Name: Kuimu Ren
USC ID Number:1473482531
USC Email:kuimuren@usc.edu
Submission Date: 03/28/2024

Problem1

Texture Analysis

Abstract and Motivation:

Spatial Texture analysis is a fundamental research area in the fields of image processing and computer vision, aimed at describing and understanding the texture features present in images. Texture typically refers to the repetitive and structured patterns found in surfaces.

Approach and Procedures:

First, we need to set up the Laws filter. Laws' filters are a set of 1D convolution kernels designed for texture analysis. These filters are used to capture various textural properties such as level, edge, spot, wave, and ripple in an image. The Laws' filters are constructed by combining five 1D kernels, each representing a different textural property:

L5: L5 = [1,4,6,4,1] Level filter, emphasizing areas with constant intensity.

E5: E5 = [−1,−2,0,2,1] Edge filter, highlighting edges or sharp transitions in intensity.

S5: S5 = [−1,0,2,0,−1] Spot filter, detecting small regions or spots with intensity variations.

W5: W5 = [−1,2,0,−2,1] Wave filter, capturing repetitive patterns or wave-like structures.

R5: R5 = [1,−4,6,−4,1] Ripple filter, identifying repetitive ripples or oscillations in intensity.

Second, extend the image boundary with 2 pixels width for each edge by reflection and subtract image mean to reduce illumination effects. Then, use the 2D filter obtained above, convolute the input image and get a filtered image. Calculate the square sum as the representation of the energy using the following equation.

$$average\ energy = \sqrt{\frac{1}{MN}\sum_{i=1}^{i=M}\sum_{j=1}^{j=N} I(i,j)^2}$$

M represents the height and N represents the width of the image. "Square sum" represents the texture property obtained by each filter.

Third, we can use PCA to reduce the dimensionality of the feature vectors from 25 to 3. PCA is a common technique used for dimensionality reduction. It accomplishes this by linearly transforming high-dimensional data into a lower-dimensional space while preserving the maximum variance in the data. Through dimensionality reduction, complexity and storage space of the data can be reduced, facilitating easier visualization and extraction of key features.

Fourth, using the nearest neighbor rule based on the Mahalanobis distance to realize texture classification. We need to do the same step for our test cases fist. Then, compute the mean vector and covariance matrix for each class in the training dataset. We iterate over each class, extract the features of samples belonging to that class, compute the mean vector and covariance matrix,

and store them separately. The Mahalanobis distance between the test feature vector and the class mean vector is then calculated using the corresponding covariance matrix. Finally, the nearest neighbor method is used to classify the test image based on Mahalanobis distance. For each test feature vector, it computes the Mahalanobis distance to each class mean vector and selects the class with the smallest distance as the predicted label for that test sample

$$D_M(\vec{x}) = \sqrt{(\vec{x} - \vec{\mu})^T S^{-1} (\vec{x} - \vec{\mu})}$$

$$S = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})' (X_i - \bar{X})$$

D is Mahalanobis Distance and S is Covariance Matrix

Last, we can also use Unsupervised Method(K-means clustering) and Supervised Method(Support Vector Machine) to classify the 25-D features and 3-D features of the image.

Experimental Results:

Shown below are the results of Problem 1:

Figure1: 25-D feature vector for each image

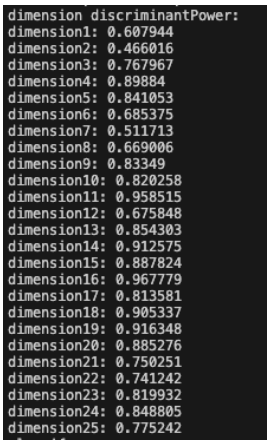

Figure2: discriminant power of each dimension in 25-D features vector

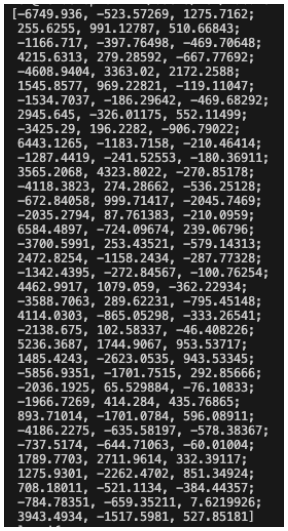

Figure3: 3-D feature vector for each image

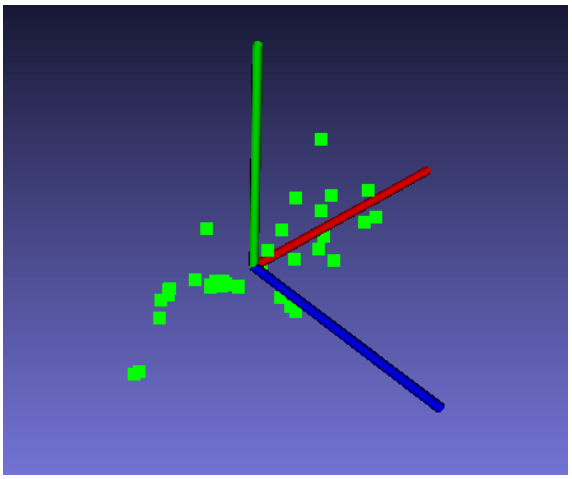
Figure4: reduced 3-D feature vectors of train data in the 3-D feature space

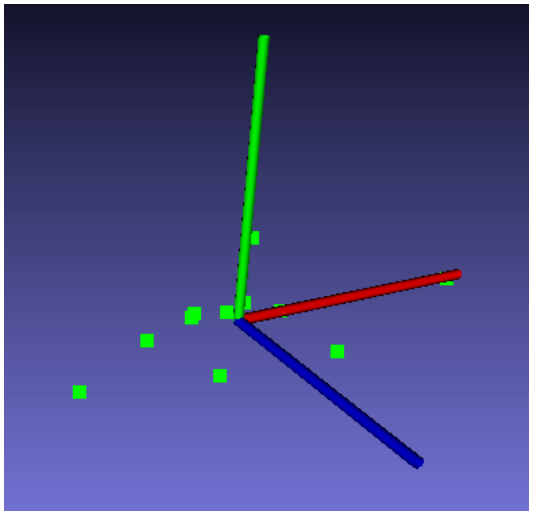

Figure5: reduced 3-D feature vectors of test data in the 3-D feature space

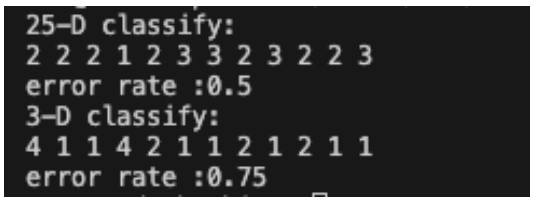

Figure6: test error rate by the nearest neighbor rule

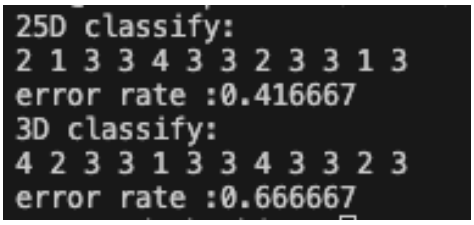

Figure7: test error rate by K-means

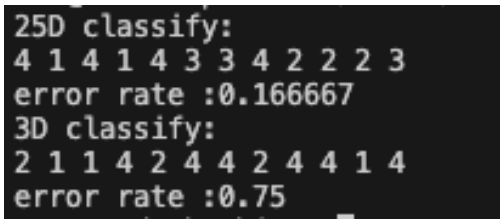

Figure8: test error rate by SVM

Homework Answer and Discussion:

We used 5x5 Laws Filters to extract the response vectors and calculate the average energy feature, then we can obtain the discriminant power of the 25-D feature by computing the energy feature. Because discriminant power is equal to intra-variety divided by inter-variety, we can find that the smaller discriminant power is the better feature is found. From figure2, we can find that the feature of dimension2 is the best, while the feature of dimension16 is the worst.

Reduce the feature dimension from 25 to 3 using the principal component analysis (PCA). Then, from figure4, 5, we can observe the distribution of reduced 3-D feature vectors in 3-D feature space of train data and test data respectively. We can find that the distribution of features according to the X-axis is more obvious. The distribution of features according to the other two axes is not obvious.

Then we will use the nearest neighbor rule, K-means and SVM for texture classification. We can obviously find from figure6,7,8 that the nearest neighbor rule and K-means methods are not very effective on the samples we tested. The correct rate is maintained at about 50%, while the error rate of SVM is only 17%. We can also find that compared with the data after using PCA dimensionality reduction, the accuracy of the original data is higher, which is likely because the reduction from 25D to 3D leads to the loss of a lot of important information carried by energy feature, thus affecting the judgment of the classifier

Problem2

Texture Segmentation

Abstract and Motivation:

Texture segmentation is a fundamental problem in the fields of image processing and computer vision. It aims to partition an image into regions with similar texture properties based on the texture features present in the image. Texture refers to visual characteristics in an image that exhibit local spatial repetition, such as color, brightness, texture orientation, and frequency.

A) Basic Texture Segmentation

Approach and Procedures:

First, we need to subtract the local mean and extract response vectors just like we did in problem one, then we can get a 25-D response vectors.

Second, we need to compute the energy measure for each center pixel based on step1. We have to choose different window sizes to slide on the response vectors and use the window averaging result as the central pixel's value by following equation. Then we can get a 25-D energy map.

$$average\ energy = \sqrt{\frac{1}{MN}\sum_{i=1}^{i=M}\sum_{j=1}^{j=N} I(i,j)^2}$$

Third, because all of the filters' responsive results have zero-mean except for L5L5 filter, all the other filter's results can be normalized and L5L5 results can be discarded. Then we can get a 24-D feature vector.

Last, apply K-means cluster with 24-D feature vector, and we can get 5 different classes. Using different colors to mark the different textures in the image to observe the results.

Experimental Results:

Shown below are the results of Problem 2(a):

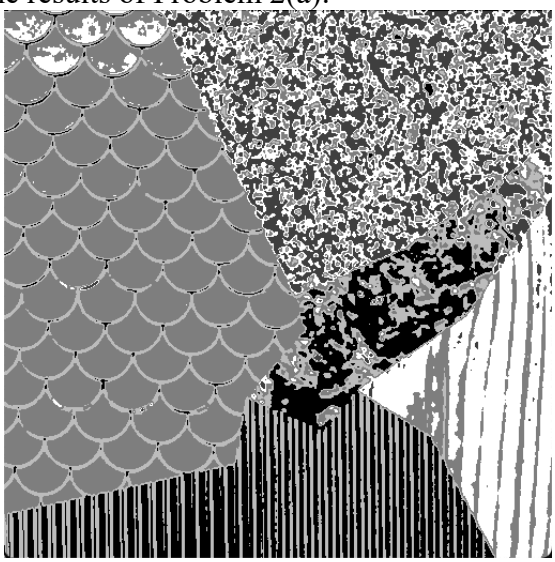

Figure1: Texture Segmentation with windowSize(3)

Figure2: Texture Segmentation with windowSize(9)

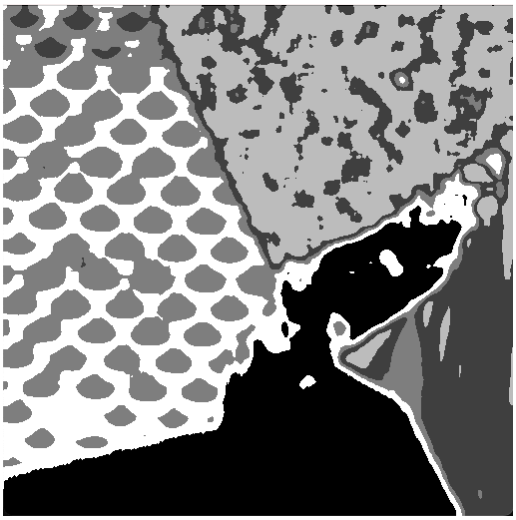

Figure3: Texture Segmentation with windowSize(15)

Figure4: Texture Segmentation with windowSize(35)

Figure5: Texture Segmentation with windowSize(55)

Homework Answer and Discussion:

Figure 1 to 5 shows the segmentation result of different window size. From the figure, we can observe that the larger window size is, the better the effect of image segmentation. When window size = 55, it shows the a great result. However, we can find that there is a region in the graph that is not segmented correctly. It is likely that the boundary of the graph is separated into a class by K-means algorithm, and the middle region may be ignored because it is too small or not obvious.

B) Advanced Texture Segmentation

Approach and Procedures:

First, all the steps are the same as in A.

Second, before we perform K-means, we need to perform PCA dimensionality reduction on the 24-d future vector, and then we can obtain the 3-d future vector, which can then be classified using K-means algorithm.

Third, we can also apply morphological closing operation on image result. This operation expands and then corrodes the regions in the image, effectively filling small holes or gaps within the regions.

Experimental Results:

Shown below are the results of Problem 2(b):

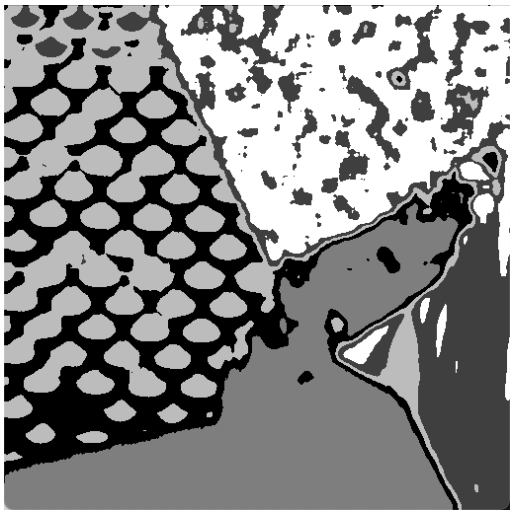

Figure6: Texture Segmentation with windowSize(15) by using PCA

Figure7: Texture Segmentation with windowSize(15) by using PCA and post-processing

Figure8: Texture Segmentation with windowSize(35) by using PCA

Figure9: Texture Segmentation with windowSize(35) by using PCA and post-processing

Figure10: Texture Segmentation with windowSize(55) by using PCA

Figure11: Texture Segmentation with windowSize(55) by using PCA and post-processing

Homework Answer and Discussion:

Figure 6,8,10 shows the texture segmentation applied to different Windows after PCA dimensionality reduction. Figure 7,9,11 shows the texture segmentation applied to different Windows after PCA dimensionality reduction and post-processing. We can find that the effect of texture segmentation does not change much after PCA processing, so PCA is not suitable for this experiment. On the contrary, by developing a post-processing technique to merge small holes, some misclassifications classified to outlying areas are eliminated, showing better results.

Problem3

SIFT and Image Matching

Abstract and Motivation:

The primary goal of the SIFT algorithm is to extract image features that are invariant to scale, rotation, and illumination changes, and utilize them for tasks such as image matching, object recognition, and reconstruction.

(a) Salient Point Descriptor

Homework Answer and Discussion:

1. The SIFT algorithm is robust to various geometric modifications, including scale changes, rotation, translation, affine transformations and so on.
2. Scale Changes: SIFT features are detected and described at multiple scales, allowing them to be robust to changes in the size of objects in the image. This means that the same feature can be detected at different scales.
   Rotation: SIFT features are invariant to rotation, meaning they can be detected even when the image has been rotated.
   Translation: While not explicitly invariant to translation, SIFT features can still be matched across images with slight translation variations.
   Affine Transformations: SIFT features are robust to affine transformations such as shearing and stretching, although they may not be completely invariant to these transformations.
3. SIFT's robustness to illumination changes stems from its focus on local gradient-based descriptors, its scale-invariant feature detection approach, and its local matching strategy, all of which reduce the reliance on absolute pixel intensities and enhance the algorithm's ability to capture and match distinctive local image structures.
4. The use of the Difference of Gaussians in the SIFT algorithm offers computational efficiency, smoother scale space representation, improved localization of keypoints, and robustness to noise, making it a preferred choice for scale-invariant feature detection and description.
5. The SIFT algorithm outputs a feature vector of size 128 for each keypoint detected in the image.

(b) Image Matching

Approach and Procedures:

I will extract SIFT features from images and visualize the matching between the keypoint with the largest scale in one image and its closest neighbor in another image.

First, we need to extracts SIFT features, including keypoints and descriptors, from the input images and then we use the descriptor to select the keypoints that do well.

Second, we need to identifies the keypoint with the largest scale in the first image and records its index.

Third, finds the keypoint in the second image that is closest to the keypoint with the largest scale in the first image.

Fourth, Draws the keypoints corresponding to the largest scale in the first image and the closest neighboring keypoint in the second image.

Fifth, matches the extracted features.
Last, draws and displays the matching results.

Experimental Results:

Shown below are the results of Problem 3(b):

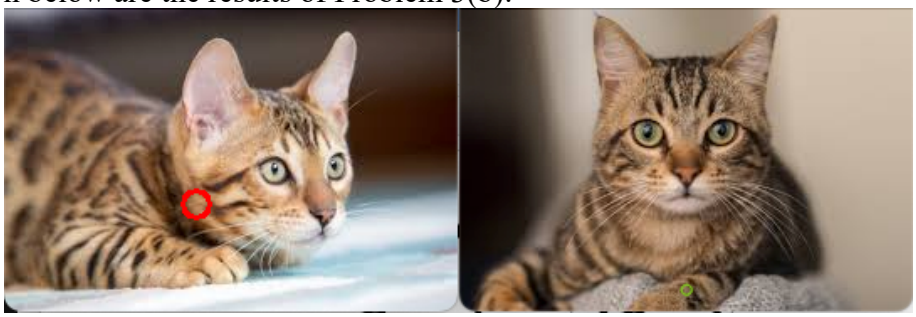

Figure1: largest scale in Cat_1 and closet neighboring key point in Cat_3

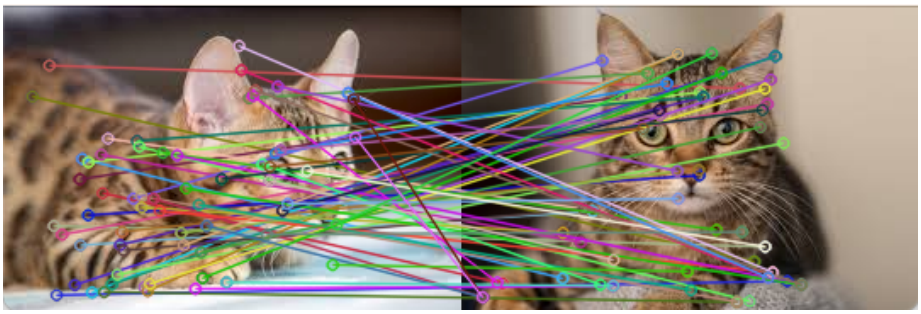

Figure2: corresponding SIFT pairs between the Cat_1 and Cat_3

Figure3: largest scale in Cat_3 and closet neighboring key point in Cat_2

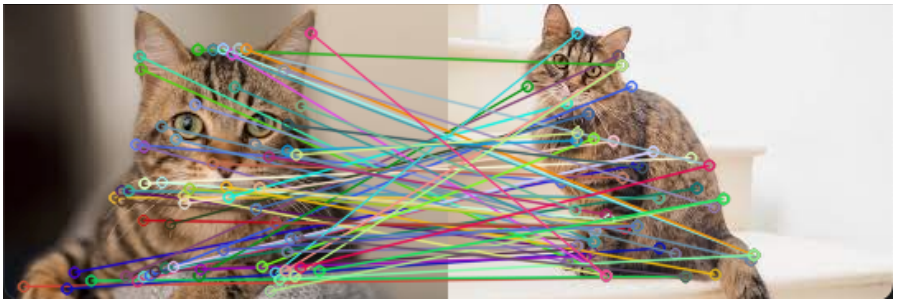

Figure4: corresponding SIFT pairs between the Cat_3 and Cat_2

Figure5: largest scale in Dog_1 and closet neighboring key point in Cat_3

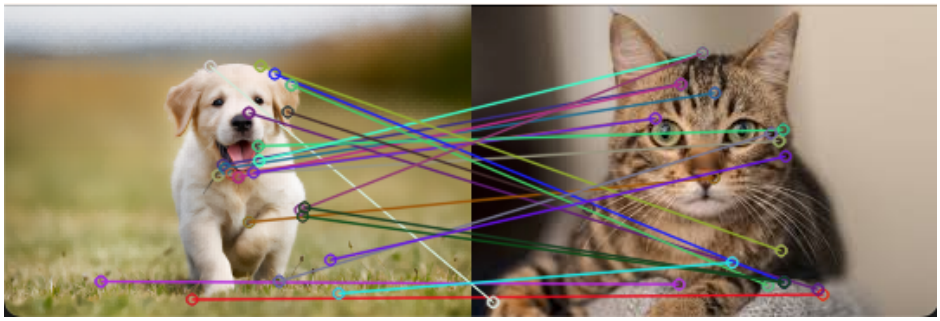

Figure6: corresponding SIFT pairs between the Dog_1 and Cat_3

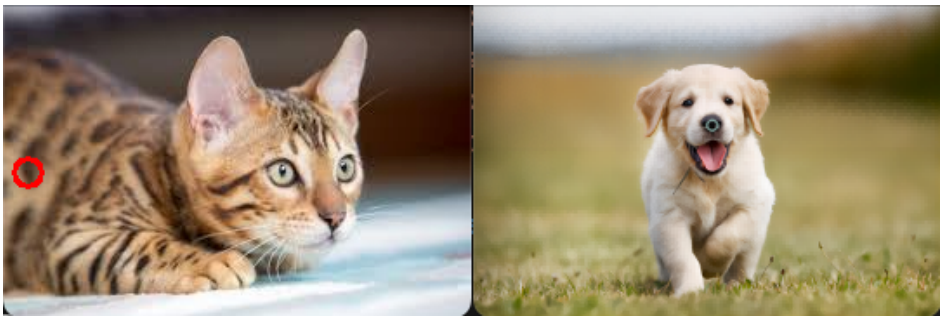

Figure7: largest scale in Cat_1and closet neighboring key point in Dog_1

Figure8: corresponding SIFT pairs between the Cat_1 and Dog_1

Homework Answer and Discussion:

From figure 1 to 8, we can find that in SIFT, matching may not perform well between different objects or the same object viewed from significantly different angles. The success or failure of matching depends on the similarity between objects and the extent of viewpoint differences. For objects that are similar and have similar viewpoints, matching tends to be more

accurate as they have similar descriptors in feature space. However, when there are significant differences between objects or large changes in viewpoint, SIFT may produce mismatches or fail to find matches because of the large differences in their feature descriptors, making it difficult to find suitable correspondences.

(c) Bag of Words

Approach and Procedures:

First, we need to extract SIFT features from all four image and, then reduce dimension of the 128-d feature vector to the 20-d feature vector by PCA.

Second, we need to apply K-means clustering to create 8 dictionary's visual words.

Third, for each feature in each of four images, we need to calculate the nearest neighbor of cluster center in the dictionary.

Fourth, then for each of four images, build a histogram of length k (frequency of words in that image).

Last, calculate the similarity index between different image and compare codeword histogram intersection of the three other images with Cat_3 image codeword histogram.

$$Similarity\ Index = \frac{\sum_{j=1}^{k} \min(I_j, M_j)}{\sum_{j=1}^{k} \max(I_j, M_j)}$$

Experimental Results:

Shown below are the results of Problem 3(a):

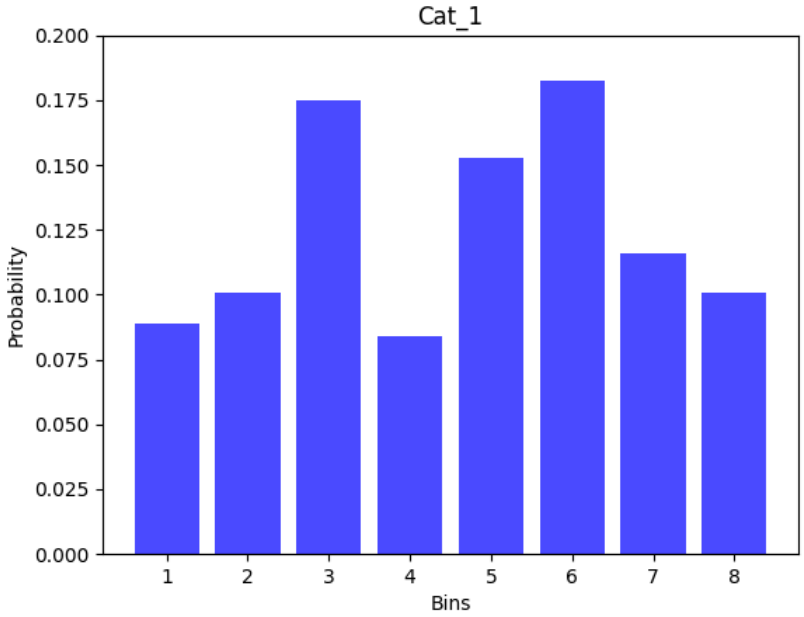

Figure1: codeword histogram for Cat_1

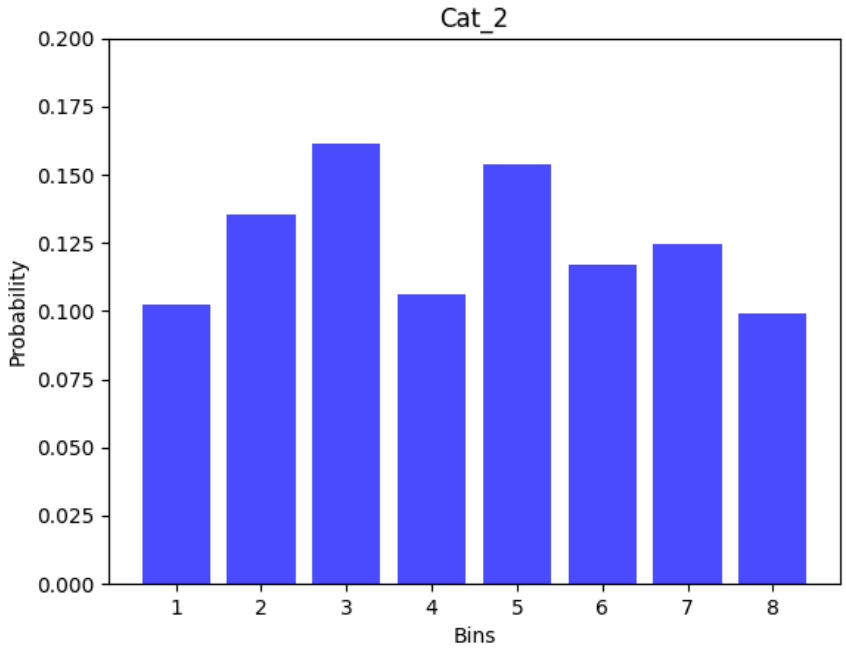

Figure2: codeword histogram for Cat_2

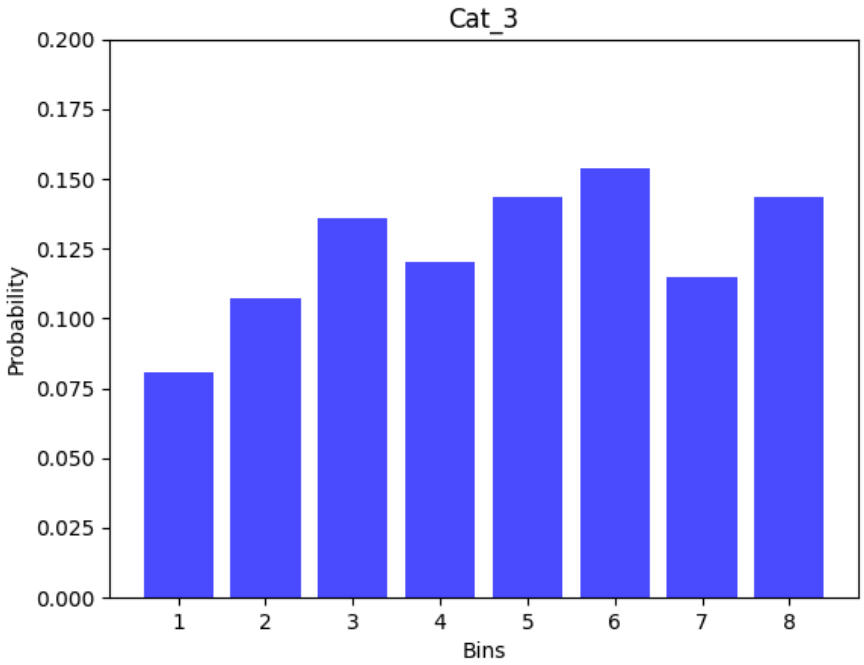

Figure3: codeword histogram for Cat_3

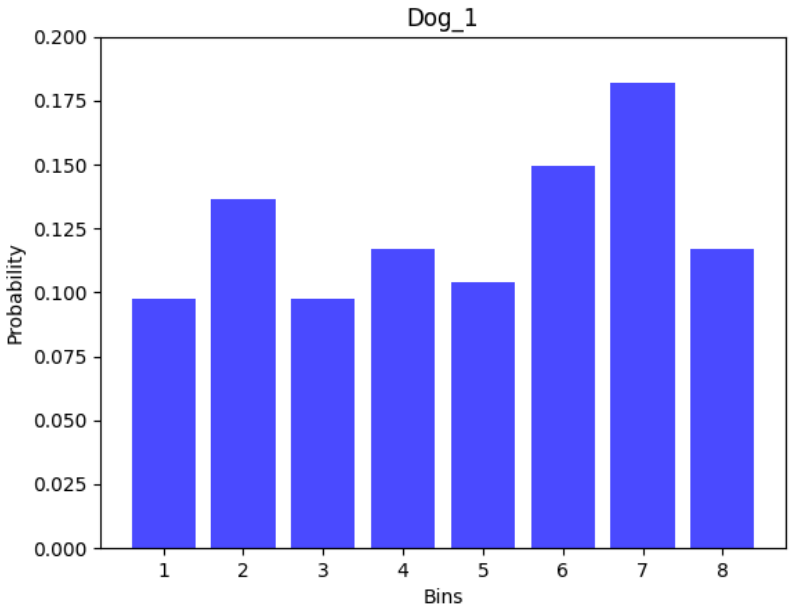

Figure4: codeword histogram for Dog_1

```
similarity between cat_3 and cat_1
0.843245
similarity between cat_3 and cat_2
0.8258
similarity between cat_3 and dog_1
0.79741
```
Figure5: similarity between different image

Homework Answer and Discussion:

From figure 1 to 4, we can find the histogram of different image, by comparing the histogram intersection, the similarity between images can be quantified to determine which images are more similar to Cat_3 images. From figure 5, we can find the similarity index between Cat_3 and other images. We can obviously find that the similarity of pictures of the same object is greater than that of pictures of different objects, but there may be some similarity between images of different objects, especially in similar scenes.