SYDE-572: Introduction to Pattern Recognition

Lab 3: Image Classification
Report
Group 17: Marta Zheplinska - 21050969
mzheplin@uwaterloo.ca
Professor - Alexander Wong

1 Introduction

In the Lab 3, labeled MICD and unlabeled K-means methods of image classification based on two features are considered. Features determine the variability of images in vertical and horizontal directions. In the Lab 3, MICD classifiers were constructed based on three matrices with labeled data, and these classifiers were tested with other corresponding matrices, a confusion matrix was built. In the second part, each pixel of the given image was classified and a comparative characteristic of input and output images was made. The third part is devoted to familiarization with the unlabeled classification method based on the K-mean method.

2 Labelled Classification

Question
1. By looking at the texture images themselves, (e.g., on the SD372 home page), which do you think would be most likely to be confused with the other images? Why? 2. Which images are likely to be most distinct? Why? Be sure to answer the questions in the context of the features extracted from the texture images.

Answer
1. In my opinion, in the context of the features considered in the task, it will be easiest to misclassify pictures of a cork and a grass, since a certain pattern of shade change is the worst visible in their example, and therefore a randomly selected block can be classified as a block of another picture.
2. I have an assumption that the texture of the face will be best classified, since a large level of vertical shade change is visible, which distinguishes it from other pictures.

Question
For each of the three feature matrices f2, f8 and f32, develop the MICD classifier. You are not required to plot the classification boundaries. Apply the classifier to the test data f2t, f8t and f32t. 2. What was the misclassification rate for each image and for each n? Prepare three confusion matrices, each like the one shown in Table 1, one table for each value of n, to compare how the images are classified. 3. Compare and explain the results for the different values of n

2.1 feature matrix f2

This section demonstrates the results of building the MICD classifier, where the plane point can be classified to 1 of 10 textures. Files **MICD.m , MICD_discriminant.m , Classify_point_MICD.m** contain program's realization of the MICD classifier. The construction of decision boundaries was not a mandatory task but was done for the sake of interest (Figure 1).

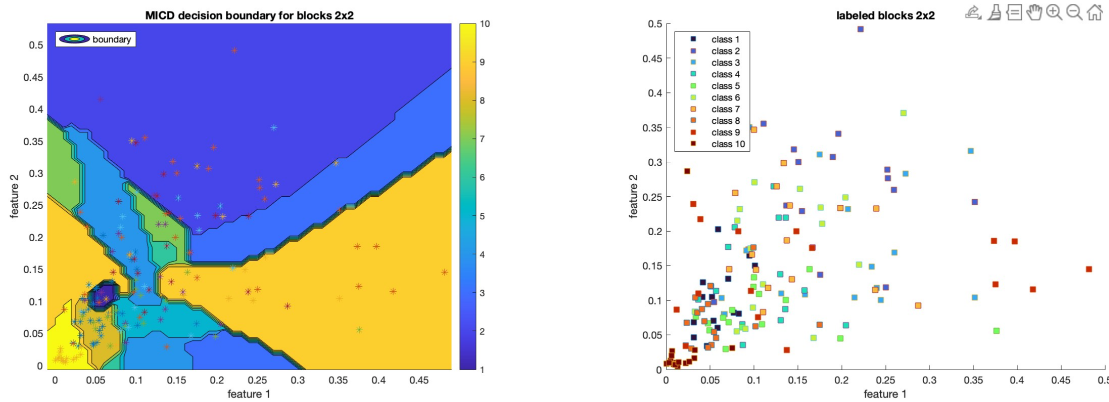

Figure 1: MICD decision boundary for 2x2 blocks(left) and labeled clusters from matrix f2(right)

Code for calculating a confusion matrix for some trainig data is stored in file **MICD_conf_matrix.m**. The confusion matrix below shows the classification results for 10 classes considering 2x2 blocks. Each row represents the true class, and each column represents the predicted class. The diagonal elements represent correct predictions, while off-diagonal elements represent misclassifications.

true \predicted	1	2	3	4	5	6	7	8	9	10	misclassification rate
1	1	0	0	2	3	0	1	4	5	0	0.9375
2	0	7	4	0	2	1	2	0	0	0	0.5625
3	0	3	0	3	0	0	1	1	7	1	1
4	0	3	0	1	4	0	1	1	6	0	0.9375
5	1	0	0	2	4	0	0	5	4	0	0.75
6	0	2	3	2	0	2	1	2	3	1	0.875
7	0	5	4	0	2	0	1	0	4	0	0.9375
8	0	0	0	2	0	0	0	9	3	2	0.4375
9	2	1	2	4	2	0	2	1	2	0	0.875
10	0	0	0	0	1	0	0	1	0	14	0.125

Overall accuracy: 0.2562.
Overall rate of misclassification : 0.7438.

When considering 2 by 2 blocks, the MICD classifier tested on the data of the **f2t** matrix showed a very high level of incorrect classification of clusters. Only for face texture the misclassification rate is relatively low, while class 3 - grass clusters were 100% misclassified. Most of them were classified as class 9 clusters. In fact, in the final result, a large number of clusters (32) of different classes were determined to belong to the 9th class, although only 2 clusters of this class were correctly classified. In general, it is obvious that 2x2 blocks carry a very low information value, which is then converted into numerical data, so this classifier turned out to be very weak as a result.

2.2 feature matrix f8

This section demonstrates the results of building the MICD classifier based on clusters that correspond to blocks 8x8, where a cluster belongs to 1 of 10 textures. Figure 2 demonstrates a decision boundary for built MICD classifier which was tested on testing data from matrix **f8t**:

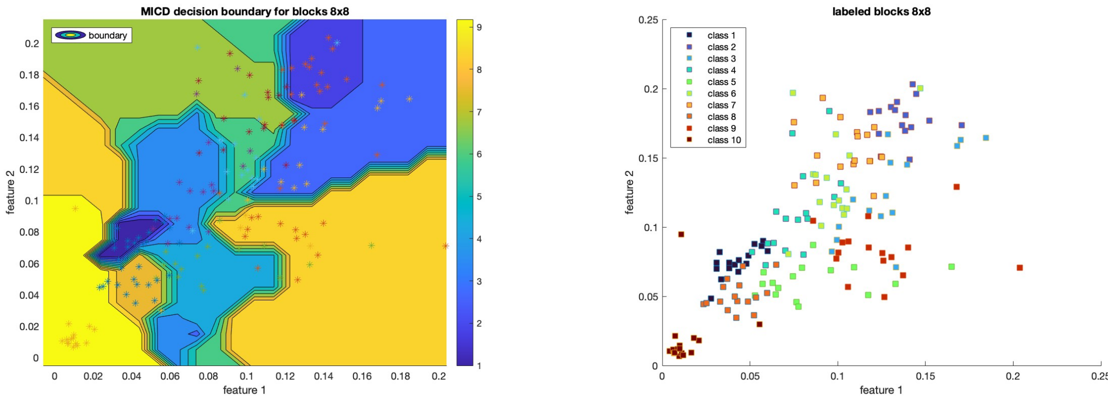

Figure 2: MICD decision boundary for 8x8 blocks(left) and labeled clusters from matrix f8(right)

Below is the confusion matrix calculated for this case. The right column contains information on the level of misclassification of each class.

true \predicted	1	2	3	4	5	6	7	8	9	10	misclassification rate
1	9	0	0	3	0	0	0	4	0	0	0.4375
2	0	10	2	0	0	1	3	0	0	0	0.375
3	0	1	4	2	0	4	3	0	2	0	0.75
4	1	0	0	12	0	1	1	0	1	0	0.25
5	0	0	0	1	5	0	0	1	9	0	0.6875
6	0	0	4	3	0	2	4	0	3	0	0.875
7	0	0	0	2	0	8	6	0	0	0	0.625
8	0	0	0	0	2	0	0	10	0	4	0.375
9	0	0	1	1	3	0	0	0	11	0	0.3125
10	0	0	0	0	1	0	0	0	2	13	0.1875

Overall accuracy: 0.5125.
Overall rate of misclassification : 0.4875.

In the case of consideration of 8x8 blocks, blocks belonging to the cork texture (class 6) were classified incorrectly the most times. At the same time 8 clusters of class 7 were classified as cork. The overall accuracy of the classifier exceeded 50%, but this is still a low indicator, and unreliable classification results are expected. This tells us that the 8x8 blocks still do not contain enough information for the decision to be made correctly. Figure 2 also shows that some class objects are mixed together, which leads to confusion.

2.3 feature matrix f32

The last but not least was MICD classification boundary learned on blocks of size 32x32 stored in **f32** matrix. Figure 3 (rigth) demonstrates how all clusters are accurately grouped by their classes. There is decision boundary on the left of Figure 3.

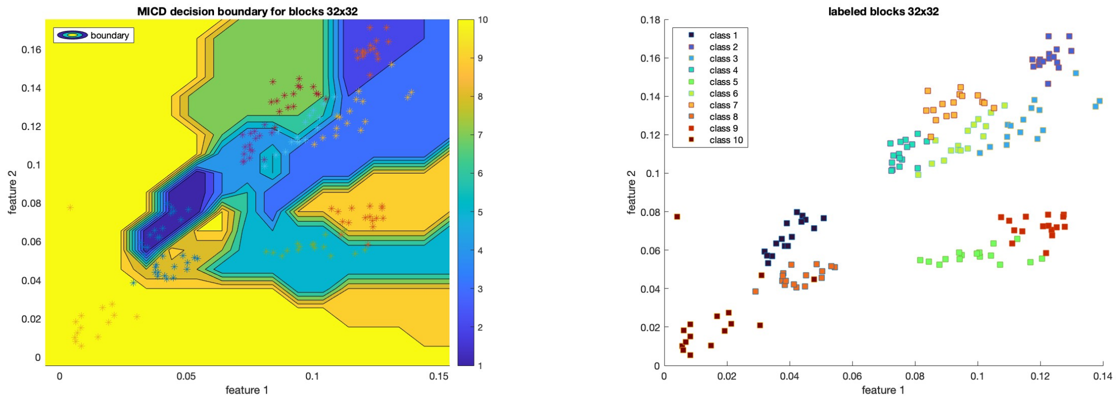

Figure 3: MICD decision boundary for 32x32 blocks(left) and labeled clusters from matrix f32(right)

Subsequently, this classifier was tested on the data of the **f32t** matrix. Below is a calculated confusion matrix, where the rows are the blocks of true belonging to the class, and the columns are their predicted classification using a classifier. The level of error is also calculated for each picture.

true \predicted	1	2	3	4	5	6	7	8	9	10	misclassification rate
1	12	0	0	0	0	0	0	0	0	4	0.25
2	0	16	0	0	0	0	0	0	0	0	0.0
3	0	1	15	0	0	0	0	0	0	0	0.0625
4	0	0	0	16	0	0	0	0	0	0	0.0
5	0	0	0	0	15	0	0	0	1	0	0.0625
6	0	0	6	1	0	7	2	0	0	0	0.5625
7	0	0	0	0	0	1	15	0	0	0	0.0625
8	0	0	0	0	0	0	0	11	0	5	0.3125
9	0	0	0	0	1	0	0	0	15	0	0.0625
10	0	0	0	0	0	0	0	1	0	15	0.0625

Overall accuracy: 0.8562.
Overall rate of misclassification : 0.1437.

Classification of 32x32 blocks showed much more accurate results, so clusters of classes 2 and 4 were correctly classified with 100% accuracy. However, class 6 blocks representing cortical particles were again classified as low. This can be explained by the fact that the texture of corks is difficult to predict and even blocks of size 32x32 pixels do not provide enough information for the recognition to be correct. In general, the accuracy of the classifier this time is high, and the error rate is much lower than in previous versions (for f8 and f2 matrices).

2.4 Comparison

As a result of building 3 classifiers and calculating confusion matrices, we can come to the conclusion that it is much better to classify textures when more pixels are taken into account, which provides more accurate information about the nature of changes in texture shades along the x and y axes. Each texture has its own characteristics of changing these parameters, but taking an insufficient amount of data, it is difficult to make successful predictions. Thus, the MICD classifier for matrix f2 turned out to be very weak, while the same classifier for matrix f32 showed a very high level of accuracy. The classifier trained on the f8 matrix showed average results, not good enough for its further use.

3 Image Classification and Segmentation

Question

Use your MICD classifier corresponding to n = 8 to classify each pixel in multf8. Create an array cimage such that cimage(i,j) contains the classified class number. Plot the result using imagesc(cimage)

Answer

In this task, each pixel of the image **multim**, which is a composition of several textures, was classified using the MICD classifier trained on the data of the f8 matrix. The classification results and comparison with the original image can be seen in Figure 4:

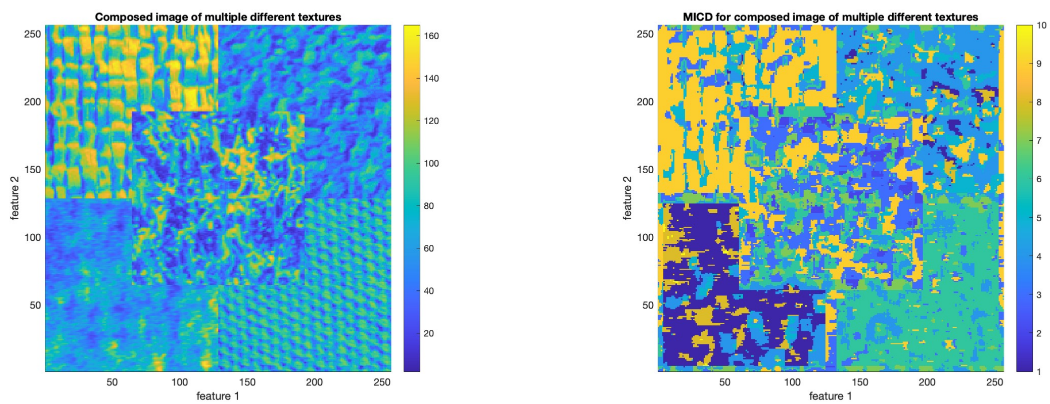

Figure 4: Image **multim**(left) and the result of its classification(right)

Question

State your observations. How do the classified regions in the resulting classified image cimage compare to original texture image multim?

Answer

The resulting picture partially reproduces the patterns seen in the original picture. The change in shades of the initial image corresponds to the change in the class to which the 8x8 block was classified. We can clearly distinguish the boundaries where one picture ends and another begins, which is a positive indicator. In general, the ideal would be if we would see 5 squares of different colors, which would mean that each block of pixels is classified 100% correctly, but this is practically an unattainable goal, since many blocks belonging to different images are very similar to each other and have the calculated values of their features are very close. The classification of the picture in the center was the most varied, as evidenced by the different colors of the pixels in the center. For the other four textures, certain colors predominated, others occurred less often.

4 Unlabelled Clustering

In the third part of the laboratory work, unlabeled clustering was implemented using the K-means and fuzzy K-means approaches. The data of the f32 matrix were considered as unlabeled data. 10 random points were chosen from them, and the distances of all other data to these points were calculated. The closest prototype determines whether a point belongs to the corresponding class. After that, the mean values of each class were calculated and considered as new prototypes. If the prototypes differed from their predecessors, all points were reclassified. The process continued until the prototypes stopped changing. In the algorithm of fuzzy K-means, the difference is that the point is classified to b "closest" classes, instead of one. In the lab 3, b = 2 was considered. Algorithm realization is stored in the file **K_means_fuzzy.m** During the execution of the task, many trials were conducted for K-means and fuzzy K-means algorithms. Figure 6 shows two results of matrix **f32** points clustering for each approach. Hexagrams depict prototypes for each class, points x are depicted in the color of the prototype to which the distance is the smallest.

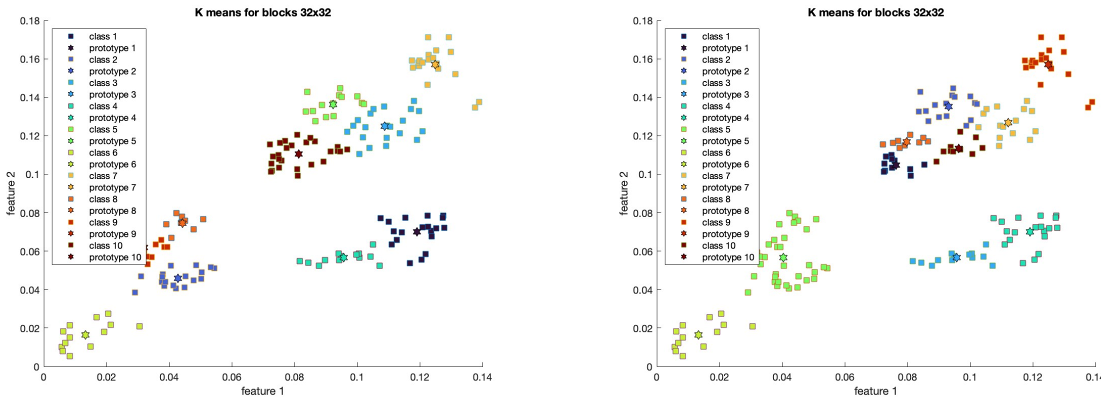

Figure 5: Unlabelled clustering with K-means algorithm, 2 trials

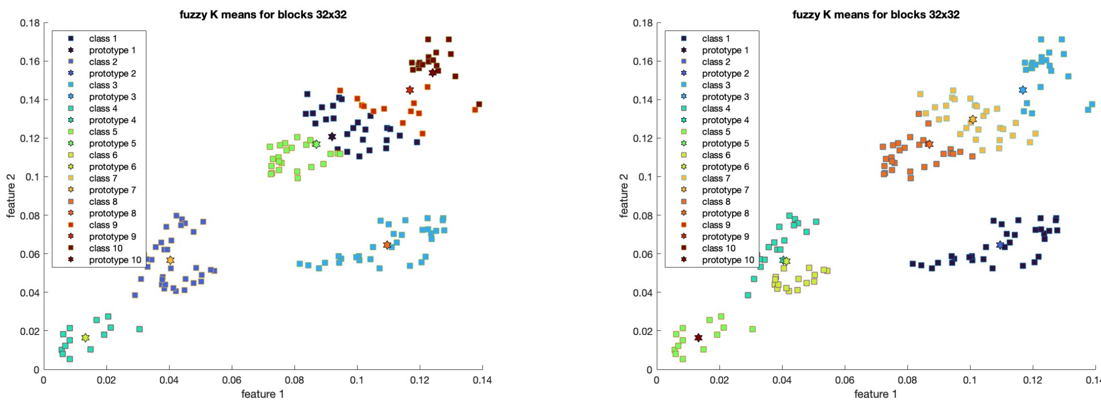

Figure 6: Unlabelled clustering with fuzzy K-means algorithm (b=2), 2 trials

Question

Compare the variability in the clustering from your trials. State your conclusions and observations. Furthermore, qualitatively compare your results from the unlabelled clustering with the results you obtained using the MICD classifier. How do they compare?

Answer

As a result of conducting many trials, quite different results were obtained. The K-means algorithm turned out to be very sensitive to the choice of initial prototypes, the variety of their choices gave various classifications. In the case of the fuzzy algorithm, it often happened that a certain set of points were identically classified into two classes and thus they merged into one class, so we got fewer classes. Compared to the MICD algorithm, the K-means algorithm is very weak, since we do not take into account the actual distribution of objects, we only group them by the closest prototype. Results vary widely when choosing different initial prototypes.

Question

Take a look again at the data points in matrix multf8. These are unlabelled data – without looking at multim, we don't actually know which pixel location belongs to which class. How well do you think K-means would work on the data in multf8? Could we use unlabelled clustering to classify the image multim

Answer

I think that K-means will not always show good results because it will only group points by distance from each other. The results can turn out to be very messy, but it all depends very much on the chosen prototypes, still it can happen by chance that they will be very successful.

5 Conclusions

In summary, in the laboratory work, the analysis of given textures and image classification was performed, MID classifiers were built. We can conclude that for accurate texture discrimination, it is worth taking into account a larger number of pixels at the same time, because they carry much more information about the features of the texture according to the features we have chosen. K-means clustering based on unlabeled data is very sensitive to the choice of initial choice and as a result we can get unreliable prototypes.