so from the perspective of a small block, is in most samples highly invariable in both directions. This property makes the texture of stone relatively unique resulting in better classification performance. The classifier performs extremely well on the texture of face for similar reasons: Except in the case of edges, the majority of samples of small block sizes produce features that indicate a high degree of invariability in both directions, due to the smoothness of the texture. Hence the classification performance for the texture of face is actually enhanced using small block sizes due to its uniqueness from other textures from this perspective. (Note that the texture of stone is easily confused with the texture of face due to this similarity.)

Table 1: Confusion matrix using 2x2 blocks

	Classified										
Truth	1	2	3	4	5	6	7	8	9	10	$P(\varepsilon)$
1	4	0	0	2	3	0	0	4	0	3	0.750
2	0	7	2	2	2	0	3	0	0	0	0.563
3	2	1	1	2	2	1	3	1	2	1	0.938
4	5	3	0	2	4	0	1	1	0	0	0.875
5	6	0	0	1	6	0	0	2	1	0	0.625
6	2	2	0	5	1	0	4	1	0	1	1.000
7	0	6	0	1	1	0	7	1	0	0	0.563
8	4	0	0	1	1	0	0	6	0	4	0.625
9	3	1	1	2	6	0	2	1	0	0	1.000
10	0	0	0	0	1	0	0	1	0	14	0.125

The misclassification rate for $n = 8$ is found to be $P(\varepsilon) = 0.4190$, a significant improvement over the misclassification rate for $n = 2$. The confusion matrix and breakdown of misclassification by image for this block size is shown in Table 2. The improvement in classification performance is expected, because the increased block size encompasses a greater portion of a full pattern repetition or even multiple pattern repetitions in the case of finer-grained textures, and is hence better able to characterize the average variability of each texture. Better characterization also results in lower variability of features between different blocks of the same image.

The classifier generally performs similarly well for most textures, apart from a few exceptions. As previously mentioned, the texture of cork is easily confused with the texture of grass and the texture of paper, because the three share similar average variability due to the high contract of their edges, and similar pattern size. The texture of cork is most often misclassified because there is a higher degree of variability between the features of its sample blocks; the granularity of the texture is slightly variable, causing it to be easily misclassified as grass in blocks where the pattern size is smaller, and misclassified as paper in blocks where the pattern size is bigger. Note that the texture of grass is also easily confused with paper. This is again due to the high degree of variability between features of its sample blocks; some parts of the texture image are not as fine-grained or as high contrast as others. Finally, notice that the classification performance for the texture of face actually decreases slightly using larger blocks. This is because there is a greater chance that edges would land in large block samples, hence not all block samples reflect how uniquely smooth the texture is in general.

Table 2: Confusion matrix using 8x8 blocks

Classified

Truth	1	2	3	4	5	6	7	8	9	10	$\mathrm{P}(\varepsilon)$
1	12	0	0	1	0	0	0	3	0	0	0.250
2	0	12	0	0	0	1	3	0	0	0	0.250
3	0	1	4	2	0	2	5	0	2	0	0.750
4	1	0	0	12	0	1	1	0	1	0	0.250
5	0	0	0	1	6	0	0	1	8	0	0.625
6	0	0	4	3	0	1	5	0	3	0	0.938
7	0	0	0	2	0	3	11	0	0	0	0.313
8	0	0	0	0	2	0	0	11	0	3	0.313
9	0	0	1	1	2	0	0	1	11	0	0.313
10	0	0	0	0	2	0	0	0	1	13	0.188

Table 3: Confusion matrix using 32x32 blocks

The misclassification rate for $n = 32$ is found to be $\mathrm{P}(\varepsilon) = 0.1128$, again a significant improvement over the misclassification rate for $n = 8$. The confusion matrix and breakdown of misclassification by image for this block size is shown in Table 3. The improvement in classification performance is again expected, because the increased block size encompasses a greater portion of a full pattern repetition or even multiple pattern repetitions in the case of finer-grained textures, and is hence better able to characterize the average variability of each texture. Better characterization also results in lower variability of features between different blocks of the same image.

The classifier generally performs similarly well for most textures, apart from relatively poor performance in classifying the texture of cork. This is because granularity of the texture varies throughout the image, and using larger block sizes cannot fully correct for the higher variability between features of its sample blocks. Finally, the classification performance for the texture of face again improves for the bigger block size, since it is likely for most blocks to contain edges, hence the variability between features of sample blocks is reduced; furthermore, the high contrast occurring at edges is balanced out by the overall smoothness of each sample block.

Classified

Truth	1	2	3	4	5	6	7	8	9	10	$\mathrm{P}(\varepsilon)$
1	15	0	0	0	0	0	0	1	0	0	0.063
2	0	16	0	0	0	0	0	0	0	0	0.000
3	0	1	15	0	0	0	0	0	0	0	0.063
4	0	0	0	16	0	0	0	0	0	0	0.000
5	0	0	0	0	15	0	0	0	1	0	0.063
6	0	0	6	1	0	7	2	0	0	0	0.563
7	0	0	0	0	0	1	15	0	0	0	0.063
8	0	0	0	0	0	0	0	14	0	2	0.125
9	0	0	0	0	1	0	0	0	15	0	0.063
10	0	0	0	0	0	0	0	2	0	14	0.125

Image Classification and Segmentation

The results of classifying the multi-texture image shown in Figure 1 using 8x8 sample blocks is shown in Figure 2.

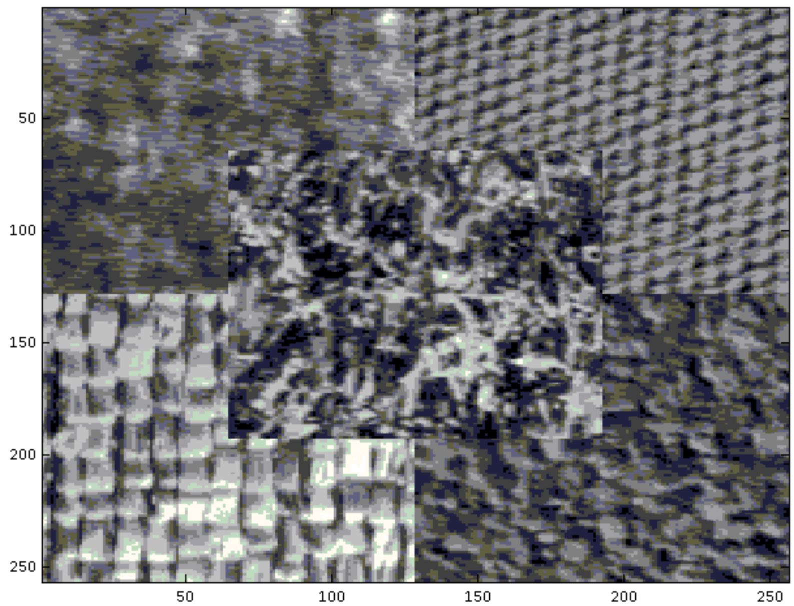

Figure 1: Image multim

The classification successfully identifies the general arrangement of the multiple textures in the image, and shows correct boundaries between textures compared to the original image. As expected, classification performance near the edges and borders are relatively poor, because blocks located near the edges of the image do not contain enough information for classification, and blocks located near the borders contain two different textures and hence do not accurately represent either texture. The texture of cloth, in the upper-left corner, and the texture of raiffa, in the lower-left corner, are generally classified well, and this is reflective of the fact that the MICD classifier for 8x8 blocks classifies these two textures relatively well, as shown in Table 2. The texture of grass in the middle and the texture of cork in the lower-right corner are both poorly classified, as can be expected from the relatively poor performance of the MICD classifier for 8x8 blocks for these two textures. The texture of cotton in the upper-right corner is not classified as well as can be expected, given the good classification performance of the MICD classifier.

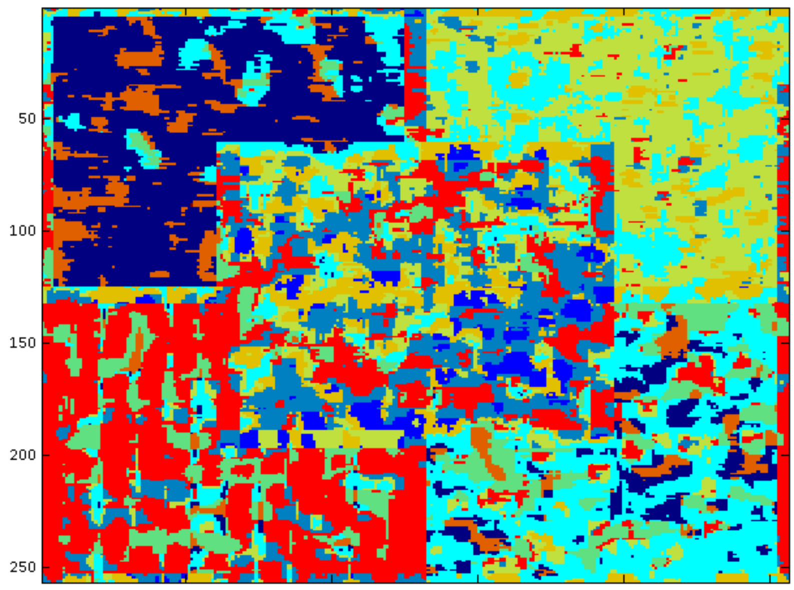

Figure 2: Classified cimage

Unlabelled Clustering

K-Means Clustering

Assuming $K = 10$, the K-means algorithm is used to cluster the 32x32 block features, using random data points as initial prototypes. The results are shown in Figure 3.

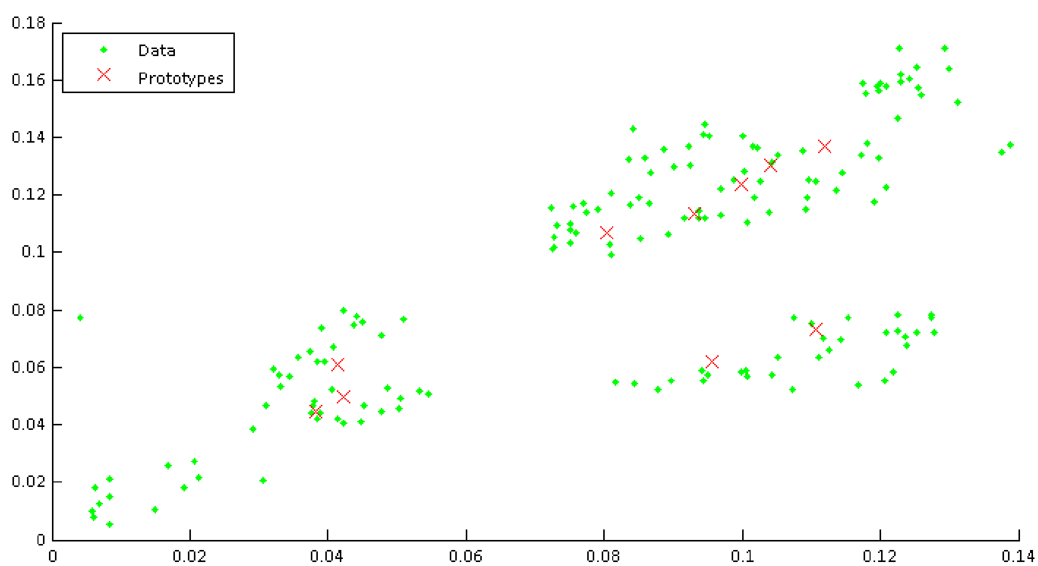

Figure 3: Results of one trial of K-means clustering

The prototypes generated from four trials of K-means clustering are shown in Figure 4. The prototypes generated by K-means clustering on each trial are strongly influenced by the randomly selected set of initial prototypes, and hence the results are highly variable.

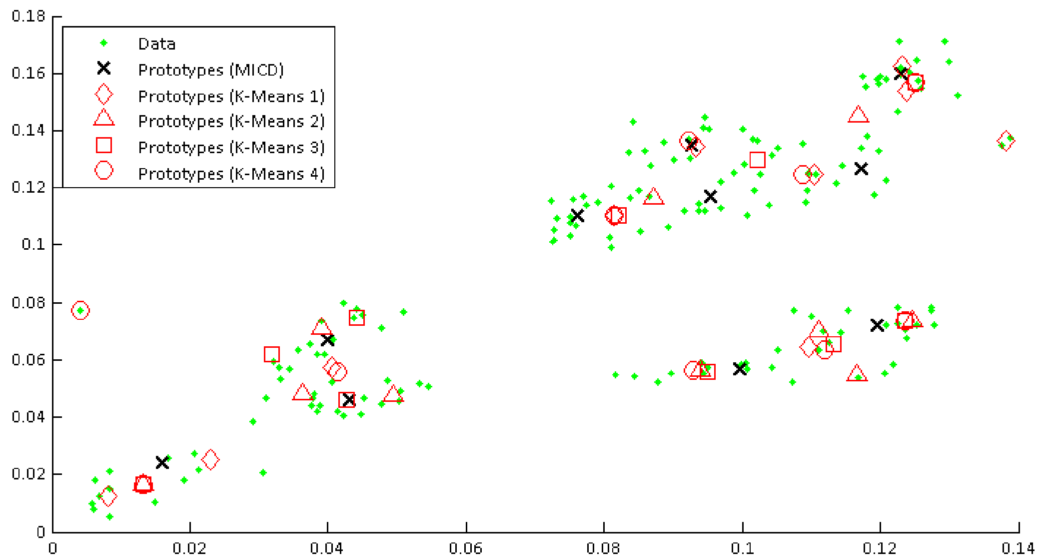

Figure 4: Results of four trials of K-means clustering, superimposed over prototypes used in MICD classifier

It should be noted there appears to be five roughly separable clusters which are be easily distinguishable by eye. K-means clustering is observed to consistently place at least one cluster prototype in each of these clusters. The variability in prototype locations is due to each of these relatively separated clusters being divided further as the algorithm must account for ten, not five, clusters. Since no clear division can be easily made, the final result is highly dependent on initialization. It is reasonable to assume that results produced by the k-means clustering algorithm would be far less variable if the number of clusters were set to more closely reflect data shape, perhaps to $K = 5$ or $K = 6$. The features represented by the plotted data set are extracted from $K = 10$ classes, hence using a lower value of K may render the results meaningless; however, this result nevertheless demonstrates that several of these classes are confusable and furthermore not compact, resulting in poor performance when clustering using the K-means algorithm.

It is also observed that when a cluster prototype happens to be randomly initialized to an outlier, significantly separated from all other data points, it would never converge towards other clusters. Trial 4 demonstrates an example of this case.

Although variable in location, cluster prototypes produced by the k-means are often found to approximate the prototypes determined by the MICD classifier remarkably well. Without knowing true class means, it is not unreasonable to suggest a person could do little better than K-means clustering, given this data set.

Fuzzy K-Means Clustering

Again assuming $K = 10$, the fuzzy K-means algorithm is used to cluster the 32x32 block features, using $b = 2$ and random data points as initial prototypes. The results are shown in Figure 5.

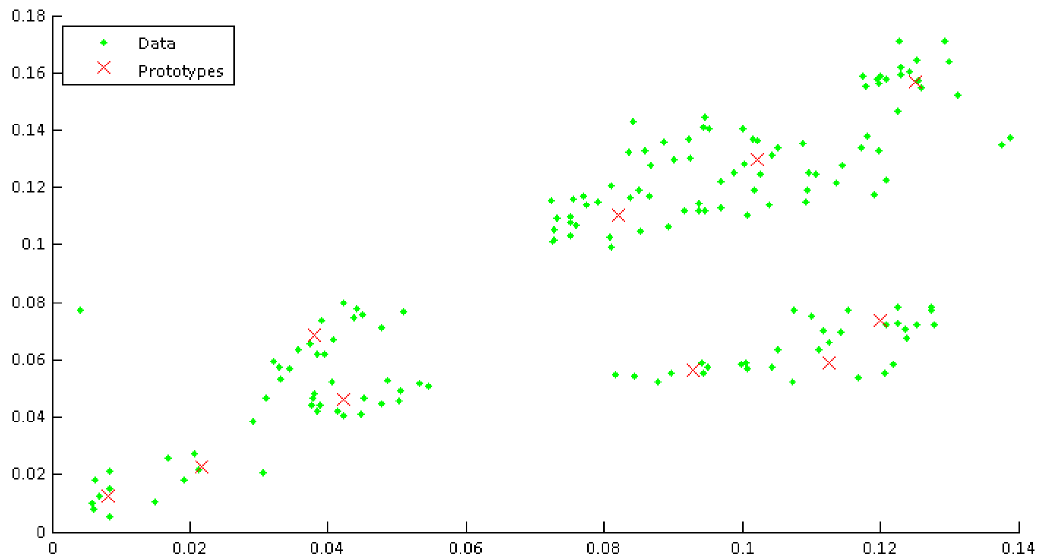

Figure 5: Results of one trial of fuzzy K-means clustering

Prototypes generated by fuzzy K-means clustering are observed to be far less variable, compared to K-means clustering. Only occasionally would the initialization points significantly influence converged prototypes. Nearly all application of the algorithm converged to a far more limited and consistent set of cluster prototypes in comparison to regular K-means. The results of four trials of fuzzy K-means clustering, each with a different set of initial prototypes, are shown in Figure 6. These four trials demonstrate the common phenomenon where converged prototypes from the four trials completely overlap. Compared to K-means clustering, the converged prototypes are generally shifted towards the center of the range of the data set. This result is expected, because the prototype of each cluster is influenced to some degree by the cumulative contribution of low probabilities of data points from other clusters distributed across the data set. On final important observation is that fuzzy K-means is far less likely to become stuck on outliers.

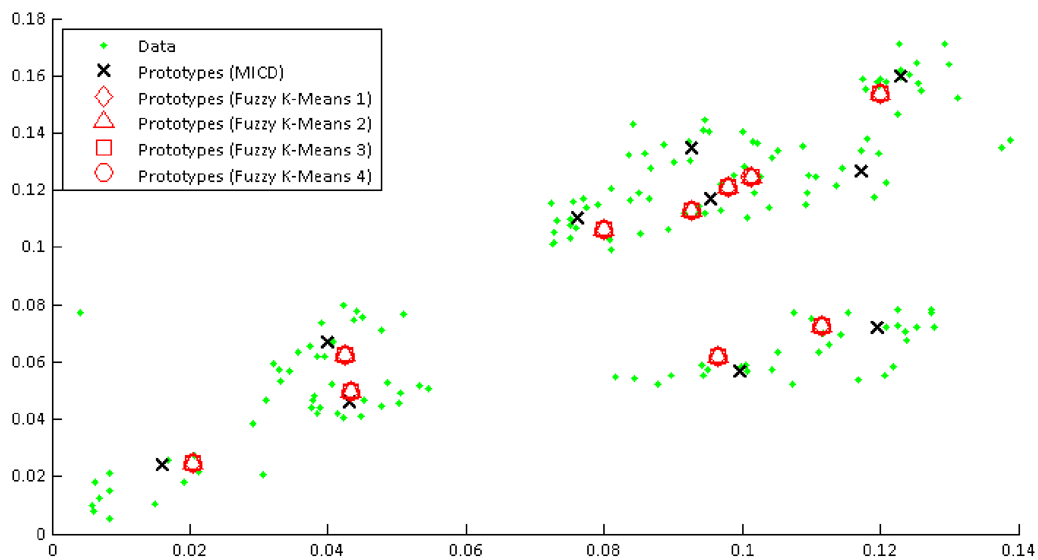

Figure 6: Results of four trials of fuzzy K-means clustering, superimposed over prototypes used in MICD classifier

Despite its consistency, fuzzy K-means fails to accurately estimate prototypes for many classes. This is likely due to the noted phenomenon that converged prototypes from fuzzy K-means clustering tends to shift towards the center of the range of the data set. However, it is difficult to draw any concrete conclusions about the impact on classification performance. For example, consider the two classes located at the bottom-right of the data set. Although the cluster prototypes are not accurate and shifted from the MICD prototypes towards the center, the degree and direction of the shift are similar, and an MED classifier between the pairs of prototypes would likely be shifted in the same direction, but still produce reasonable results. This is not always the case, however, as shown by the relatively large cluster encompassing four overlapping classes. Here, the prototypes generated by fuzzy K-means differs significantly from those generated by the MICD classifier, and this is likely due to the fact that these classes are neither compact nor separable.

Image Classification

Since K-means is a clustering method, it could not be used to classify and label textures in the image; however, it could used to identify regions of texture similarity. Once these regions have been identified, additional information would be required to label these regions as specific textures. For example, this could involve human labeling of key portions of the image. Labeling must occur for clusters to be properly interpreted.

Given that some form of labeling is performed on the clusters, K-means could then be used to cluster and then classify and label the image. The error rate of the algorithm would be highly dependent the degree of separation of the classes in the feature space. Accurately identifying class prototypes through clustering has already been seen to be inaccurate when classes have significant overlap. In such cases, significantly worse performance could be expected from K-means when compared to MICD. Furthermore, there appears to be a tradeoff between better approximations in some cases (achieved by K-means) and consistency between trials (achieved by fuzzy K-means).

K-means does have one significant advantage over the MICD classifier, potentially resulting in lower error rates. Training data for the MICD classifier comprises all ten possible textures, yet there are only five different textures present in the image. Assuming this is known, we may apply K-means clusters assuming $K = 5$. This may possibly result in significantly separated classes being identified far more accurately. The added accuracy is derived from the fact that MICD might attempt to classify blocks as textures not present in the image, while limiting to $K = 5$ using K-means eliminates that possibility. However, if the number of textures present in the image is not known before clustering, setting $K > 5$ would in fact be prone to misclassifying large portions of the image, since K-means would be forced to cluster more classes than are actually present.

Conclusions

The simple features extracted to characterize textures based on average variability in the vertical and horizontal directions generally worked well and demonstrate a reasonable degree of success when used to perform classification and clustering. Given the contrast, granularity, and difference in variability between the two directions, some textures are more likely to be confused than others, and experimental results validate expectations. Block size is strongly correlated with classifier performance, which is expected, as an increased block size encompasses a greater portion of a full pattern repetition or even multiple pattern repetitions in the case of finer-grained textures, and is hence better able to characterize the average variability of each texture. Better characterization also results in lower variability of features between different blocks of the same image. Applying the developed MICD classifier to a multi-texture image demonstrates a reasonable degree of success for those textures that are not easily confused with others, but performance is generally poor at edges and borders due to the fact that the blocks no longer accurately reflect a single texture in those areas.

The prototypes generated by K-means clustering on each trial are strongly influenced by the randomly selected set of initial prototypes, and hence the results are highly variable. It is also observed that when a cluster prototype happens to be randomly initialized to an outlier, significantly separated from all other data points, it would never converge towards other clusters. Although variable in location, cluster prototypes produced by the k-means are often found to approximate the prototypes determined by the MICD classifier remarkably well.

In comparison, prototypes generated by fuzzy K-means clustering are observed to be far less variable, compared to K-means clustering. Nearly all application of the algorithm converged to a far more limited and consistent set of cluster prototypes in comparison to regular K-means. Compared to K-means clustering, the converged prototypes are generally shifted towards the center of the range of the data set. This result is expected, because the prototype of each cluster is influenced to some degree by the cumulative contribution of low probabilities of data points from other clusters distributed across the data set. Due to this result, despite its consistency, fuzzy K-means fails to accurately estimate prototypes for many classes.

Given that some form of labeling is performed on the clusters, K-means could then be used to cluster and then classify and label the image.