

**Name: Ameer Ayman Ahmed**

**ID: 135058**

**Major: CS**

**Problem**

We have dataset of different image these images categorized in three classes. Class A, B and C. each class has three images and we have sample image. The three class express the training images and the sample image is for testing. So, we need to classify the sample image to which class it belongs to. We can classify the image by extract the features from each training set of images and the testing image then comparing between the features for the training images and the testing image. But, we have a problem that the input image is in spatial domain.

**Solution**

We need to classify the input image by extracting the essential features from it. We use Fourier transform which is image processing tool to transform the training and testing images from spatial domain to frequency domain. After transforming the images to frequency domain this step makes extracting features from image easier for us.

**Steps for classification images**

First, we read the images that we will using it in training

trainImg1=imread('A1.jpg');

Then we transform the image from rgb scale to grey scale.

img1Gray1=rgb2gray(trainImg1);

After that we read the testing image and if the testing image has noise we are using the compatible filter to remove this noise to make the image clear and then transform the sample image to gray scale also.

filteredImg=medfilt2(imgGray);

sampleImg=imcomplement(filteredImg)

After reading the training and testing images and transforming them to gray scale, we take each image from the training set and converting the image to double values then apply the fast Fourier transform on the image. After that taking the result values from the transformation and get the absolute of them to ignore any negative values if founded.

fourierTransform9=fft2(double(img1Gray9));

featuresOfImg9=abs(fourierTransform9(:));

Now we have the frequency domain or the features of the image so, we are sorting these frequency in descending order and take the highest three frequency values from it. We can take more than three features but, the highest three represent the major details. We do these steps also for the sample image.

featuresOfImg9=sort(featuresOfImg9,'descend');

featuresOfImg9=featuresOfImg9(1:3);

After these steps we save the highest three frequency values for each image into array then looping on the array that contains the features for every image.

features=[featuresOfImg1,featuresOfImg2,featuresOfImg3,featuresOfImg4,featuresOfImg5,featuresOfImg6,featuresOfImg7,featuresOfImg8,featuresOfImg9];

To classify the testing image, we use the K-nearest neighbor algorithm for classification that is working by calculating the nearest value from the training image to the testing image by calculating function of Euclidian distance.

for i=1:9

nearest(i)=sqrt((sampleImagefeatures(1)-features(1,i))^2+(sampleImagefeatures(2)-features(2,i))^2+(sampleImagefeatures(3)-features(3,i))^2);

end

The nearest value of the testing to the training image will be it.