University of Girona

Spain

Scene Segmentation and Interpretation

Lab 2 - Image Characterization using Texture

Submitted by :
Mr. Shubham WAGH
Mr. Nayee Muddin Khan
DOUSAI

Contents

List of Figures

1 Introduction

Texture is an important image feature used in many computer vision systems. It is considered as a patron of pixels that repeats in the image; in consequence, in theory it is possible to obtain information about the texture that identify clearly the patron. There are different kinds of textures, for instance a rough surface will show large differences in intensity between the high and low peaks. On the other hand a silky surface will show little differences in intensity between high and low peaks. In this lab we investigate the effects of using image texture in segmentation and classification.

2 Problem Statement

This lab coursework deals mainly with two tasks. First exercise deals with extraction of texture features using most common statistical methods i.e using co-occurrence matrices. It is mainly oriented to work with these statistical methods using **graycomatrix** and **graycoprops** in Matlab. For segmentation a local approach has been taken wherein texture features are computed for each pixel taking into account its neighborhood. These texture descriptors are used to enhance the Region Growing segmentation algorithm.
Second exercise deals with the classification of images in different classes. Here a global approach has been taken to produce a vector of features for an image. Then these feature vectors are used to classify the images into correct categories.

3 Texture Algorithm Analysis

The Gray-Level Co-occurrence Matrix (GLCM) is a tabulation of how often different combinations of pixel brightness values (grey levels) occur in an image. It is a second order measure which considers the relationship between groups of two (usually neighbouring) pixels in the original image. It outputs a square matrix (when normalized), represents the probability that two points with gray-levels i and j exist in a distance d and orientation θ. Every element (i, j) of GLCM is the sum of the number of times that the pixel with value i occurred in the specified spatial relationship to a pixel with value j in the input image.

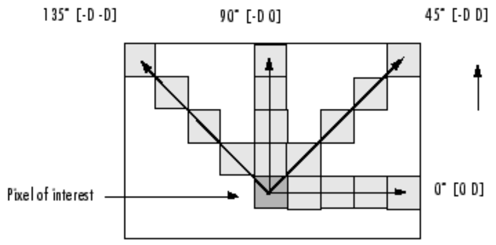


Figure 1: Spatial relationships of pixels

Various statistics can be generated based on the values of GLCM. The Matlab function *graycomatrix* provides an implementation of GLCM. From the GLCM matrix different texture feature properties can be computed (using *graycoprops* - Energy, Contrast, Correlation and Homogeneity). Some properties like (these properties are used for this lab coursework):

1. **Contrast** - determines local image variations. The statistical equation for contrast is given as:

$$Contrast = \sum_{i,j=0}^{N-1} P_{ij}(i-j)^2$$

2. **Energy** - determines uniformity in the image. The statistical equation for energy is given as:

$$Energy = \sum_{i,j=0}^{N-1} (P_{ij})^2$$

3. **Homogeneity** - determines concentration of high values along the GLCM diagonal. The statistical equation for homogeneity is given as:

$$Homogeneity = \sum_{i,j=0}^{N-1} \frac{P_{ij}}{1+(i-j)^2}$$

4. **Correlation** - determines correlation between pixel and neighbor. The statistical equation for correlation is given as:

$$Correlation = \sum_{i,j=0}^{N-1} P_{ij}$$

5. **Entropy** - determines uniformity in the image (another uniformity measure). The statistical equation for Entropy is given as:

$$Entropy = \sum_{i,j=0}^{N-1} -ln(P_{ij})P_{ij}$$

where:
P_{ij} = Elements i,j of the normalized symmetrical GLCM
N = Number of gray levels in the image

3.1 Segmentation Problem

We are given a set of images, each having different textures in them. For enhancing our region growing algorithm, we require texture features along with RGB. To extract these texture features from images, we need to compute Gray-Level Co-occurrence matrices (GLCM). Computation of GLCM requires gray-scale image as an input, so we converted the input image to grayscale image. We also normalized the images between 0 and 1

Then we compute the texture features of each pixel by considering a user defined patch size taking into account its neighborhood. Since Matlab has inbuilt functions for computing GLCM matrices, we did not specifically program this part. However, Matlab inbuilt function only accommodates four statistical properties for GLCM (energy, correlation, homogeneity and contrast). We wrote the code for entropy texture feature based on the math studied in the class.

There are two main functions related to the GLCM in Matlab. The function **graycomatrix** computes the co-occurrence matrix for the entire input image. Since for local descriptors, we require a co-occurrence matrix for each pixel, we used small images based on the patch size chosen for GLCM local descriptor computation. These small images are given as input to **graycomatrix** function along with other parameters as defined in Matlab. These parameters include offset and number of gray-scale levels. The variation in offset changes the distance and the angle between the current pixel and its neighborhood. The number of gray-scale level specifies the size of GLCM matrix. Also before GLCM computation, we normalize the image between 0 and 1.

Second function **graycoprops** is used to get statistic measure out of the resultant GLCM matrix. We used Matlab's inbuilt statistical characterisitics i.e. Energy, Homogeneity, Correlation, Contrast. We also computed Entropy from the normalized GLCM matrix. Each matrix element is multiplied by its logarithmic value and the result of all the pixels was added to compute the feature vector of the pixel.

After computing these GLCM descriptors of every pixel, texture characteristics were incorporated into the region growing algorithm implemented in lab1, to improve the performance of the algorithm. Before we use these texture descriptors we normalize the input RGB image between 0 and 1. Also we altered the dimension of the image to accept more than three parameters(apart from RGB, we include texture features as well).

4 Texture results

1. *Feli Texture Image with gray level = 8*

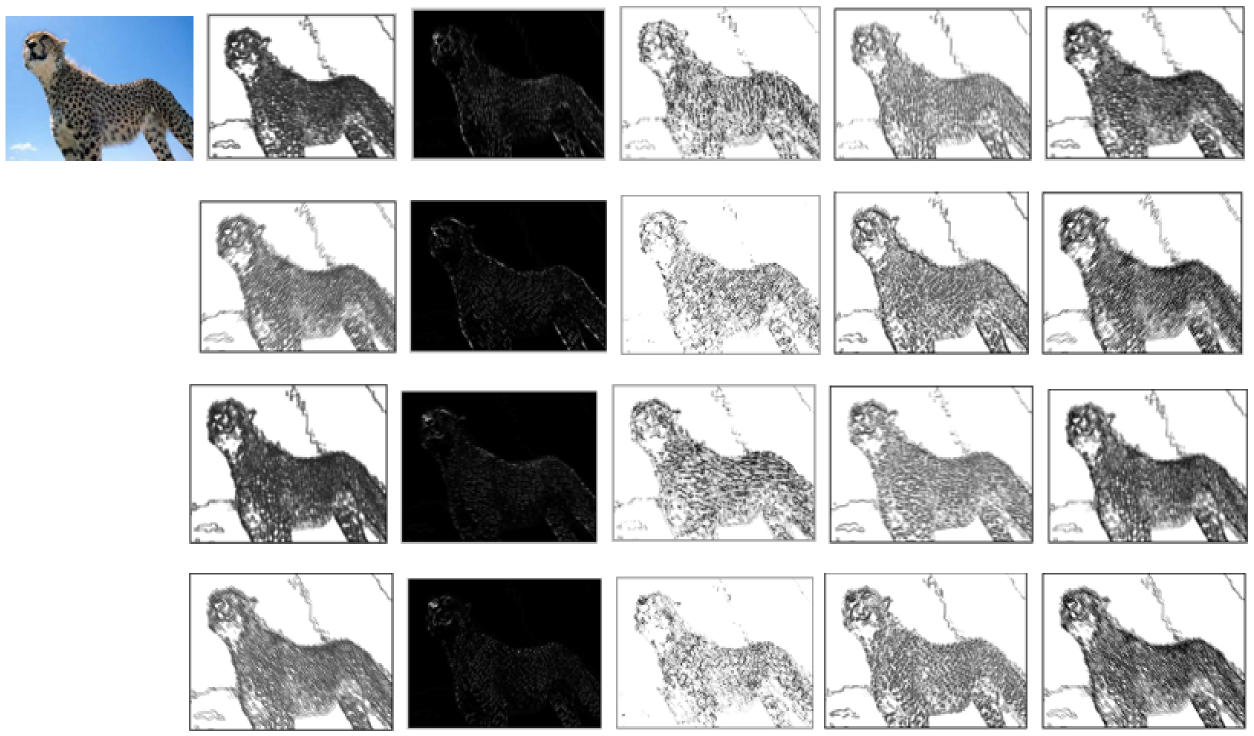


Figure 2: Feli Image texture features with patchsize 5 × 5 and offset distance 3
From left to right: Energy, Contrast, Correlation, Homogeneity and Entropy
Orientations from top to bottom: 0° , 45° , 90° and 135°

2. *Mosiac Texture Image with gray level = 8*

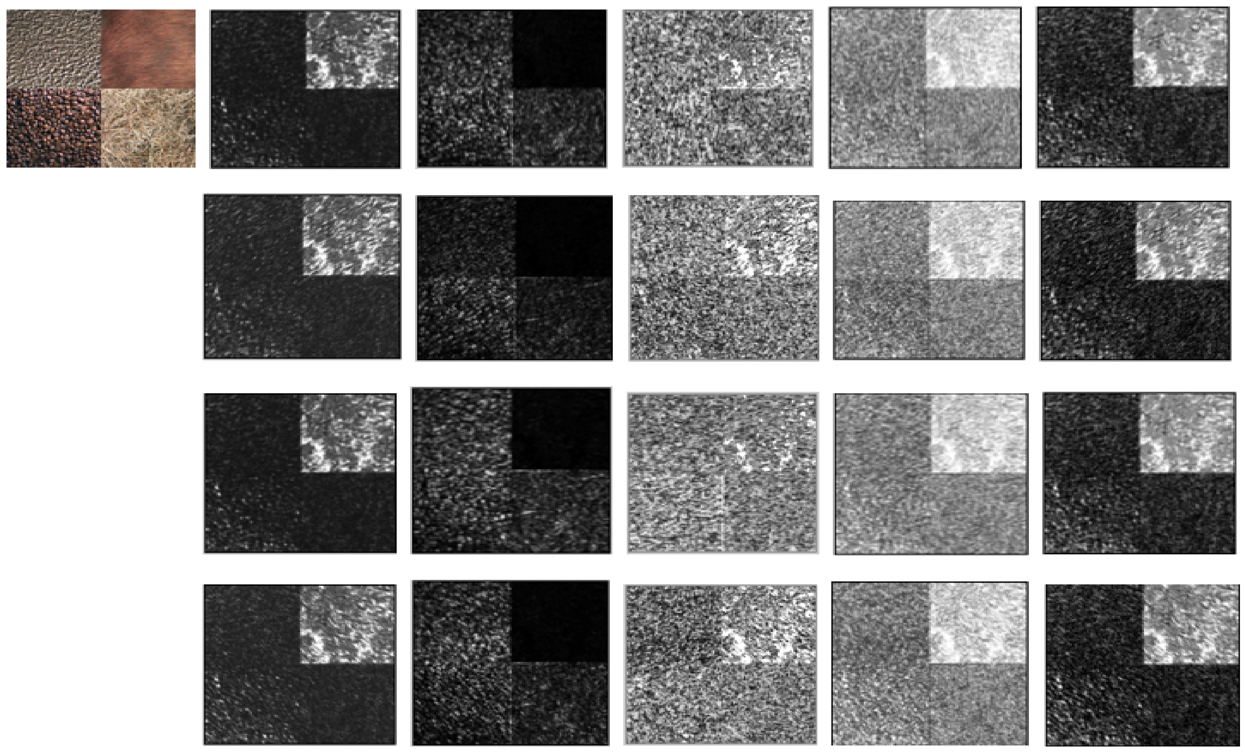


Figure 3: Mosiac Image texture features with patchsize 5×5 and offset distance 2
From left to right: Energy, Contrast, Correlation, Homogeneity and Entropy
Orientations from top to bottom: $0°$, $45°$, $90°$ and $135°$

3. *Hand Texture Image with gray level = 8*

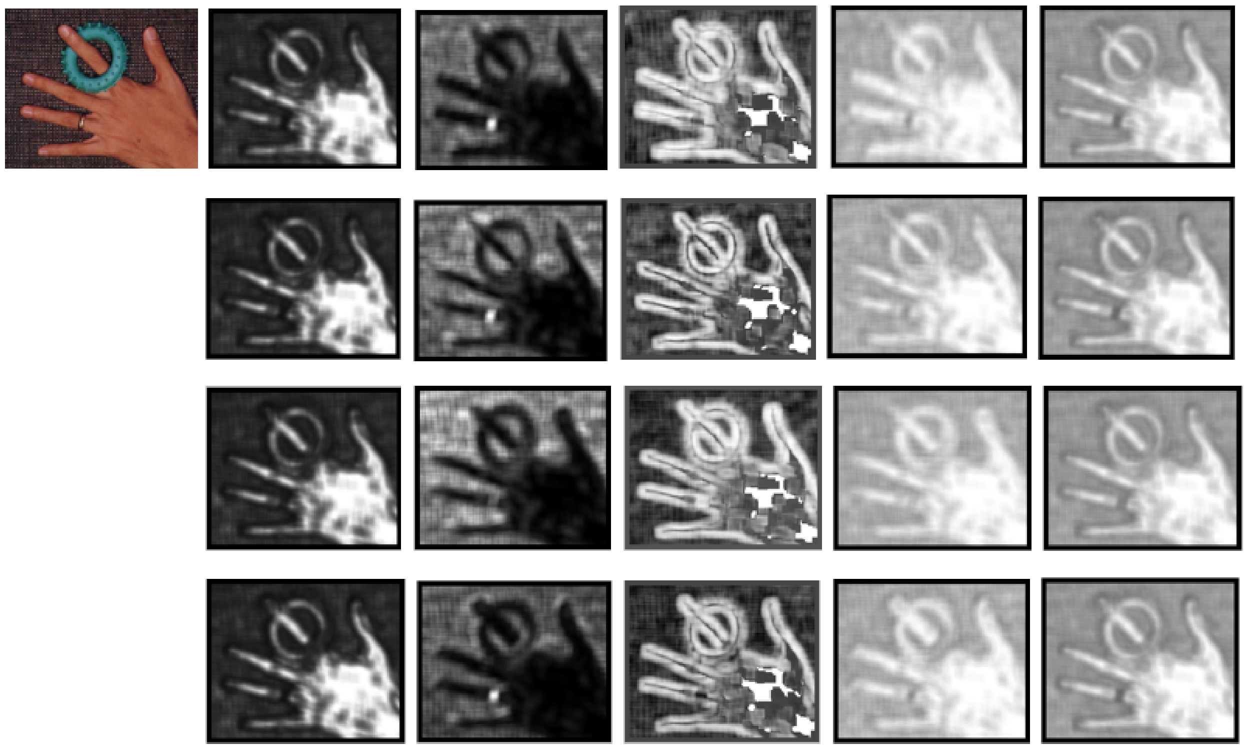


Figure 4: Hand Image texture features with patchsize 15 × 15 and offset distance 2
From left to right: Energy, Contrast, Correlation, Homogeneity and Entropy
Orientations from top to bottom: 0° , 45° , 90° and 135°

4. *PingPong Texture Image with gray level = 8*

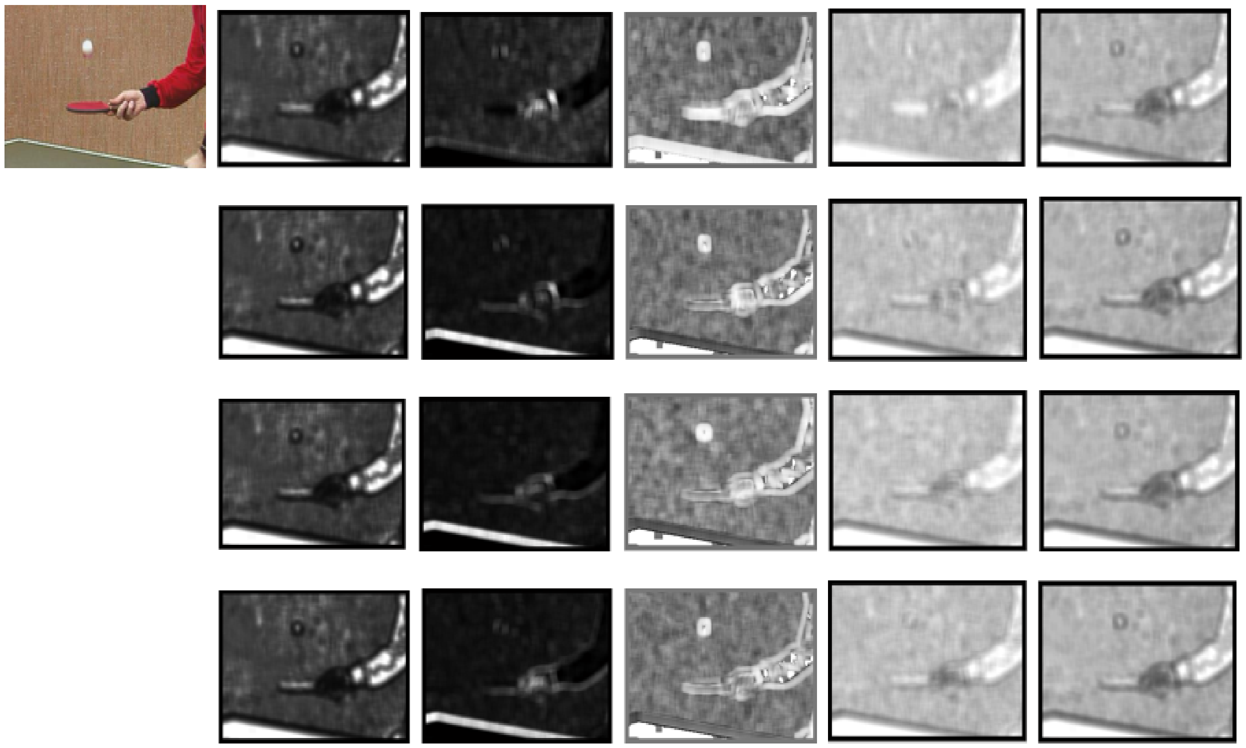


Figure 5: PingPong Image texture features with patchsize 15 × 15 and offset distance 2
From left to right: Energy, Contrast, Correlation, Homogeneity and Entropy
Orientations from top to bottom: 0° , 45° , 90° and 135°

5. *Feli Texture Image with gray level = 4*

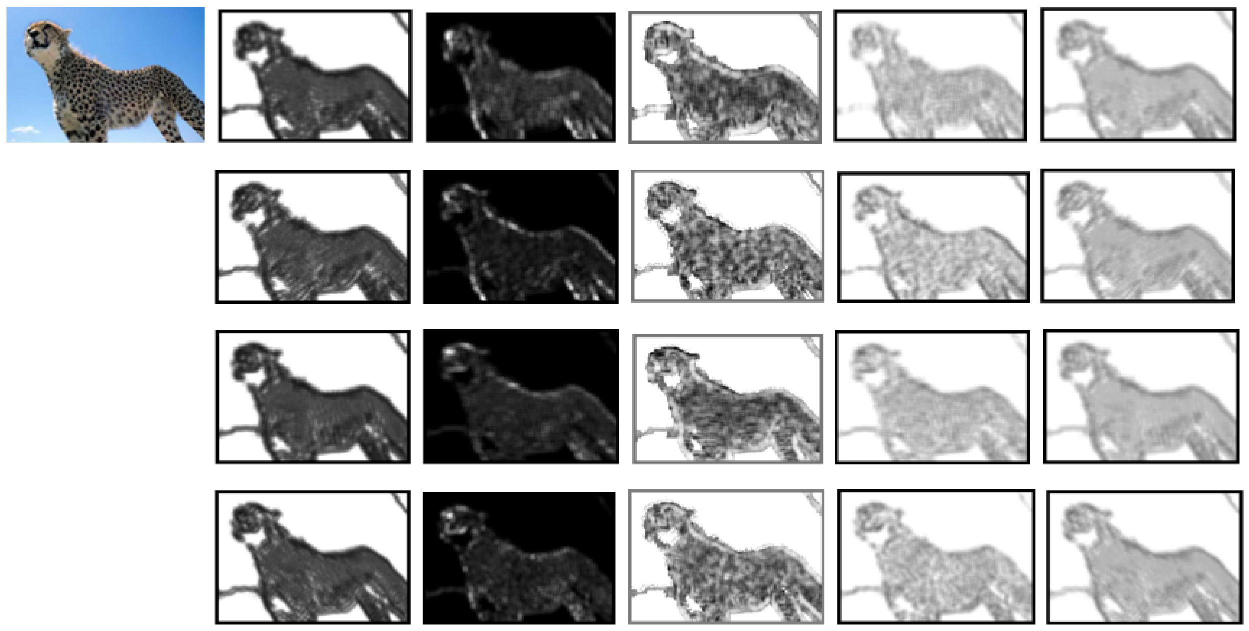


Figure 6: Feli Image texture features with patchsize 15 × 15 and offset distance 5
From left to right: Energy, Contrast, Correlation, Homogeneity and Entropy
Orientations from top to bottom: 0° , 45° , 90° and 135°

6. *PingPong Texture Image with gray level = 4*

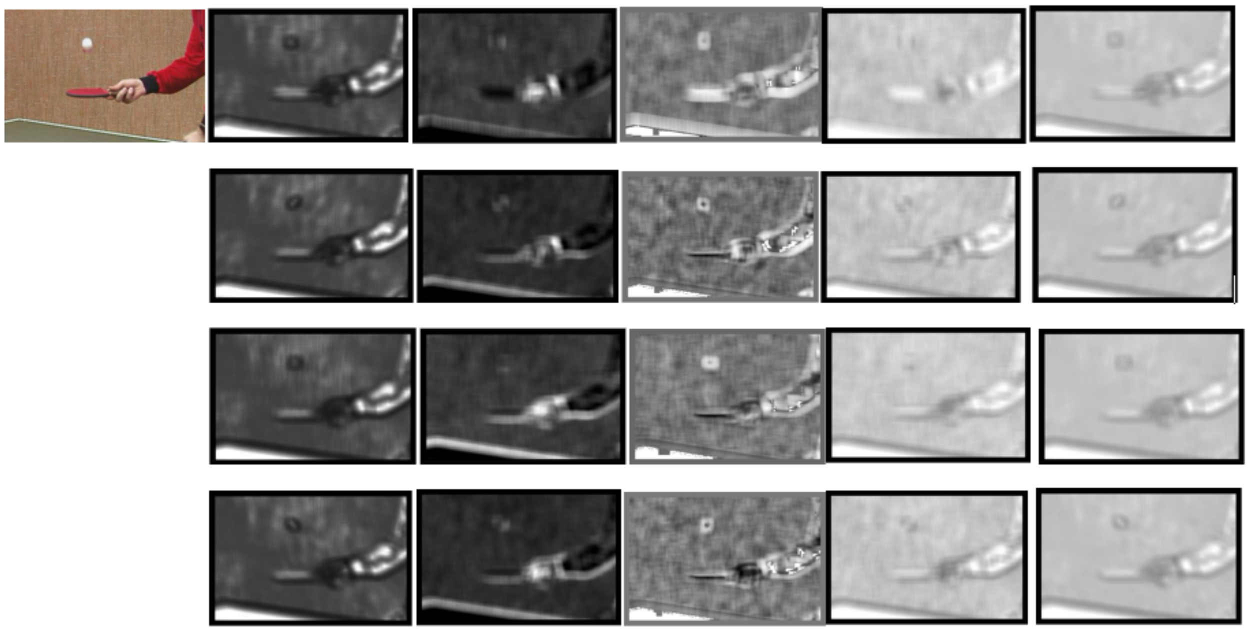


Figure 7: PingPong Image texture features with patchsize 20 × 20 and offset distance 5
From left to right: Energy, Contrast, Correlation, Homogeneity and Entropy
Orientations from top to bottom: 0° , 45° , 90° and 135°

Based on the above texture based feature results (local approach), we observe that -

- **Patch size**: Increase in the patch size blurs the image.
- **Distance offset**: Increasing this offset gives regions with same texture and are seen to be homogeneous and this can be used to our advantage.
  Lowering the value gives us more homogeneous areas.
- **Gray-levels**: Decreasing this value deteriorates the output image.
- **Angle offset**: It highly depends on the type of regions in the image and their surface boundaries.

5 Segmentation using Region Growing Algorithm

In the previous section we saw different texture results of different images with varying window size, offset (both in distance and orientation) and number of gray-levels. Based on the above results we now add different features into our region growing algorithm.

5.1 Segmentation Results

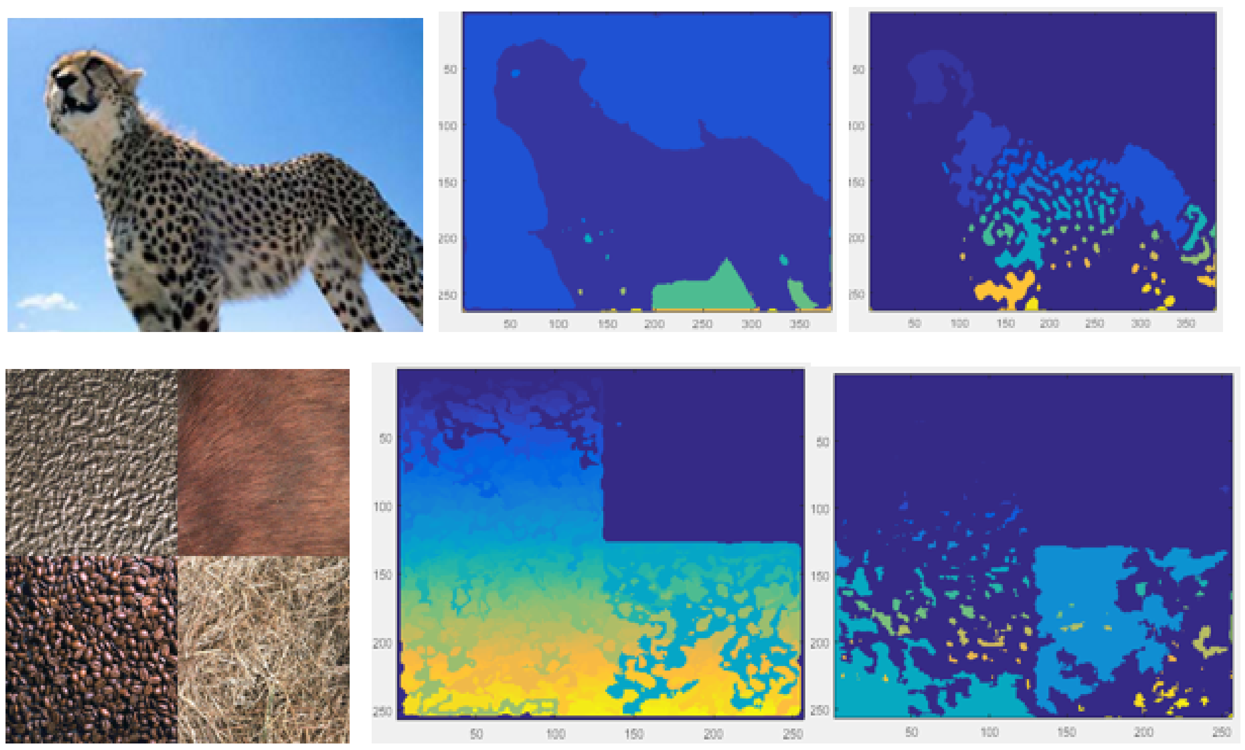


Figure 8: Segmentation results with and without features, From left: Original image, Segmentation with features and Segmentation without features (intensity based)

1. For the first image - **Feli** (Fig.8), we took patch size: 15×15, smoothing filter window: 7×7, features used for segmentation: Energy and Homogeneity each at distance: 2, angle: 45°.

2. For the second image - **Mosaic** (Fig.8), we took patch size: 5×5, smoothing filter window: 7×7, features used for segmentation: Contrast at distance: 2, angle: 0° and Contrast at distance: 2, angle: 90°.

3. In third image - **Hand** (Fig.9), we took patch size: 15×15, smoothing filter window: 7×7, features used for segmentation: Contrast and Homogeneity each at distance: 2, angle: $0^{/}circ$.

4. In the final image - **PingPong** (Fig.9), we took patch size: 15×15, smoothing filter window: 7×7, features used for segmentation: Energy and Entropy each at distance: 2, angle: $45^{/}circ$

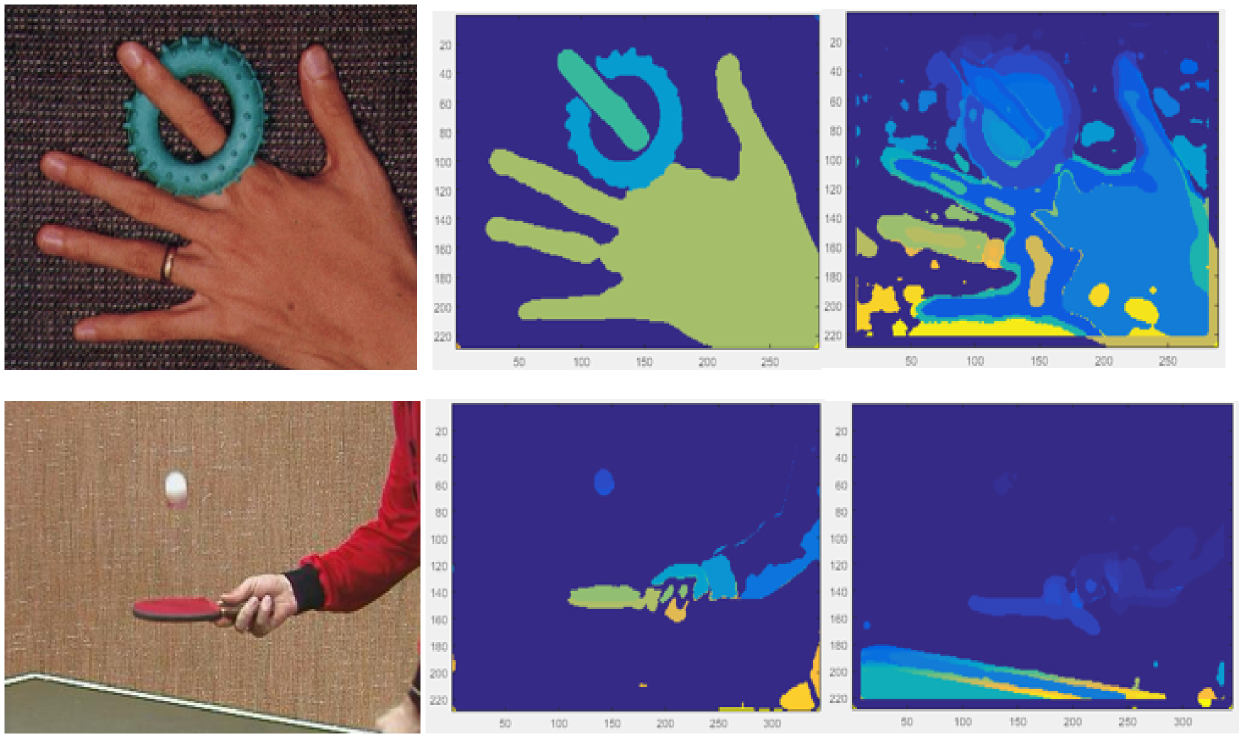


Figure 9: Segmentation results with and without features, From left: Original image, Segementation with features and Segmentation without features (intensity based)

From the above results, we observe that texture regions are represented as one or less regions. With the addition of texture based features along with intensity feature, it certainly improves segmentation result but not a perfect segmentation result.

6 Classification Problem

In this section we discuss about classification of images based on different surfaces using texture features. The goal was to produce a vector of features for an image, and use this information to classify the images in different classes

For classification we used MatlabPRtools, which is a toolbox for pattern recognition. We were provided with the script *scr_classifyPR.m* that implements K-Nearest Neighbor means classifier. It basically builds a confusion matrix of the test data and then displays the fraction of the correct classification. This script calls the functions *computeFeatureVector.m* and *computeIntensityFeatureVector.m*, which extracts the features from the images and their intensity values in the form of vector.

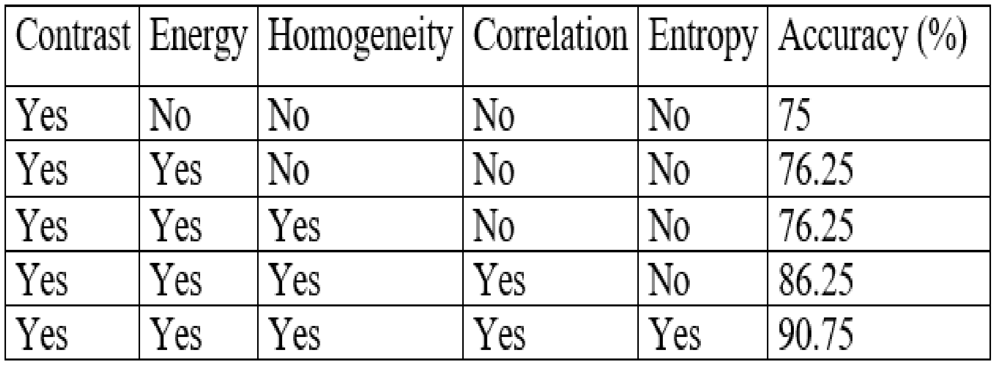
1. *computeIntensityFeatureVector.m* - Extracts mean intensity of all the columns in the matrix and builds the feature vector.
2. *computeFeatureVector.m* - Extracts global texture characteristics using GLCM and builds the feature vector.

We were provided with data set of 20 different type of images, each containing 6 images. We used 4 images out of 6 for training and the remaining 2 images for classification testing.

7 Classification Results

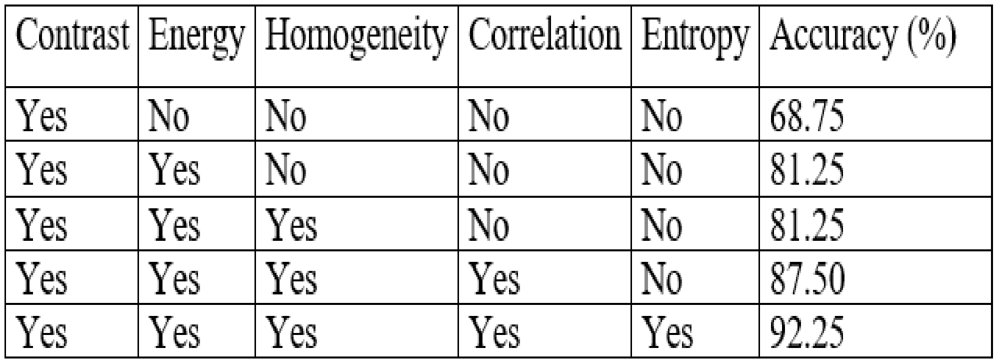
- First we performed classification based solely on intensity feature vector. And we achieved an accuracy of 96.25% for proper classification of images (using *computeIntensityFeatureVector.m*).
- Then we performed classification based on global texture features. An accuracy of 92.25% was achieved for successful classification.
- Finally, we performed classification based on both intensity features as well as texture based global features. We got an accuracy of around 96%.

In Fig.10 and 11 we performed classification using different combination of global texture based features and the resultant accuracy for each combination is shown in the table.



Contrast	Energy	Homogeneity	Correlation	Entropy	Accuracy (%)
Yes	No	No	No	No	75
Yes	Yes	No	No	No	76.25
Yes	Yes	Yes	No	No	76.25
Yes	Yes	Yes	Yes	No	86.25
Yes	Yes	Yes	Yes	Yes	90.75

Figure 10: Texture based classification, GLCM size: 10



Contrast	Energy	Homogeneity	Correlation	Entropy	Accuracy (%)
Yes	No	No	No	No	68.75
Yes	Yes	No	No	No	81.25
Yes	Yes	Yes	No	No	81.25
Yes	Yes	Yes	Yes	No	87.50
Yes	Yes	Yes	Yes	Yes	92.25

Figure 11: Texture based classification, GLCM size: 8

From the above table we observe that accuracy increases when all the features are considered. Also the above results can be improved by varying parameters like distance and angle offset.

Note: Here we used different orientations and distance offset combinations just to increase the range, so that the accuracy can be improved.

8 Organization and development

The lab coursework was completed in two weeks. In the first week, we started with the classification task using texture based features. We became familiar firstly with Matlab's inbuilt functions for GLCM and PRtools. Matlab's inbuilt texture feature function made our work load little easier and we were able to extract global texture features of each image easily by referring to Matlab documentation examples. Additionally, we wrote the code for Entropy feature ourselves based on the slides taught to us during the class.
Then in the second week we proceeded with the local texture feature extraction of different images with different patch sizes, orientation, distance and gray-levels. Later, we incorporated these features into our region growing algorithm which was implemented during Lab1 coursework.
In the third week we spent time testing with different parameters and comparing the output. The report writing was completed in the last 2-3 days of the third week.

9 Conclusion

In this lab coursework we worked on both local and global approach of texture based features extraction of images also their applications in segmentation and classification respectively. We enhanced our region growing algorithm for segmentation by incorporating local texture features for pixels. We also applied global texture based features to classify our image data set. Accuracy for classification increased when texture based features were included along with image intensity information. We learnt the importance of texture as an image characteristic through experiments and results shown in the previous sections. This lab coursework has solidified our knowledge in image characterization using texture.