Kieran Broekhoven - 20502763
Nicholas Manna - 20490473
Naveen Nirmalaraj - 20509461

SYDE 372 - Pattern Recognition

Lab 3

1.0 Introduction

This lab applies concepts learned in this course to solve the problem of material classification on an image. Images of different textures are provided. Using both labelled classification (MICD) and unlabelled clustering (K-means) the materials presented in test images will be analysed.

2.0 Feature Analysis

Instead of analyzing the block as (n x n)-dimension vectors. The input vectors are reduced to 2d vectors having features X_1 and X_2 which correspond the variability of the image in the horizontal and vertical directions. The reduced feature vector is given by:

$$\underline{x}_{ij} = \begin{bmatrix} \sum_{\alpha=1}^{n} \sum_{\beta=1}^{n-1} \left(d_{ij}(\alpha,\beta) - d_{ij}(\alpha,\beta+1)\right)^2 / ((n)(n-1)) \\ \sum_{\alpha=1}^{n-1} \sum_{\beta=1}^{n} \left(d_{ij}(\alpha,\beta) - d_{ij}(\alpha+1,\beta)\right)^2 / ((n)(n-1)) \end{bmatrix}$$

1. By visual inspection of the images, grass, pigskin, cork and paper, have similar variability in the vertical and horizontal directions. From the feature plot that can be seen in Figure 1, it may be noted that these classes are grouped close together and as such it is difficult to produce decision boundaries that would distinctly divide them without overfitting.

2. Visually speaking, face, wood and cotton are very distinctive and would be the easiest to classify. Face has the lowest variability in either direction, while cotton has the the highest variability due to its frequently repeating pattern in both vertical and horizontal directions. This is clearly shown in Figure 1.

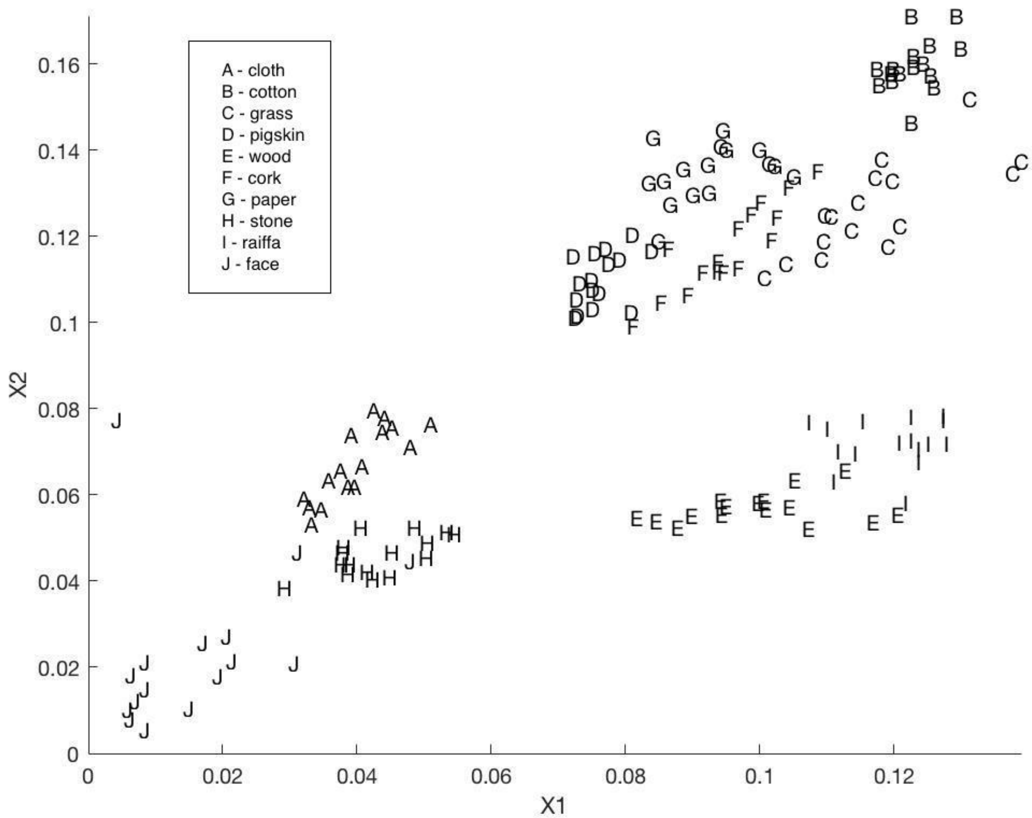

Figure 1: Plot of Features using F32 training data

3.0 Labelled Classification

The results produced by the MICD classifiers are provided below:

n	Error Rate (%)
2	74.38
8	48.75
32	14.37

Table 1: Error Rate of MICD classifier for different sample window sizes

	Classified									
Truth	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	1	0	0	2	3	0	1	4	5	0

2	0	7	4	0	2	1	2	0	0	0
3	0	3	0	3	0	0	1	1	7	1
4	0	3	0	1	4	0	1	1	6	0
5	1	0	0	2	4	0	0	5	4	0
6	0	2	3	2	0	2	1	2	3	1
7	0	5	4	0	2	0	1	0	4	0
8	0	0	0	2	0	0	0	9	3	2
9	2	1	2	4	2	0	2	1	2	0
10	0	0	0	0	1	0	0	1	0	14

Table 2: F2 Confusion Matrix

	Classified									
Truth	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	9	0	0	3	0	0	0	4	0	0
2	0	10	2	0	0	1	3	0	0	0
3	0	1	4	2	0	4	3	0	2	0
4	1	0	0	12	0	1	1	0	1	0
5	0	0	0	1	5	0	0	1	9	0
6	0	0	4	3	0	2	4	0	3	0
7	0	0	0	2	0	8	6	0	0	0
8	0	0	0	0	2	0	0	10	0	4
9	0	0	1	1	3	0	0	0	11	0
10	0	0	0	0	1	0	0	0	2	13

Table 3: F8 Confusion Matrix

	Classified									
Truth	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	12	0	0	0	0	0	0	0	0	4
2	0	16	0	0	0	0	0	0	0	0
3	0	1	15	0	0	0	0	0	0	0

4	0	0	0	16	0	0	0	0	0	0
5	0	0	0	0	15	0	0	0	1	0
6	0	0	6	1	0	7	2	0	0	0
7	0	0	0	0	0	1	15	0	0	0
8	0	0	0	0	0	0	0	11	0	5
9	0	0	0	0	1	0	0	0	15	0
10	0	0	0	0	0	0	0	1	0	15

Table 4: F32 Confusion Matrix

As can be seen from table n=32 produces the most accurate classifier. This is further supported by the confusion matrices in tables 2-4. This is due to the fact, that having two small of a sample window does not capture the essence of a texture. For the n=2 case, only four pixels are used, which is not enough information to produce a meaningful feature.

4.0 Image Classification and Segmentation

Figure 2 shows the plotted classification of the given image **multim.**

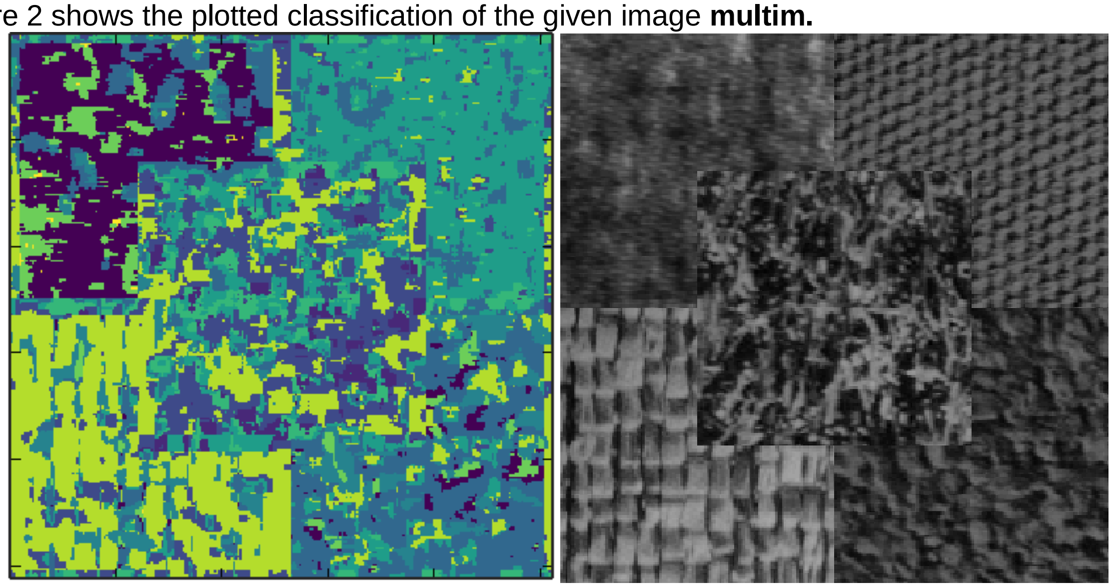

Figure 2: Classification of **multim** (left) compared to the original image (right)

From figure 2, the 5 textures of the original image can easily be visually distinguished. The outer four regions each have a strongly prevailing colour, representing the index of their respective classes. The center region has no prevailing colour, but can be identified because the other regions are so distinct.

There is a significant amount of incorrect colour present in each region. This is present because the classifier has a 48.75% error rate, causing very noticeable failures. The classification could be improved significantly by increasing n.

5.0 Unlabelled Clustering

In this part of the lab, unlabelled clustering is conducted via the k-means method and the alternative fuzzy k-means method. In both cases, it is assumed that the number of classes in the data set is 10. The k-means method is quick with only 6 iterations. Unfortunately, it is also inaccurate at times, as it places one prototype in a group of samples belonging to 2 or more classes, or it places too many prototypes in a group of samples, as can be seen in figure 3 below.

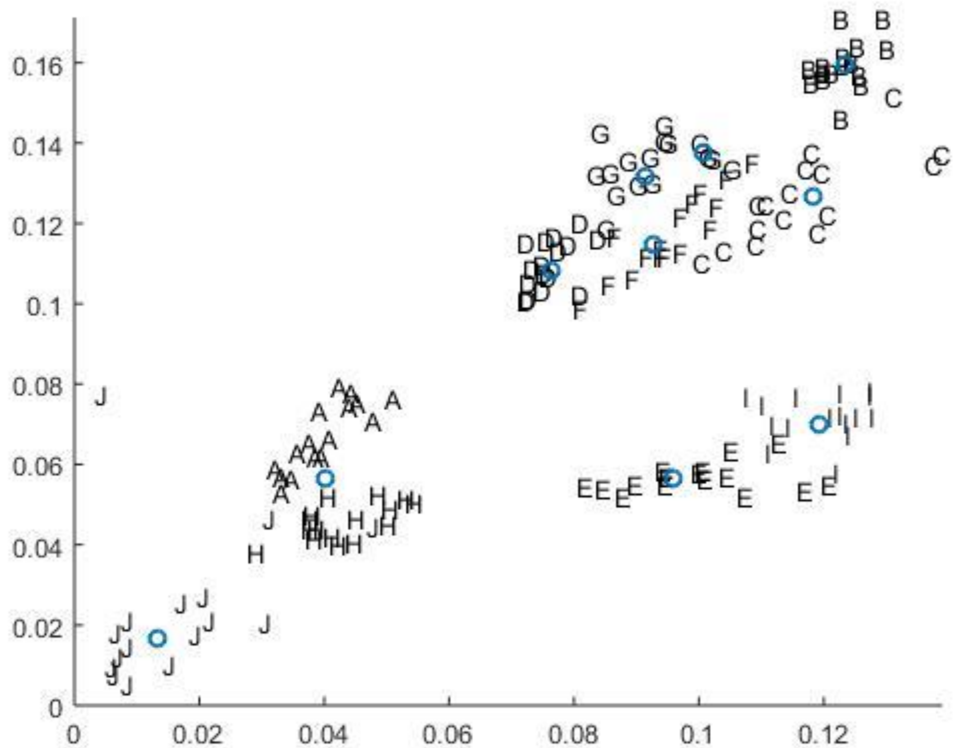

Figure 3: Unlabelled clustering via k-means.

Alternatively, the fuzzy k-means method takes 21 iterations to converge. Though this it takes longer, this method is a bit more accurate. Prototypes are placed in reasonable locations, with each of the 10 prototypes in the center of regions with samples of only one class, as can be seen in figure 4 below.

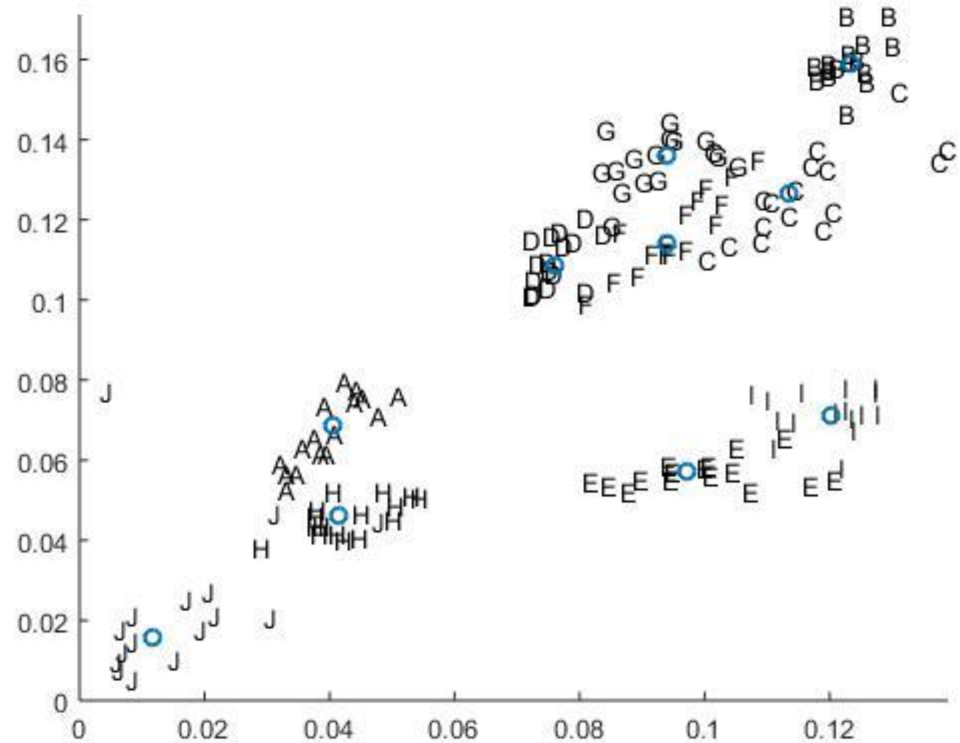

Figure 4: Unlabelled clustering via fuzzy k-means.

When running both these methods multiple times, it can be seen that there is a degree of variability with each run. What is common is that for more “obvious” clusters, like with class J or class B as can be seen in the two figures above, both it is easy for k-means and fuzzy k-means to place prototypes. However, prototyping for more convoluted clusters, like for D, G, F, and C in the two figures above, is very difficult for both methods. Sometimes the appropriate number of prototypes are placed at appropriate locations, other times there are too many prototypes or too few prototypes. As for comparing the two methods with one another, it can be seen that k-means has a tendency to overfit the data, especially if the starting point for the algorithm is an outlier, as can be seen in figure 5 below.

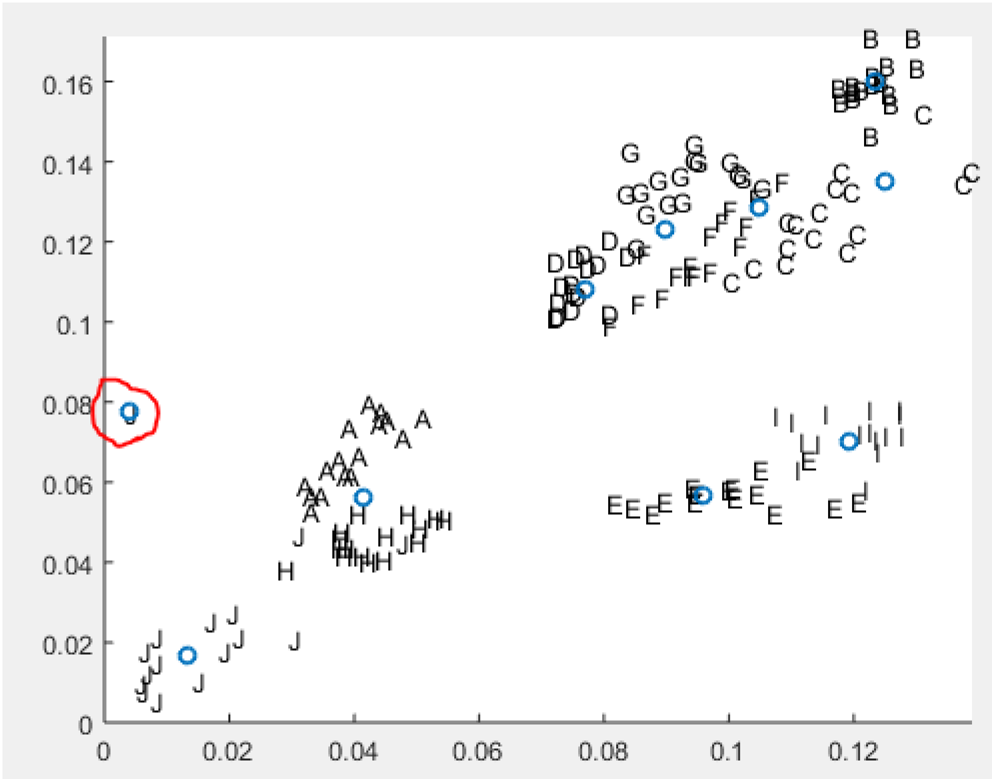

Figure 5: k-means overfitting data, resulting in prototyping for an outlier (red circle).

When comparing k-means and fuzzy k-means to the MICD classifications of previous sections, it can be seen that k-means and fuzzy k-means has the potential to be highly accurate. If appropriate starting positions are chosen (such as by avoiding outliers, if they could be detected beforehand), these two methods can be much more accurate than the MICD classification method.

However, the same cannot be said for classifying an image like multim from section 4. Using unlabelled clustering can be a hassle as overfitting scenarios may occur. Additionally, all that k-means and fuzzy k-means can tell us is the location of prototypes – we would not be able to match them to their appropriate pictures/classes afterwards. Lastly, the geometric nature of the image might also be confusing to these methods, which tend to work better when the data may be organized into “blob”-like regions. An image like multim is spliced together such that four of the original images are under a fifth image that is in the center – this results in odd shapes for the data.

6.0 Conclusion

One aspect explored in this lab is varying the size of image blocks. There is no perfect number for block sizes, as this value is application specific however, as observed in this lab, too small of a number leads to a large number of misclassifications. Additionally, MICD classification, k-means, and fuzzy k-means are all useful methods of classification when applied in appropriate situations.