EE569 Digital Image Processing

HOMEWORK #4

Issued: 03/09/2022 Due: 11:59PM, 03/27/2022

Problem 1: Texture Analysis (35%)

1.1 Motivation

Texture analysis is the extraction and examination of spatial distribution patterns in grayscale images (intensity). Texture analysis is used in a variety of photographs, including remote sensing images, radiographs, and interpretation and processing of cellular images. There is no single mathematical model that describes the texture. It derives from the concept of textures that characterize the surface properties of textiles and is used to describe the placement of all material components, including lung textures in medical radiographs, vascular textures, and aerospace lithofacies textures. Available (or aerial) topographic photos. In image processing, visual textures are commonly viewed as a repeated arrangement of some fundamental pattern (tonal primitives). Describing a texture therefore entails identifying the tonal primitives that comprise the texture and determining their interrelationships. Texture is a geographical property, hence it is tied to the region's size and form. By examining if the texture metric changes considerably, the border between two texture patterns may be detected. Texture is a reflection of an item's structure, and studying texture may reveal essential information about the object in a picture, which is useful for image segmentation, feature extraction, and classification recognition. Texture analysis may be conducted on spatial domain pictures or transform domain images (see Image Transformation) using both statistical and structural approaches.

Statistical texture analysis searches for numerical characteristics that define the texture and classifies areas (rather than individual pixels) in the picture using these features alone or in conjunction with other non-texture information. Commonly utilized digital texture characteristics include the autocorrelation function of picture local areas, the grayscale co-generation matrix, the grayscale tour, and different grayscale distribution statistics. The grayscale co-occurrence matrix, for example, quantifies texture in terms of grayscale spatial distribution. Because the grayscale distribution of coarse textures fluctuates far more slowly with distance than that of fine textures, their grayscale covariance matrices are radically different.

The study of the primitives that make up a texture and their organization principles is known as structural texture analysis. A primitive might be the grayscale of a pixel or a linked group of pixels with particular attributes. Tree grammars are frequently used to explain the organization rules of primitives.

1.2 Approach

1.2.1 Texture Classification – Feature Extraction

Step 1. Load train and test images.

Step 2. Apply 25 5x5 Laws filters to get 25 filtered images.

Table.1 1D Kernel for 5x5 Laws Filters

Name	Kernel
L5 (Level)	[1 4 6 4 1]
E5 (Edge)	[-1 -2 0 2 1]
S5 (Spot)	[-1 0 2 0 -1]
W5 (Wave)	[-1 2 0 -2 1]
R5 (Ripple)	[1 -4 6 -4 1]

Step 3. Get feature vectors for training and testing images.

Step 4. Merge all feature vectors to a larger matrix and evaluate the discriminant power of train features.

Let y_{ij} be the jth observation in the ith class. The overall average is

$$\bar{y}_{\bullet\bullet} = \frac{\sum_i \sum_j y_{ij}}{\sum_i \sum_j 1}.$$

The average within class i is

$$\bar{y}_{i\bullet} = \frac{\sum_j y_{ij}}{\sum_j 1}$$

(where the number of values of j may depend on i, i.e. not all classes must have the same size).

Then the total corrected sum of squares is

$$\sum_i \sum_j (y_{ij} - y_{\bullet\bullet})^2.$$

The intra-class sum of squares is

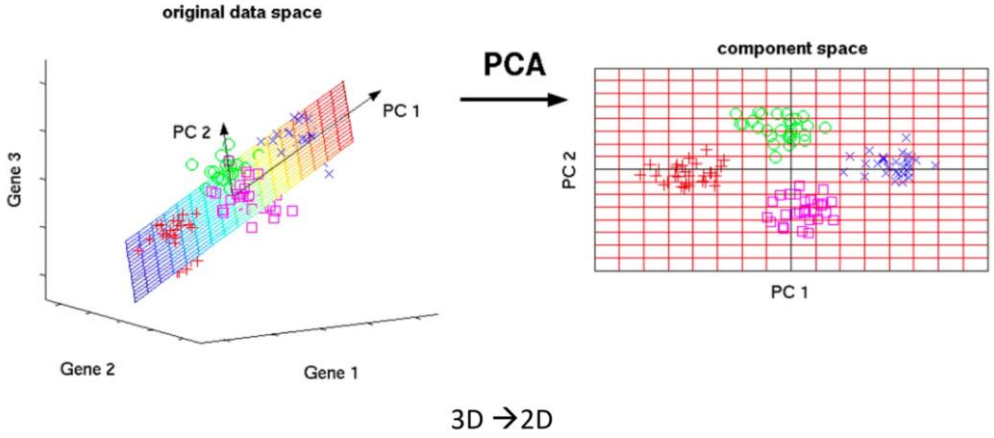
$$\sum_i \sum_j (y_{ij} - \bar{y}_{i\bullet})^2).$$

The inter-class sum of squares is

$$\sum_i \sum_j (\bar{y}_{i\bullet} - \bar{y}_{\bullet\bullet})^2 = \sum_i \left(n_i (\bar{y}_{i\bullet} - \bar{y}_{\bullet\bullet})^2\right)$$

where n_i is the number of observations in the ith class.

Step 5. Perform PCA on 25D feature set to reduce its dimension to 3.



3D →2D

Step 6. Visualize the data representation in the space of the first three principal components.

Step 7. Conduct texture classification using the nearest neighbor rule-based criterion.

Step 8. Calculate error rate of predictions.

Step 9. Save the 25-D and 3-D feature vectors obtained for next questions.

1.2.2 Advanced Texture Classification --- Classifier Exploration

Step 1. Load the 25-D and 3-D feature vectors obtained above.

Step 2. Merge all train and test features and labels.

Step 3. Apply the K-means algorithm to 25-D features and 3-D features for test images. Then compute the error rate for K- means.

Steps of K-means clustering:

1. Initialize the cluster centroids by randomly select K samples from the data
2. Calculate the distance between samples and centroids to check their similarity
   - ✓ $d(E_i, C_k) = \sqrt[2]{\sum_{j=1}^{m}(E_{i,j} - C_{k,j})^2}$, $where\ i = 1, \cdots, n\ and\ k = 1, \cdots, K$
   - ✓ Except Euclidean distance, you can try Mahalanobis distance, etc.
3. Compare the distance of the sample to K centroids to see which centroid has the minimum distance to it, label the sample to that cluster
   - ✓ $L_i = \underset{k \in \{1,2,\ldots,K\}}{\operatorname{argmin}} d(E_i, C_k)$

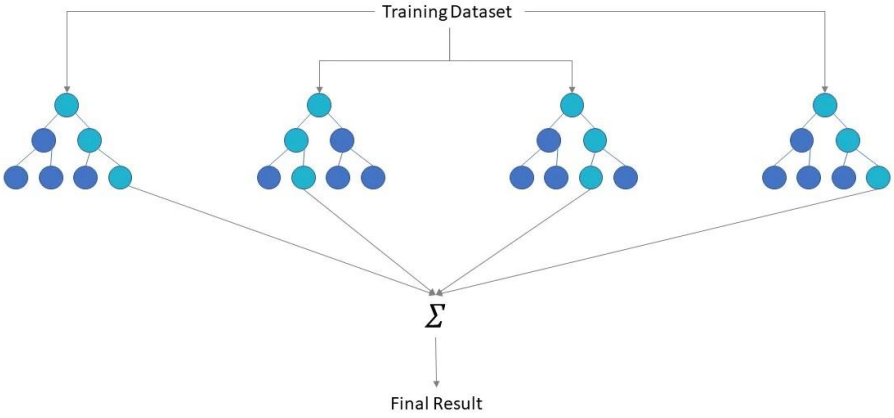
4. Obtain the new cluster centers

✓ $C_k = \frac{\sum_{i=1}^{n} l(L_i=k)E_i}{\sum_{i=1}^{n} l(L_i=k)}, where\ l(L_i = k) = \begin{cases} 1 & if\ L_i = k \\ 0 & if\ L_i \neq k \end{cases}$

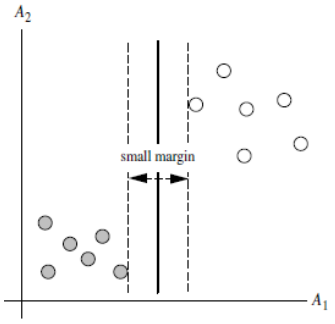
5. Repeat step 2-5 until change in cluster centroids are effectively negligible

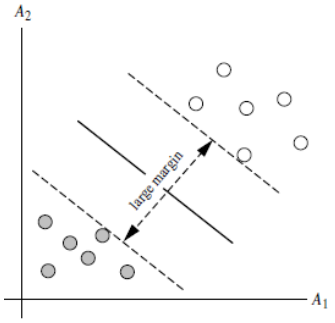
Step 4. Use the test labels to evaluate the purity of each cluster. Then compute the error rate for clustering.

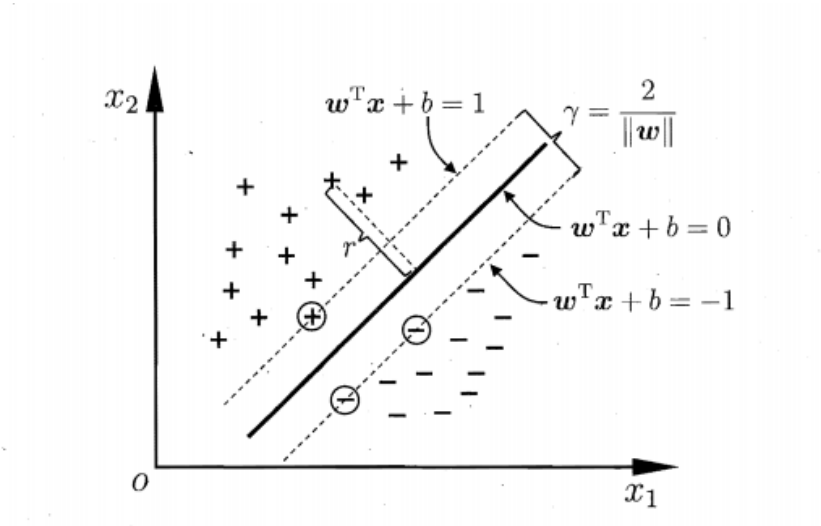
Step 5. Conduct supervised learning to train random forest by 3-D features. Then compute the error rate of predict of test labels.



Step 6. Conduct supervised learning to train Support Vector Machine by 3-D features. Then compute the error rate of predict of test labels.







1.3 Results

(a)

We find feature dimension 21, R5 * L5 has the strongest discriminant power. And feature dimension 5, L5 * R5 has the weakest discriminant power.

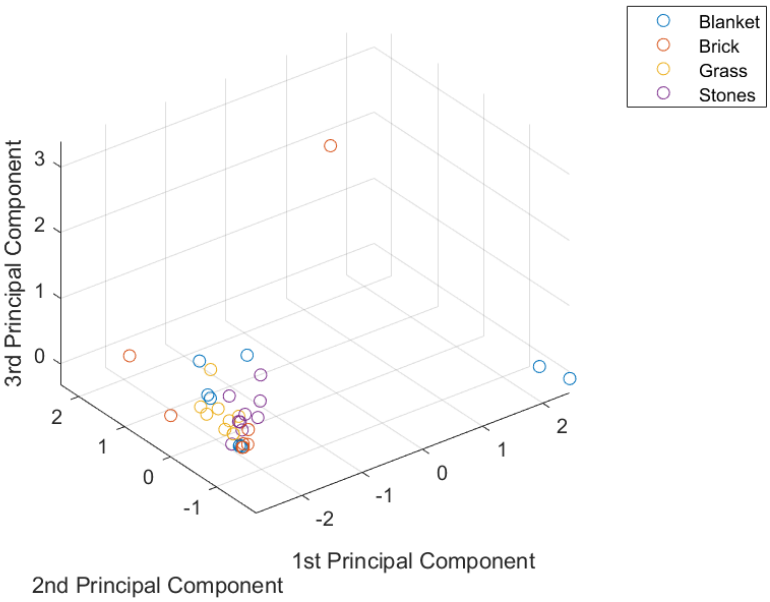


Figure 1: 3-D feature vectors

The error rate of predictions is 0.66667.

(b)

Unsupervised:

K-means: Error rate of K-means on 25-D feature is 0.5625. Error rate of K-means on 3-D feature is 0.625.

Using the test labels to evaluate the purity of each cluster: Error rate is 0.66667.

Feature dimension reduction is better because the results achieved is more consistent over K-means clustering.

Supervised:
Random Forest: Error rate of random forest is shown below:

```
The error rate for using 5 random forest is: 0.75
The error rate for using 10 random forest is: 0.75
The error rate for using 20 random forest is: 0.75
The error rate for using 50 random forest is: 0.75
The error rate for using 100 random forest is: 0.66667
The error rate for using 200 random forest is: 0.66667
The error rate for using 500 random forest is: 0.83333
```

Figure 2: Error Rate of Random Forest

Support Vector Machine: Error rate of SVM is 0.83333.

By comparing the two classifiers we can find that when we choose the numbers of trees be [5,10,20,50,100,200,500], the error rate of random forest is lower than Support Vector Machine. Which means random forest is a better supervised learning for texture classification.

1.4 Discussion

From the experiments we can see that the discriminative power of dimension 21 is the best because the data points are more evenly distributed in this feature vector. Dimension 5 is the least discriminative because the data points are more evenly distributed in this feature vector.

PCA (Principal Component Analysis) is an unsupervised approach to reducing the dimensions of data. The data is first transformed into a new coordinate system using a linear transformation. Then the dimensionality reduction idea is used, where the first large variance of the data projection is in the first coordinate and the second large variance is in the second coordinate. This dimensionality reduction concept first reduces the dimensions of the dataset and then intuitively displays the data in a two-dimensional coordinate system, while retaining the elements of the dataset that contribute most to the distribution.

The benefits of Random Forest are that it can provide high dimensional data (many features) without the need for dimensionality reduction or feature selection. He can evaluate the importance of characteristics. It can determine how distinct properties interact with each other. It's hard to over-scale. The pace of training is faster, and the parallel technique is easy to implement. It is very easy to put it into practice. It can compensate for errors in unbalanced data sets. Even in the absence of a large number of characteristics, it is still possible to maintain accuracy.

Random forests have proven to overfit one of their drawbacks, the noisy classification or regression problem. For data with attributes whose values change, the results of random forests with attribute weights are unreliable because the attributes whose values are split have a significant impact on the random forest.

According to the characteristics of the SVM algorithm, the computational complexity of training a model is defined by the number of support vectors, not by the dimension of the data. Therefore, SVMs are less susceptible to overfitting. The training model of SVM relies entirely on support vectors. Even after removing all non-support vector points from the training set and redoing the training process, the model remains the same. SVMs trained with the smallest number of support vectors are more likely to generalize.

Problem 2: Texture Segmentation (30%)

2.1 Motivation

Image segmentation is the process of dividing an image into numerous disjoint parts based on variables such as grayscale, color, spatial texture, and geometric form, so that these features exhibit consistency or resemblance within the same region while displaying clear contrasts across regions. In a nutshell, it is the separation of the target from the backdrop in two photographs.

Textured images exhibit irregularities within local regions, while showing some regularity in the whole. The arrangement of texture primitives may be random or may depend on each other, and this dependence may be structured, or may be arranged according to some probability distribution, or may be in some functional form. Image texture can be described in many qualitative terms, such as roughness, fineness, smoothness, directionality and regularity, granularity, etc. Early texture analysis relied on statistical or structural approaches to extract characteristics, with the majority of these methods focusing on texture analysis, such as the spectrum method, gray level coeval matrix method, gray level journey method, texture description model, texture grammar model, and so on. Based on prior work, scholars have developed a number of texture analysis approaches in recent years, with the advent of fuzzy mathematics, wavelets, fractals, and other theories. The primary techniques include fuzzy clustering-based classification models, neural network-based classification models, wavelet analysis and wavelet transform-based classification models, fractal theory-based classification models, mathematical morphology-based classification models, and so on.

In this experiment, texture segmentation based on clustering is used.

2.2 Approach

2.2.1 Basic Texture Segmentation

Step 1. Get 25 5x5 Laws Filters and apply these laws filters to the image after boundary expansion to extract the response vectors, which obtain 25 image responses.

Step 2. Calculate the average energy for each pixel in each image response within different window.

Step 3. Discard the feature associated with $L5^{T}L5$. Then normalize the left 24-D feature vector to $L5^{T}L5$ to get 24-D energy feature vectors.

Step 4. Apply K-means clustering to the 24-D feature vectors and divide the pixels into 6 clusters. Use the given randomly generated color map to represent the 6 regions.

	0	1	2	3	4	5
R	107	114	175	167	144	157
G	143	99	128	57	147	189
B	159	107	74	32	104	204

2.2.2 Advanced Texture Segmentation

Step 1. Get 25 5x5 Laws Filters and apply these laws filters to the image after boundary expansion to extract the response vectors, which obtain 25 image responses.

Step 2. Calculate the average energy for each pixel in each image response within different window.

Step 3. Discard the feature associated with $L5^{T}L5$. Then normalize the left 24-D feature vector to $L5^{T}L5$ to get 24-D energy feature vectors.

Step 4. Perform PCA on 24-D feature vectors to reduce its dimension to 2 and extract the feature vectors.

Step 5. Use the dimension reduced features to do texture segmentation using K-means clustering and divide the pixels into 6 clusters.

Step 6. Do post-processing using morphological process to merge small holes.

2.3 Results

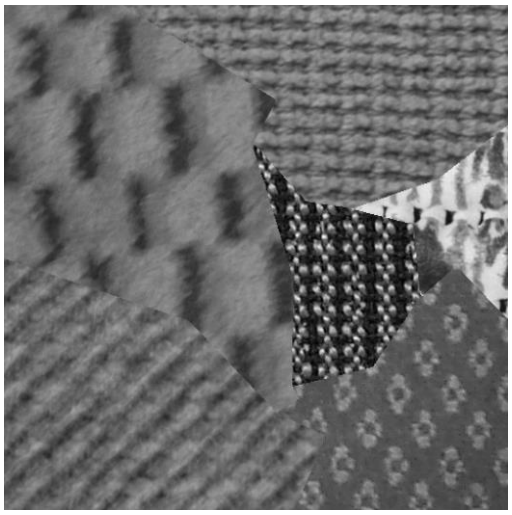


Figure 3: Original Image

(a)

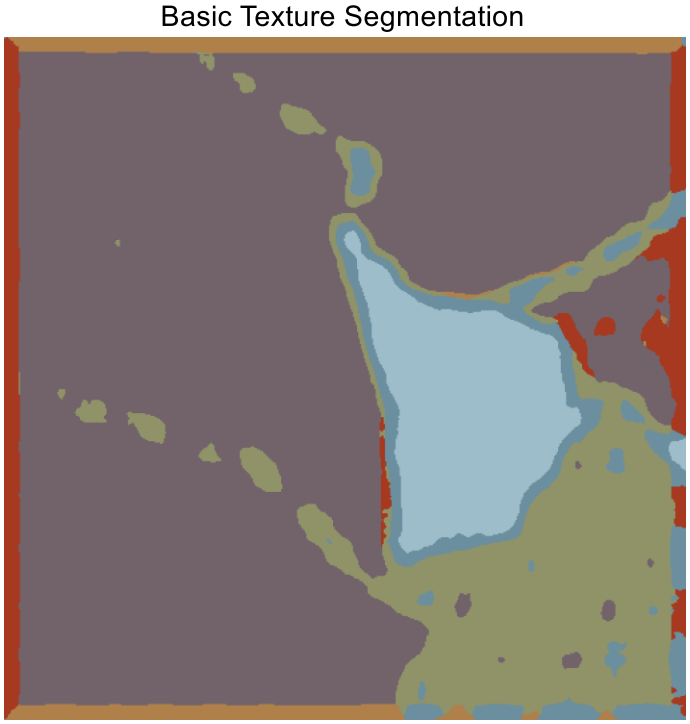


Figure 4: Basic Texture Segmentation

(b)
Use the PCA and do a processing process by morphological process to merge small holes.

Figure 5: Advanced Texture Segmentation

2.4 Discussion

K-means clustering based on intensity or color is simply picture vector quantization.

In order to enhance the pixel clustering effect, it is achieved by increasing the feature dimension and adding physical space information in addition to the color space based, this method can make the image segmentation more accurate but also bring more over-segmentation cases.

The disadvantage of k-means clustering is that k needs to be selected manually. k-means clustering is sensitive to the initialization method. Sensitive to outlier. Based on a spherical assumption.

Problem 3: SIFT and Image Matching (35%)

3.1 Motivation

Scale-invariant feature transformation (SIFT) is a computer vision technique that detects and describes local features in images by locating spatial-scale extreme points and extracting their position, scale, and rotational invariants. Applications include object identification, robotic map perception and navigation, picture stitching, 3D model construction, gesture recognition, image tracking, and motion comparison.

The description and detection of local image features are helpful for object recognition, and SIFT features are based on some local appearance regions of interest on objects that are independent of image size and rotation. They are also very sensitive to small changes in light, noise and viewing angles. Based on these features, they are highly visible and fairly simple to detect, making it easy to identify objects in large sets of attributes with minimal misrecognition rates. The detection rate for partial object occlusions using SIFT feature descriptions is also quite good, requiring only three or more SIFT object features to compute position and orientation. With today's computer hardware speed and small feature library, the recognition speed can be almost instantaneous, and the massive information in SIFT features is suitable for fast and accurate matching in large databases.

The essence of the SIFT algorithm is to find the key points (feature points) on different scale areas and calculate the orientation of the key points; The key points found by SIFT are some very prominent points that do not change due to illumination, affine shift, and noise, such as corner points, edge points, bright points in dark areas, and dark points in bright areas.

In the field of information retrieval, the bag of words model is a common way to represent a document. In information retrieval, the BoW model states that a document is treated as just a collection of a few words, ignoring its word order, grammatical, and syntactic aspects, and that the occurrence of each word in a document is independent and unaffected by occurrence. other words. In other words, every word that occurs anywhere in the document is chosen regardless of the semantic influence of the document.

Consider using the Bag-of-words approach to present the image. To present the image, consider it as a document, that is, a set of "visual words" that do not follow each other. Since the dictionary in the image is not as easily accessible as in the text document, we must first extract the visual dictionary from the image independently of each other, which usually requires three steps: the detection of the signs, the presentation of the signs and the creation of the dictionary.. To recognize this class of objectives, we can single out the common components between the individual examples in the form of the visual dictionary. Since the SIFT method is the most common approach used to extract local invariant features in images, we can use it to extract invariant characteristic points

from images as a visual dictionary and to create word lists to present images with words from a list of words.

3.2 Approach

3.2.1 Salient Point Descriptor

The features of SIFT algorithm are.

1. SIFT features are image local characteristics that are invariant to rotation, scale scaling, and brightness changes, as well as a degree of stability to viewpoint shifts, affine transformations, and noise.

2. Distinctiveness is good and informative, allowing for quick and accurate matching in a large feature database.

3. Multiplicity; even a small number of objects can provide a huge number of SIFT feature vectors.

4. High-performance, optimized SIFT matching algorithms can meet real-time needs.

5. Scalability, which allows for easy integration with different types of feature vectors.

Factors like as the target's own state, the environment in which the scene is placed, and the imaging properties of the imaging equipment all impact the performance of image alignment/target identification tracking. Furthermore, the SIFT technique can tackle target rotation, scaling, and translation, picture affine and projection transformation, lighting effects, target occlusion, clutter scene, and noise.

The SIFT algorithm is divided into four steps as follows:

1. Maximum scale and space detection: Find places for the image at all scales. The Gaussian differentiation function identifies potential sites of interest that are constant in scale and rotation.

2. Keypoint localization: For each potential position, an exact model is fitted to determine the position and scale. The main sites are chosen because of their stability.

3. Orientation Determination: Assign one or more orientations to each keypoint location based on the local gradient orientation of the image. All subsequent actions on the image data are transformed according to the orientation, scale, and position of keypoints, ensuring invariance to these changes.

4. Key point description: The gradients around the image are measured at different scales around each key point. These gradients are converted into a representation that allows for fairly significant local shape distortions and light variations.

3.2.2 Image Matching
Step 1. Convert images to single-precision grayscale maps.

Step 2. Calculate sift frames (key points) and descriptors.

Step 3. Find closest neighboring key-point (Judgment using Euclidean distance).

Step 4. Randomly select 50 features and display them, overlaying the descriptors on top of the feature points.

Step 5. Display the SIFT pairs corresponding to the two graphs according to the best matching pair scores using RANSAC Algorithm.

3.2.3 Bag of Words
Step 1. Convert images in part (b) and Dog_2 to single-precision grayscale maps.

Step 2. Extract SIFT features from the five images.

Step 3. Apply K-means clustering to the five images to create dictionary's visual words, and K = 8.

Step 4. Use cluster centers to generate histograms of length K.

Step 5. Normalize the histograms into sum 1.

Step 6. Match Dog_2's BoW representation with Cat_1 and Dog_1, calculate the similarity of histogram.

3.3 Results
(a)
1. The SIFT is robust to image scale and rotation, affine distortion, change in 3D viewpoint, addition of noise, and change in illumination.

2. SIFT achieves its robustness to scale by detection of scale-space extrema. The initial image is repeatedly convolved with a Gaussian function to produce a set of different scale space images. Then the difference of scales, D (x, y, z), and a close approximation to the scale-normalized Laplacian of Gaussian, $\sigma^2\nabla^2 G$, is calculated using the Gaussian difference function.

SIFT achieves its robustness to rotation by assigning a consistent orientation to each keypoint based on local image properties, then the keypoint descriptor can be represented relative to this orientation.

SIFT achieves its robustness to affine distortion and change in 3D viewpoint by keypoint descriptor. First, the picture's gradient magnitude and direction are sampled around the keypoints, with the size of the keypoints used to determine the degree of image Gaussian blur. By generating a directional histogram over a 4*4 sample region, the key point descriptor allows for substantial variations in gradient location. Trilinear interpolation is used to assign values to each gradient sample and place it into surrounding bins (columns). Saving the value vectors of all directed histograms yields the descriptors. Finally, the eigenvectors are changed to reduce the influence of variations in light.

SIFT achieves its robustness to change in illumination by Reducing the effect of placing this limit of no greater than 0.2 per unit feature vector on large gradient magnitudes and then renormalizing the unit length. This means that matching large gradient magnitudes is no longer an important thing, and more emphasis is placed on the distribution of directions. The value 0.2 was obtained from experiments in which the images retained different illumination for the same 3D target.

(b)
1.

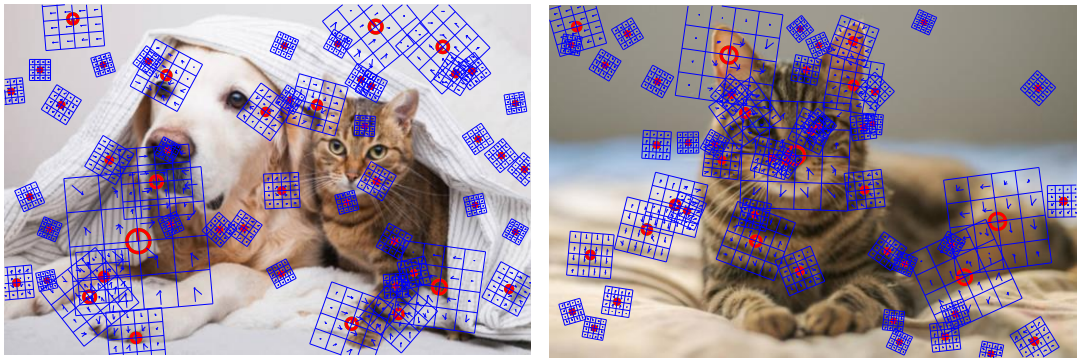


Figure 6: Key-points of the Cat_1 and Cat_Dog

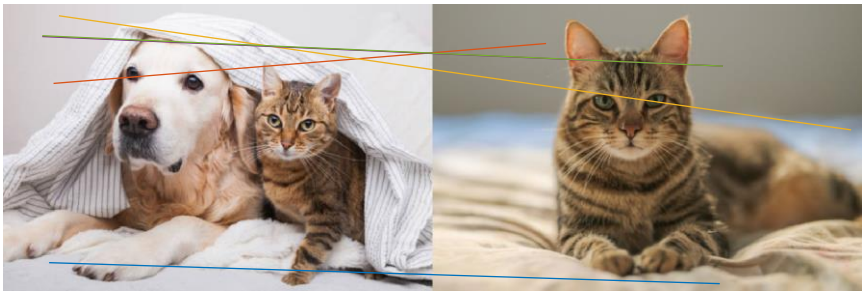


Figure 7: SIFT pairs between Cat_1 and Cat_Dog

We can see that the key point with the largest scale in Cat_1 is at the forehead of the cat, which is basically at the center, and the direction is slightly inclined to the right. the nearest neighbor feature in Cat_Dog is located at the quilt on the right side of the

dog, and the direction is also slightly inclined to the right. It is similar to the performance of the SIFT keypoint in Cat_1.

2.

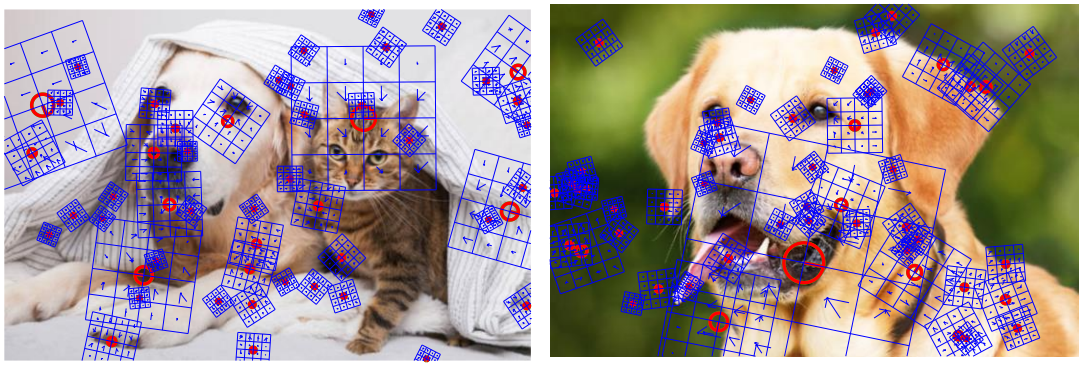


Figure 8: Key-points of the Dog_1 and Cat_Dog

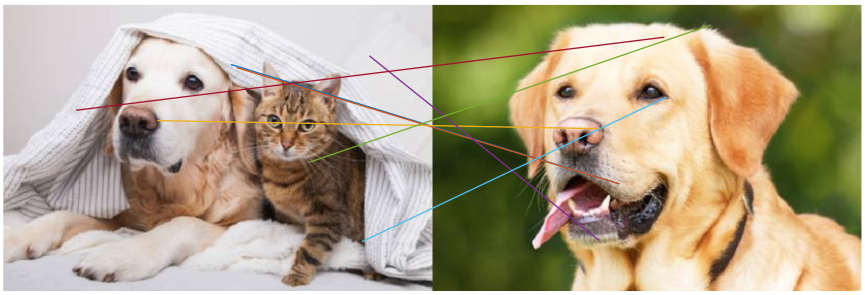


Figure 9: SIFT pairs between Dog_1 and Cat_Dog

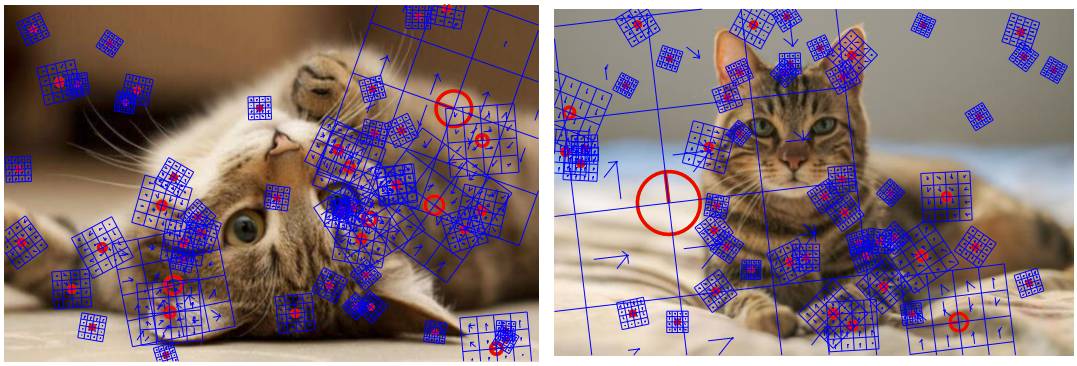


Figure 10: Key-points of the Cat_1 and Cat_2

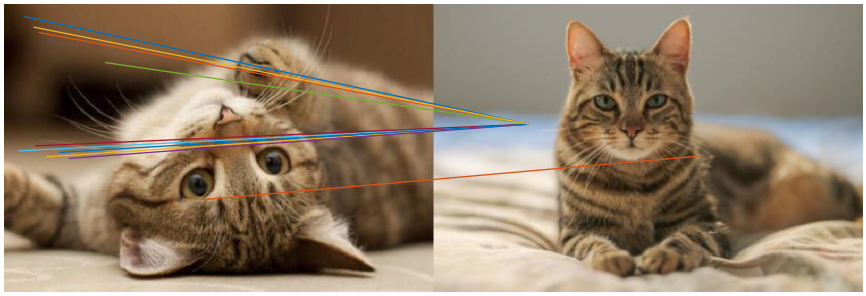


Figure 11: SIFT pairs between Cat_1 and Cat_2

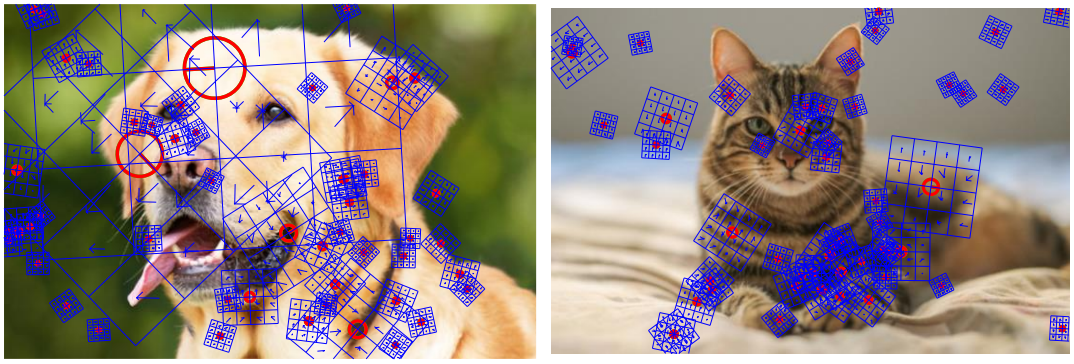


Figure 12: Key-points of the Cat_1 and Dog_1

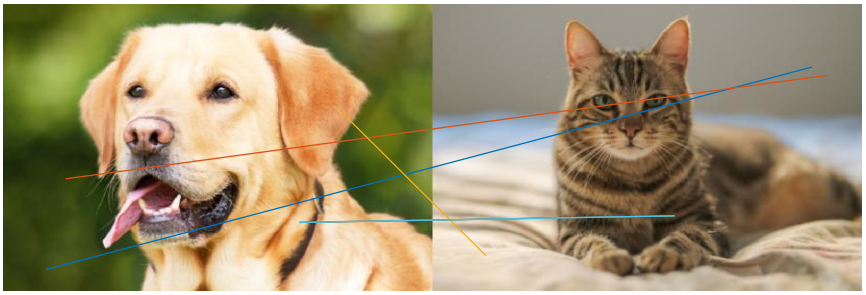


Figure 11: SIFT pairs between Cat_1 and Dog_1

I found that Dog_1 and Cat_Dog's matching result was OK, the dog's nose could be matched, Cat_1 and Cat_2's matching result was surprisingly bad, only the background could be matched. Cat_1 and Dog_1 also did not match well because cats and dogs are not the same species.

(c) Bag of Words

Figure 12: Original Images

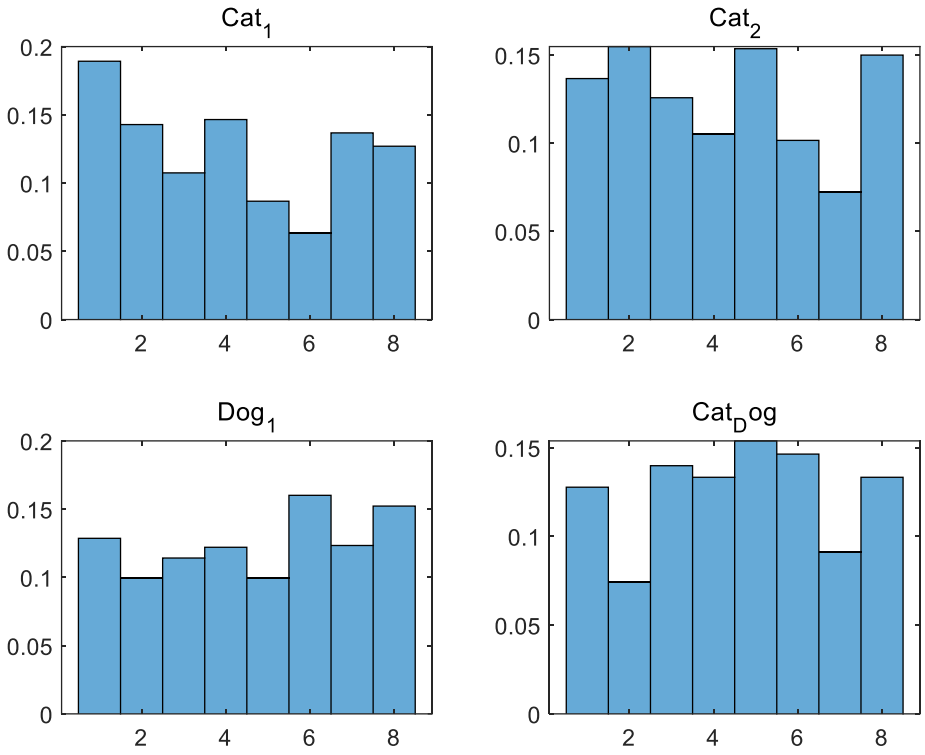


Figure 12: Histogram of codewords

The similarity index is shown below:

Matching Pairs	Cat_1 and Dog_2	Dog_1 and Dog_2
Similarity Index	0.6976	0.6843

Based on comparing the similarity indices we find that Cat_1 and Dog_2 have a higher similarity index than Dog_1 and Dog_2. This is a surprising result. We speculate that this may be related to the distance at which the object was targeted and the background of the object.

3.4 Discussion

The feature matching findings reveal that when the SIFT technique is used to compare photos from various objects or different viewpoints of the same item, the matching effect is not sufficient.

The SIFT algorithm is still very stable in terms of obtaining the image's local highest value, but it relies too heavily on the gradient direction of the pixels in the local area in the main direction finding phase, which can lead to Cardinal being found to be imprecise, and feature vector extraction and later matching rely heavily on Cardinal even if the angle of deviation is not large. Furthermore, the picture pyramid layers are not near enough to generate a scale mistake, and subsequent feature vector extraction is similarly dependent on the matching scale.

Bag of Word is easy to use, easy to build the initial word bank and add new words, and

has a high accuracy rate. The downside is that it just considers the number of occurrences of words and not their order, making it less efficient.