University of Southern California

Viterbi School of Engineering

Introduction to Digital Image Processing

2018 Spring EE569

Homework #3

Hang Dong

Student ID: 8151308667

donghang@usc.edu

March 23,2018

Problem 1: Texture Analysis and Segmentation

(a) Texture Classification

1.Abstract and Motivation

Texture analysis refers to the extraction of texture feature parameters through a certain image processing technology, therefore we obtain a quantitative or qualitative description of the texture. According to their properties, texture analysis can be divided into 4 categories: statistical analysis method, structure analysis method, signal processing method and model method. The texture feature is the statistics of the gray scale distribution function of the local property of an image. Texture analysis has an important application in the field of computer vision The Laws filter method is a typical first order method, which has great influence in the field of texture analysis.

2.Approach and Procedures

Laws filter has different size, usually $3*3 \text{ or } 5*5$ which can be conducted by one dimensional filter templates. For $5*5$ Laws filter, they are defined as follows:

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]

One example of 2-D filter:E5L5

$$\begin{bmatrix} -1 \\ -2 \\ 0 \\ 2 \\ 1 \end{bmatrix} \times [1 \quad 4 \quad 6 \quad 4 \quad 1] = \begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

Therefore we can choose E5,S5,W5 three 1-D templates to conduct nine $5*5$ Laws filters.

In order to do texture classification, the first step is extracting texture feature of different input images by applying nine Laws filter. And then we can get 9 filtered images for each input and each pixel has a 9-D feature vectors. Secondly, we average these 9-D feature vector to get the 9-D feature vector for each single image, i.e. the energy of each image. It can be expressed as the following equation:

$$\text{featurevector} = \frac{\sum F(i,j)^2}{Sizecol * Sizerow}$$

where $\mathrm{F(i,j)}$ is the corresponding energy response of every feature.

Finally we can use K-mean clustering algorithm to these texture into k parts. The K-means algorithm is to divide the samples into K clusters, which belongs to unsupervised learning. The algorithms is described as follows:

(1) Select K cluster centroid points randomly

(2) Calculate the difference between the remaining elements and K clusters centroids respectively, and group them into lowest difference cluster.

(3) According to the results of clustering, recalculate the new centroids of K clusters. The method is the arithmetic average of the respective dimensions of all the elements in the cluster.

(4) All the elements are re-clustered according to the new centroids.

(5) Repeat until the result of clustering does not change.

3.Experimental Results

(1) Feature vector

Feature num#:	EE	ES	EW	SE	SS	SW	WE	WS	WW
Image num#	378	211	71	344	141	53	67	48	19
	67	43	15	99	29	11	14	10	4
	49	25	9	208	19	7	14	8	3
	414	233	76	382	158	56	78	53	20
	56	26	9	202	20	7	15	8	3
	398	229	78	395	154	58	75	54	21
	25	15	7	246	15	7	10	7	3
	26	16	7	238	15	7	11	7	3
	50	28	9	92	21	8	13	8	3
	24	15	6	245	14	6	10	6	2
	40	22	8	213	17	7	11	7	3
	74	51	18	107	37	15	18	14	6

(2)Clustering result

```
------------Final result:-------------
Cluster1: texture2 texture9 texture12
Cluster2: texture3 texture5 texture11
Cluster3: texture1 texture4 texture6
Cluster4: texture7 texture8 texture10

Process finished with exit code 0
```

4.Discussion

Report your results and compare them with the reality (by checking the texture images by eyes). Discuss any discrepancy.

According to the clustering result its clear that texture images all have been correctly clustered to 4 clusters. Some of the features have good distinct values with each other. The images in the same cluster have similar values of their 9-means features.

(b) Texture Segmentation

1.Abstract and Motivation

Image texture segmentation is of great significance to the research in image processing, pattern recognition and computer vision filed. Texture segmentation is to divide an image which consist of different textures into several regions, in which each area has consistency or similarity. The process of texture segmentation is usually composed of two parts: feature extraction and regional consistency segmentation algorithm. The result of texture image segmentation is to assign a category symbol to each pixel of the image. It is a pixel based image processing.

2.Approach and Procedures

The first step is to get feature vector of every single pixel. Therefore, we apply the Laws filter to the input image. Here we use nine $3*3$ filters which can be conducted by E3,L3,S3 1-D filter masks.

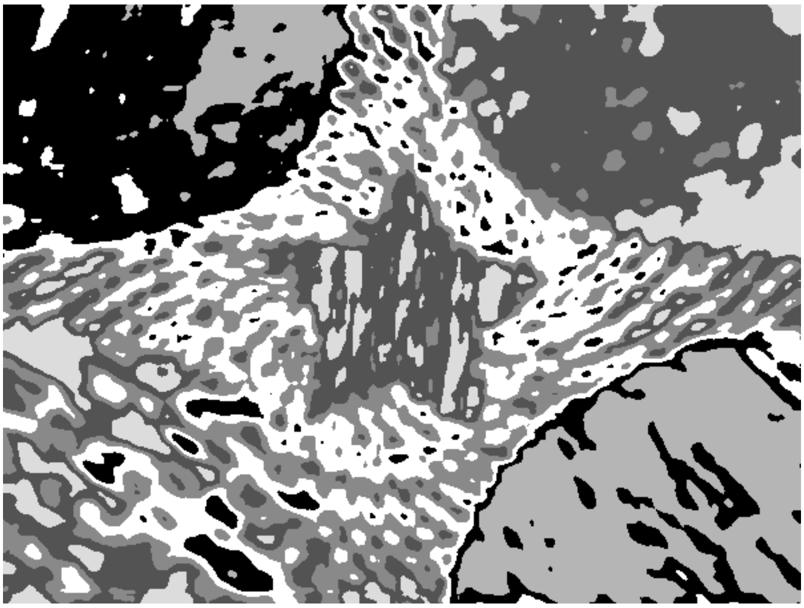
$$\mathrm{E3} = [-1 \quad 0 \quad 1]$$

$$\mathrm{L3} = [1 \quad 2 \quad 1]$$

$$\mathrm{S3} = [-1 \quad 2 \quad -1]$$

After getting 9 images of the input, we use an average window to compute feature vector for each pixel i.e. energy. Different window size leads to different results. Since the L3L3 kernel has a non-zero mean, we normalize all other features by dividing L3L3 kernel. Finally we use k-mean clustering algorithm to do segmentation for all pixels in the composite image based on the 9-D feature energy vectors. Pixels in different clusters will assign different gray-scale values for vision. Here we divide into k=6 clusters, we may assign (0, 51, 102, 153, 204, 255) to different clusters. The K-mean clustering algorithm procedure is similar, just the element is not the whole image average energy, the element becomes feature energy vector of each pixel in an image.

3.Experimental Results



4.Discussion

According to the segmentation result as above image, Its clear that the comb.raw image already be segmented into six separate parts. But its also clear the segmentation result is not very good and lots of different pixel grayscale values at each separate part. The central part looks like a star, but there are lots of different grayscales inside of it.

(c) Advanced

1.Abstract and Motivation

The results in (b) is not well, so we need improve the performance by PCA or some post-processing procedure.

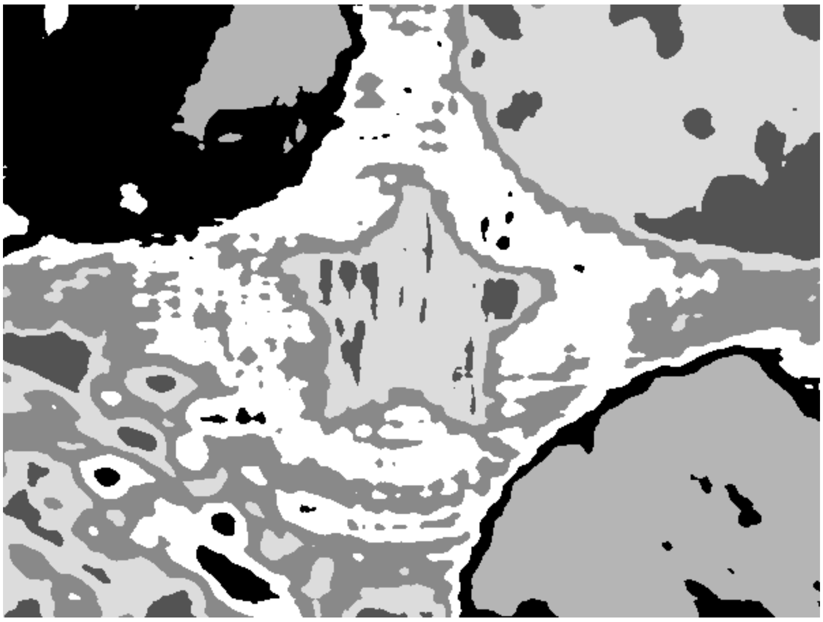
2.Approach and Procedures

I use the result image in (b) as the input image to handle. I select a larger window size to scan the image, and assign the most frequently gray-scale value the current center pixel which can remove some wrongly clustered pixels to get more smooth results.

Secondly we can apply PCA select some more salient features to do segmentation task which can reduce running time.

3.Experimental Results

(1) Window_Size=40



(2) Window_Size=50

4.Discussion

By comparing the results of my algorithm and the result of original segmentation its obvious the results get better after my algorithm operated. When window size equals to 50, the segmentation is clear and good. That due to larger window size can identify more wrong pixels' value. However, by comparing the results of window size=40 and window size=50, the larger window size also will result in edge fuzziness.

Problem 2: Edge Detection

(a) Basic Edge Detector

1.Abstract and Motivation

The gray scale level of different images is different, and there is a clear edge at the boundary. This feature can be used to divide the image. The purpose of edge detection is to find the set of pixels in the image which has intense brightness changes, the output is often represented by the contour. When there is a strong gray-scale change between two pixels, we judge there is an edge between them. The derivative edge operator, also called gradient edge operator, is used to detect edges. It uses the step change on the edge of the image, that is, the image gradient gratitude has a local maximal value on the edge pixels. In practical use, we usually use finite difference to replace gradient value.

2.Approach and Procedures

(1) Sobel Detector

The Sobel operator uses two 3*3 matrices filter i.e. vertical and horizontal filter to get a gradient image. Two filters are as follows:

$$I_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \text{ and } I_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

After defining the filters, we calculate the gradient images in the horizontal and vertical direction respectively by do convolution between a filter and input image, i.e. $G_x = I_x * A$ and $G_y = I_y * A$. The gradient value of each pixel can be calculated by equation, $\mathrm{G} = \sqrt{G_x^2 + G_y^2}$. After getting the gradient image, what we need is to calculation threshold. We can use the histogram method. We can choose the top 10% or top 5% gradient values as the edge pixels and set as gray-scale value 255 and the others are gray-scale value 0. The binary image is the desired edge map. We may choose different threshold to get a better visual effect.

(2) Zero crossing Detector

The edge pixel position is in the local maximum value pixels of first order derivative of input image, it also means that the second order derivative of the same point is zero. The second order derivative operator is a method based on this character to perform edge detection. The second order derivative of the image can be represented by the Laplacian operator: $\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$. The corresponding convolutional kernel is $\mathrm{m} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$. But this method is very sensitive to noise, therefore we need de-noise at first to reduce the noise effect on the edge, for example apply Gaussian low pass filter. The procedure of applying Gaussian filter to the image and then do second order derivative to the filtered image can be converted into do second order derivative to the Gaussian function and then apply this result to input image. It can be described as:

$$\nabla^2[G(x,y) * f(x,y)] = [\nabla^2 G(x,y)] * f(x,y)$$

Where $f(x,y)$ is the input image value. It is called LoG operator (Laplacian of Gaussian).

The LOG kernel I used is 5*5 as follow:

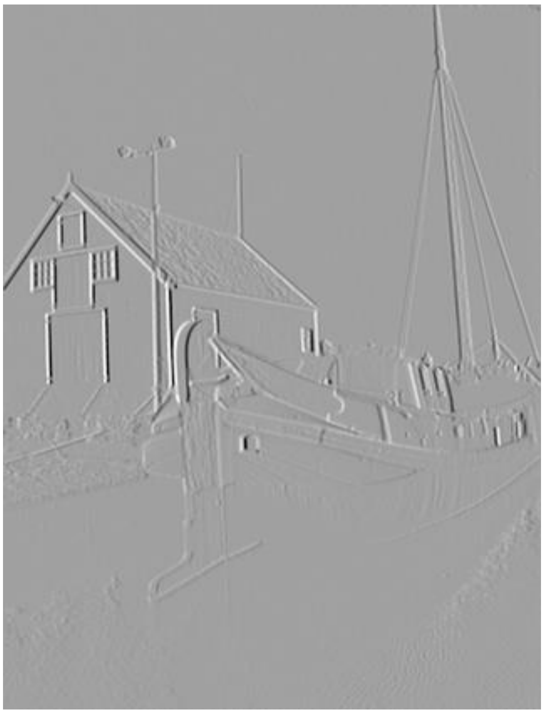
```
{{0,0,-1,0,0},{0,-1,-2,-1,0},{-1,-2,16,-2,-1},{0,-1,-2,-1,0},{0,0,-1,0,0}}
```

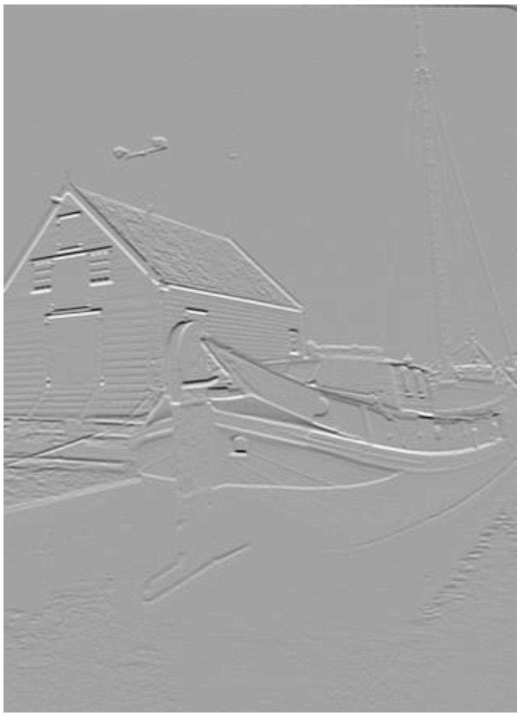
3.Experimental Results

(1) Sobel

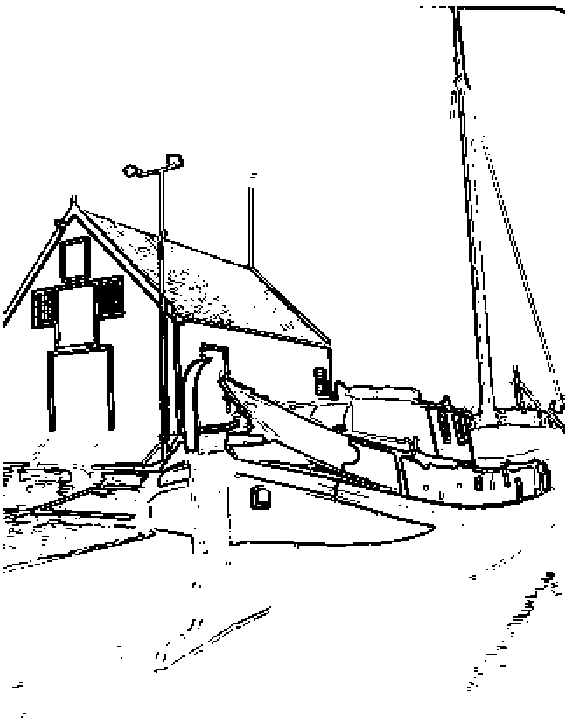
Boat:

X Direction: **Y Direction:**





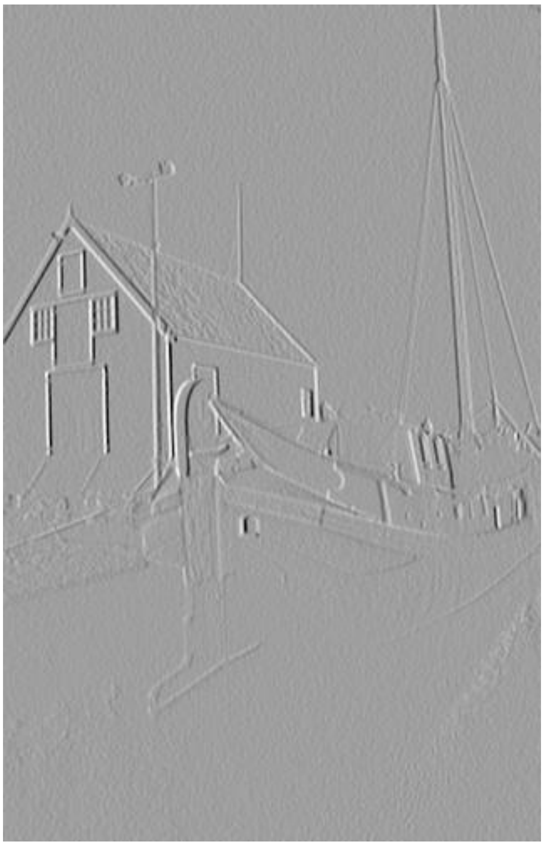
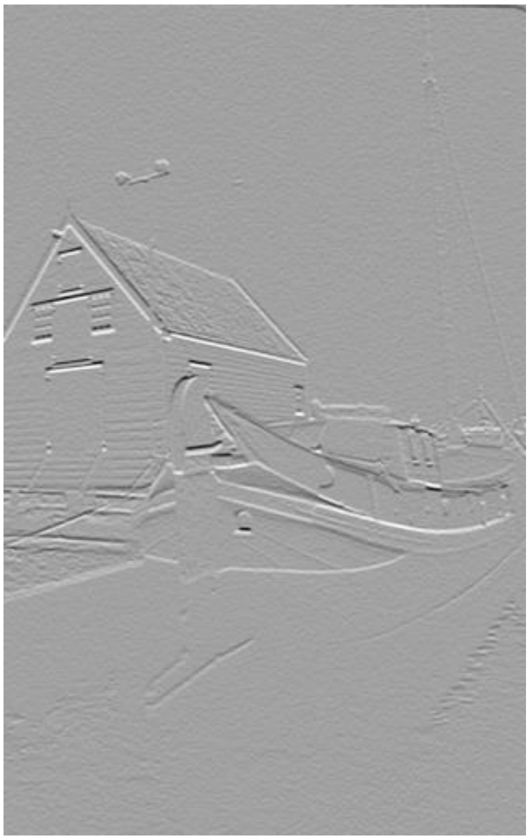
Edge:



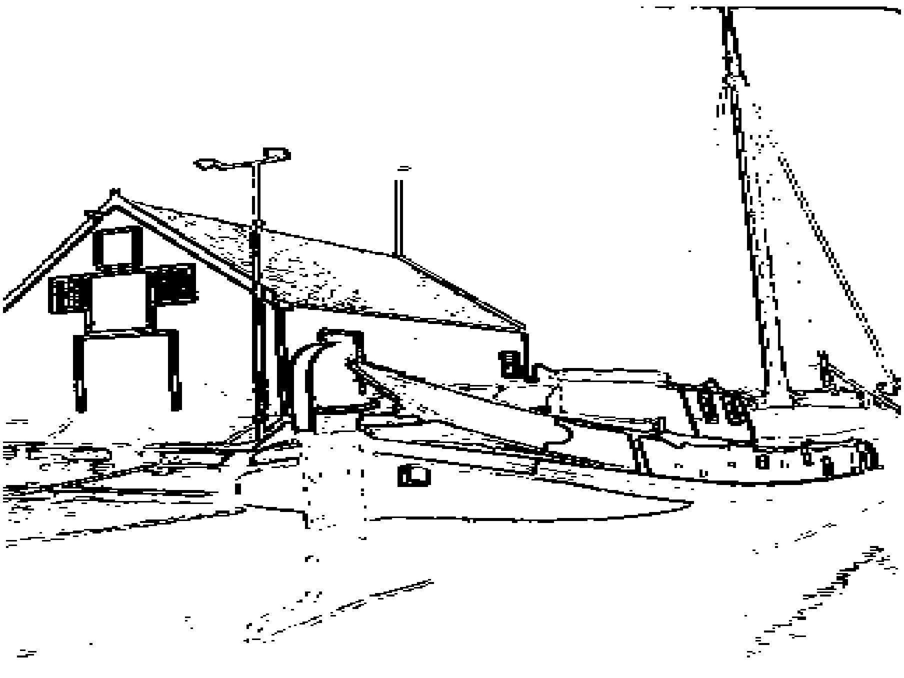
Noisy Boat:

X direction:

Y direction:

Edge:

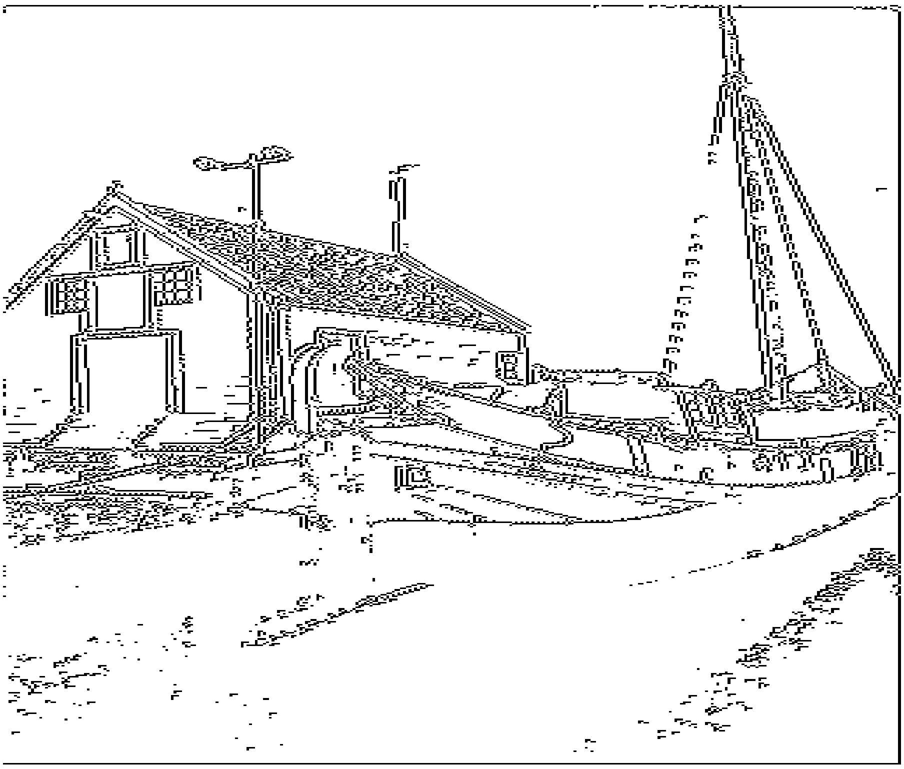


(2) LOG

Boat:

Normal:

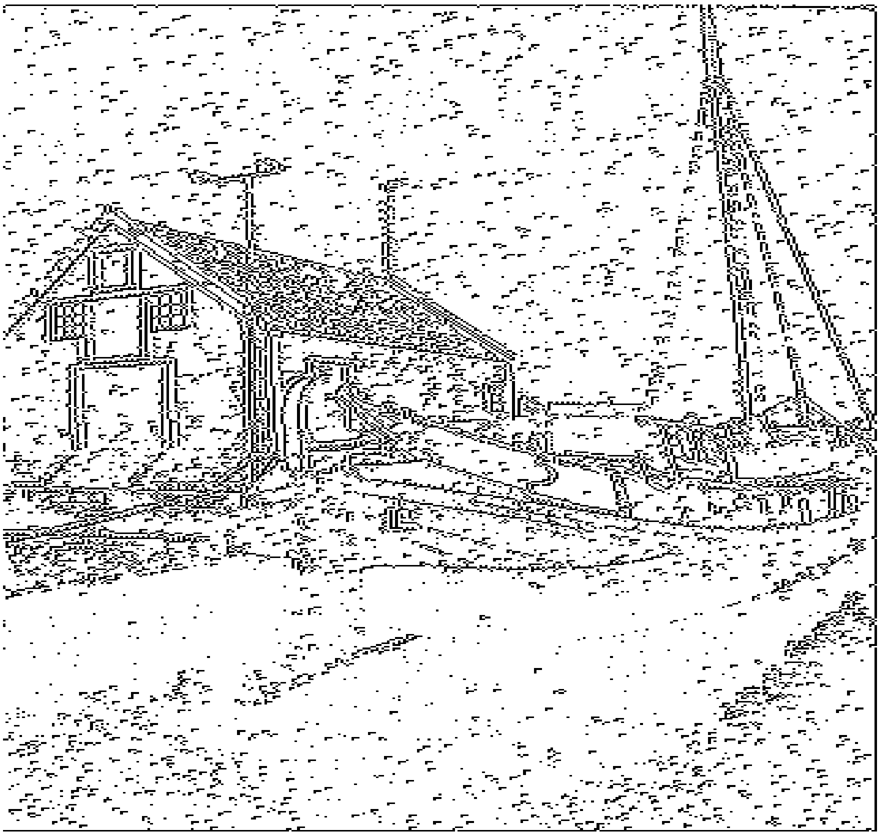
Edge:



Noisy Boat:

Normal:

Edge:



4.Discussion

b. Calculate knee points of LOG

#1 step, Converting the color image to grayscale image, which range 0-255.
#2 step, Set up the threshold **T** according to the top 10% grayscale pixels. Then use this knee point threshold for both **T** and **–T**.
#3 step, then operate the grayscale image based on **T** and **–T**. Set the pixel value to 1 if pixel value greater than **T**; Set pixel value to -1 if pixel value less than -**T**; Set pixel value to 0 for other cases;

b. Observation

By comparing the results of Sobel method and LOG/cross zero method, I can notice that both methods can get the edge of items in image. What is more, the Sobel method has better result than LOG. I think the main reason of this difference due to LOG applied Gaussian filter firstly, so it can eliminate noise that density change less than σ. However my LOG result is bad due to my σ is not good.

(b) Structured Edge

1.Abstract and Motivation

The traditional edge detection methods have the disadvantage of manual adjustment of threshold parameters, narrow application scope and so on. Therefore, we can apply machine learning theory to image edge detection, change the problem into a typical classification problem in machine learning i.e. divide the pixels in the image into two categories: edge points and non-edge points, which can improve edge detection performance. We can use objectively annotated images as training data such as BSD500 dataset.

2.Approach and Procedures

Decision tree is a kind of supervised learning algorithm. An approach based on tree is to divide the feature space into a series of categories, and then each category has a simple model (like a constant) .

Random forest is to establish a forest in a random way. There are many decision trees in the forest and there is no correlation between every decision tree in random forest. When a new sample inputs, every decision tree in the forest will have separately decision procedure to see which category the sample belongs to and then the final predicted results is a category with more ballots by voting.

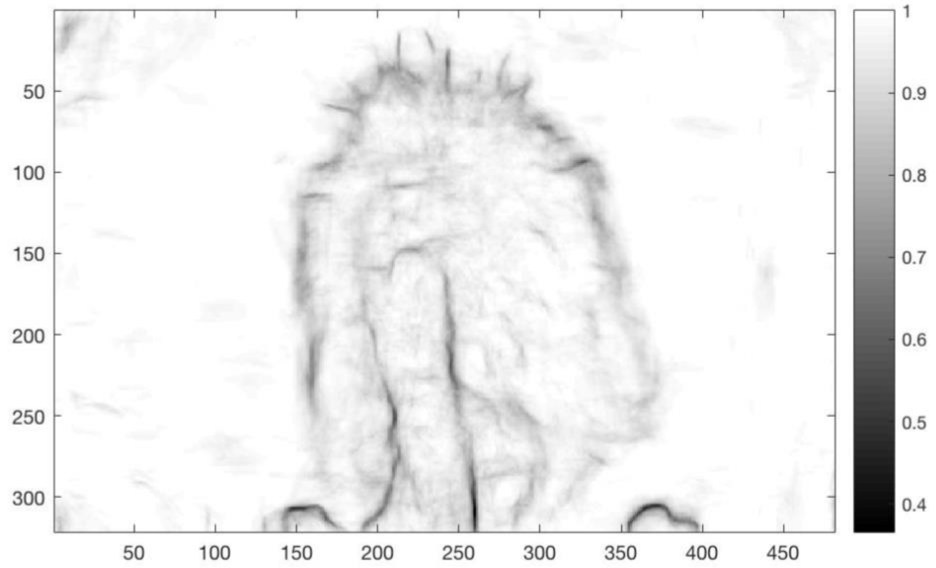
The random forest algorithm predicts each pixel in the input image separately, but ignores the connection between center and its adjacent pixels. The structured random forest learning algorithm replaces independent pixels by image patch and super pixels. This kind of method uses part of the context information. And the predictive function maps the input space to a structured label space in structured prediction.

3.Experimental Results

(1) Animal

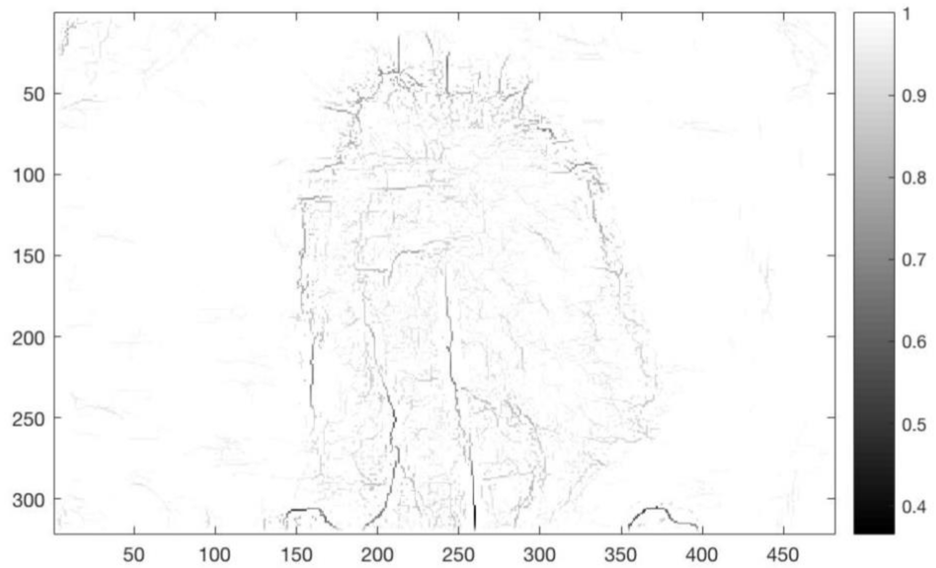
Parameters:

multiscale=1; sharpen=0; nTreesEval=1; nThreads=2; nms=0



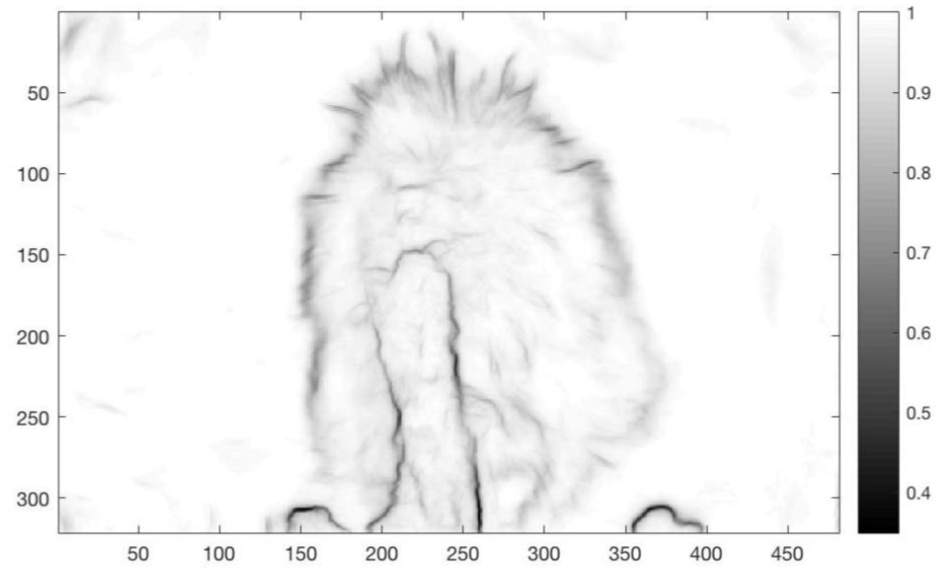
Parameters:

multiscale=1; sharpen=0; nTreesEval=1; nThreads=2; nms=1



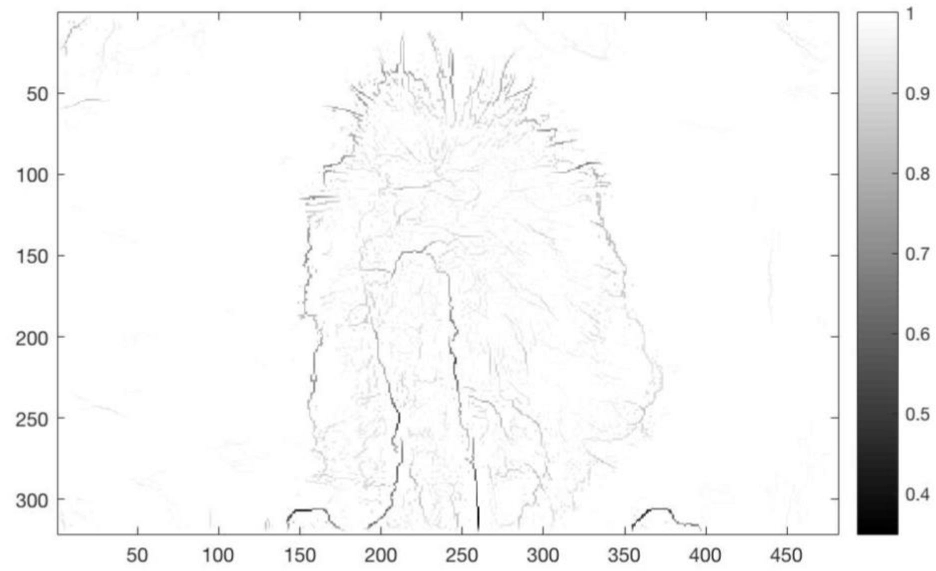
Parameters:

multiscale=3; sharpen=1; nTreesEval=5; nThreads=7; nms=0



Parameters:

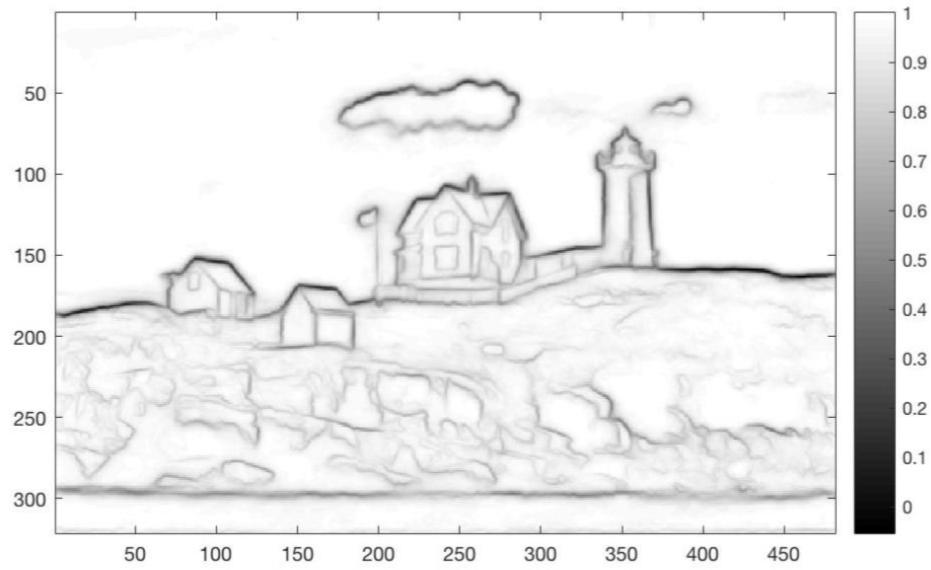
multiscale=3; sharpen=1; nTreesEval=5; nThreads=7; nms=1



(2) House

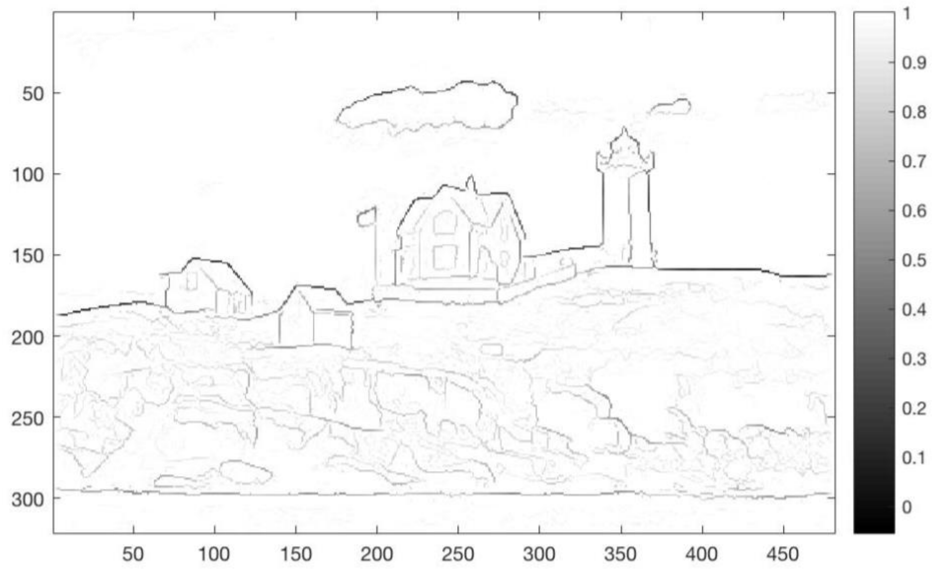
Parameters:

multiscale=2; sharpen=4; nTreesEval=3; nThreads=6; nms=0



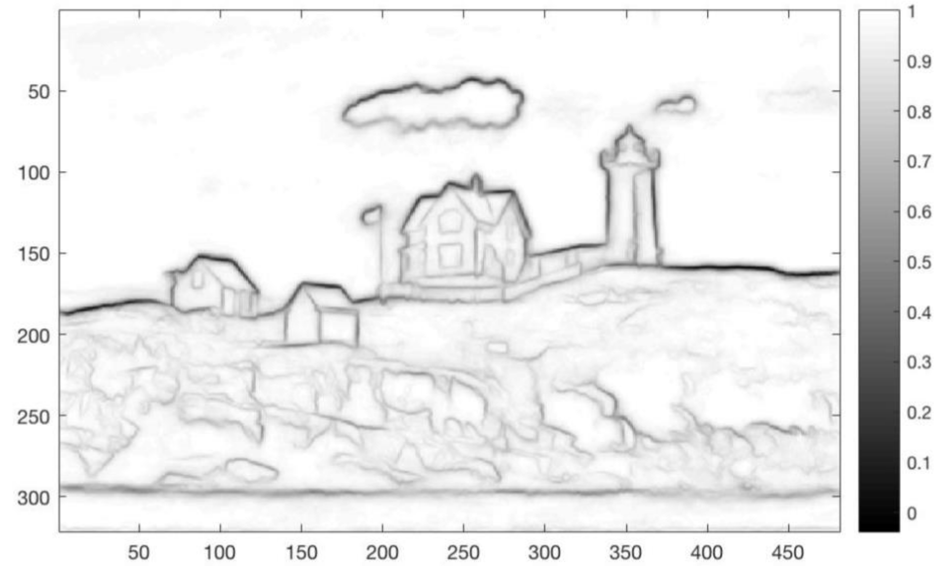
Parameters:

multiscale=2; sharpen=4; nTreesEval=3; nThreads=6; nms=1



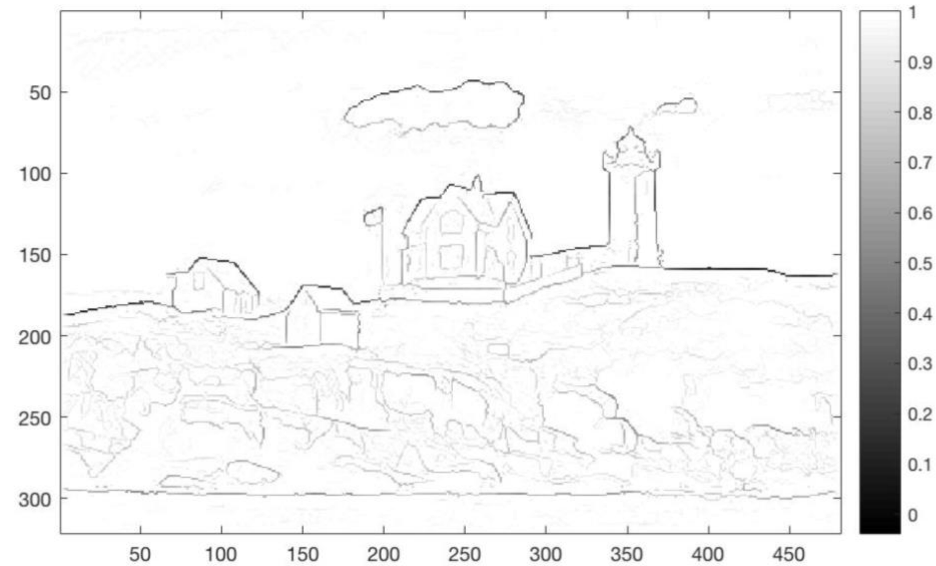
Parameters:

multiscale=5; sharpen=4; nTreesEval=2; nThreads=7; nms=0



Parameters:

multiscale=5; sharpen=4; nTreesEval=2; nThreads=7; nms=1

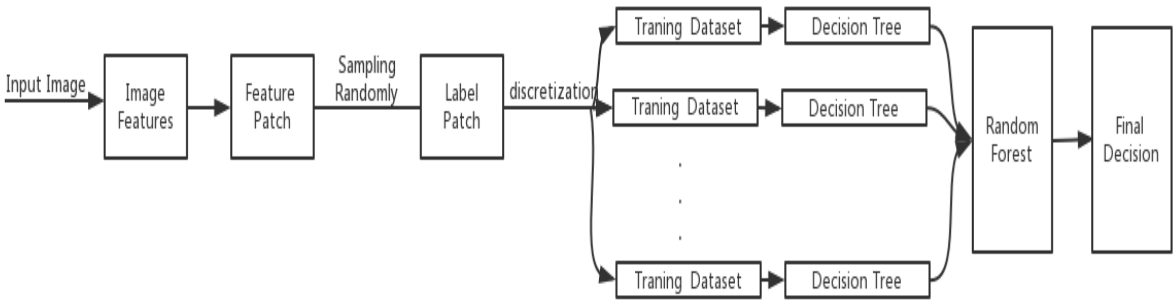


4.Discussion

(1) SE detection algorithm and flow chart

The structured RF algorithm extend random decision forest to a structured output space Y based on input image patch. In structured RF, the information gain is hard to calculate, therefore we establish the mapping relationship between structured labels and discrete labels and then we can use the traditional RF training program to calculate information gain. In order to measure the difference degree among labels, the structured label at each node need to be change to another space where the distance is calculated easily. So the mapping relation is firstly changing structured label to binary vector and then obtaining discrete labels by K-mean clustering or PCA.

Applying the structured RF algorithm to image edge detection, we need predict a 16*16 segment mask from input 32*32 image patch by 13 channels information.



(2) Decision tree construction and the principle of the RF classifier

The classification decision tree model is a tree structure that describe sample classifications. Each non-leaf node represents a test on a characteristic attribute, Each branch represents the output of this characteristic attribute, and each leaf node stores a category. The process of decision tree is to test the corresponding characteristic attributes from the root nodes, and select output branches according to their values until the leaf nodes are reached, and the category of leaf node is the output decision result. The main idea of construction decision tree is to select feature by applying information entropy rule at each node recursively. The algorithm can be described as following:

(1) Calculate the information entropy for each feature in the current dataset.

(2) Select the feature with the maximum information entropy as the decision feature for the current node

(3) Divide samples into different sub-nodes based on different types of current decision feature

(4) Do recursion on the sub-nodes until all the features are divided

The entropy of a set of samples can be calculated by:

$$\mathrm{H}(D) = -\sum_{i} p_i \log(p_i)$$

Where p_i represents the occurrence probability of a category in the whole sample set.

The information entropy of feature Y can be defined as:

$$\mathrm{Gain(Y)} = \mathrm{H}(D) - \sum_{J} \frac{|D_j|}{|D|} \mathrm{H}(D_j)$$

Where D_j is the entropy of classified subset.

A random forest is a method to discriminate and classify dataset using multiple independent decision trees. For each tree, we randomly and back ground select some samples (bootstrap sampling) as the current tree's training data. RF depends on the vote selection of each single decision tree result to determine the final classification results.

(3) Compare results

By watching results, the structure edge detection method has a more accurate result and better visual effect. Because the machine learning method learn edge rules by training and testing data which is more universal.

(c) Performance Evaluation

1.Abstract and Motivation

We can do performance evaluation using F-test method to judge the accuracy of our edge detection. The true edge map can use BSD-500 data set.

2.Approach and Procedures

There are four types edge pixels, i.e. true positive, true negative, false positive, false negative.

True positive are edge pixels that are both in predicted edge map and ground truth map. True negative are edge pixels that are neither in predicted edge map nor in ground truth map. False positive are edge pixels that are in predicted edge map but not in ground truth map. False negative are missed edge pixels that are not in predicted edge map but in ground truth map. And then we can define the concept of precision and recall.

Precision: $\mathrm{P} = \frac{\#True\ Positive}{\#True\ Positive + \#False\ Positive}$

Recall: $\mathrm{R} = \frac{\#True\ Positive}{\#True\ Positive + \#False\ Negative}$

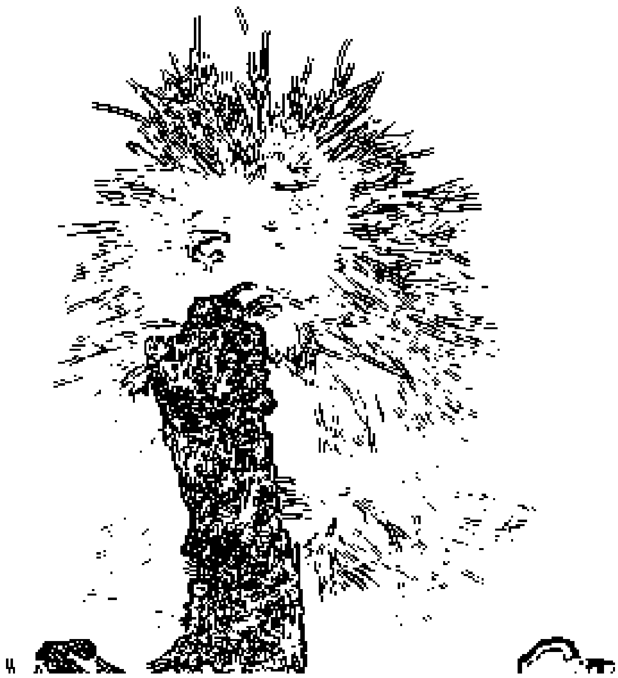
We can use F-measure to represent the performance of the edge detection algorithm.

$$F = 2 * \frac{P * R}{P + R}$$

3.Experimental Results

(1) Animal:

Sobel:



Threshold=0.18

F=0.3067

LOG:

Threshold=0.38

F=0.2793

SE:

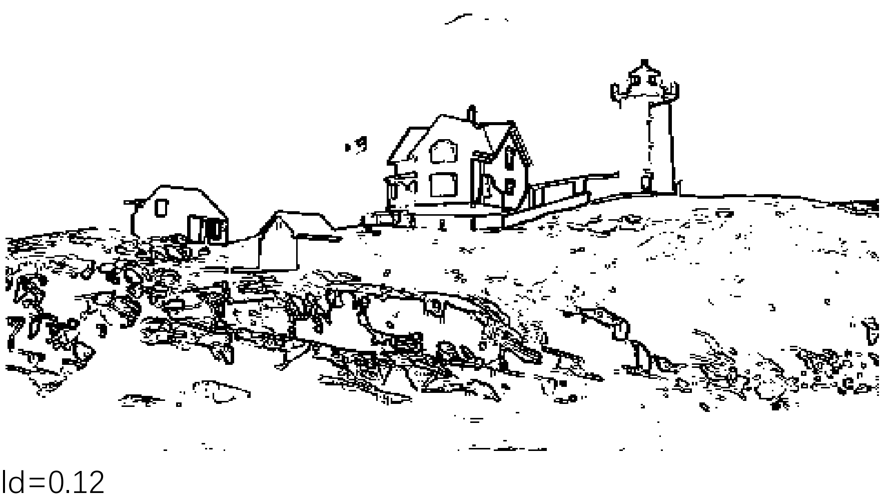
Parameters: multiscale=1; sharpen=2; nTreesEval=4; nThreads=4; nms=1

Threshold=0.13

F=0.7352

(2)House:

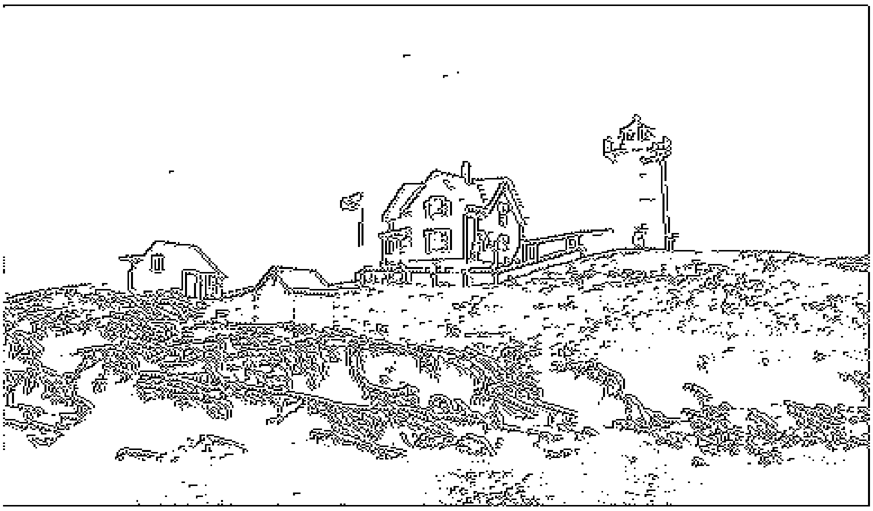
Sobel:



Threshold=0.12

F=0.4651

LOG:



Threshold=0.33

F=0.3878

SE:

Parameters: multiscale=1; sharpen=2; nTreesEval=4; nThreads=4; nms=1

Threshold=0.24

F=0.8071

Sobel: threshold: 90%

	Animal			House		
	Recall	**Precision**	**F-value**	**Recall**	**Precision**	**F-value**
G1	0.2442	0.3442		0.3928	0.49944	
G2	0.2598	0.3234		0.4891	0.4237	
G3	0.2657	0.3487		0.4879	0.4391	
G4	0.2665	0.3321		0.4783	0.4626	
G5	0.2497	02996		0.4271	0.4851	
Mean	0.2304	0.3545	0.3067	0.4771	0.4671	0.4651

LoG: $\boldsymbol{\sigma = 2}$

	Animal			House		
	Recall	**Precision**	**F-value**	**Recall**	**Precision**	**F-value**
G1	0.2541	0.3001		0.3632	0.4126	
G2	0.2043	0.3214		0.3467	0.4367	

G3	0.2984	0.2543		0.3842	03851	
G4	0.2653	0.2982		0.3821	0.4157	
G5	0.2698	0.2865		0.3563	0.4228	
Mean	0.2622	0.2987	0.2793	0.3719	0.4051	0.3878

SE: With NMS

	Animal			**House**		
	Recall	**Precision**	**F-value**	**Recall**	**Precision**	**F-value**
G1	0.7132	0.7452		0.7534	0.8482	
G2	0.7831	0.6831		0.8032	0.8087	
G3	0.6954	07986		0.7124	0.8963	
G4	0.7538	0.6952		0.7348	0.8687	
G5	0.7425	0.7345		0.7936	0.7995	
Mean	0.7277	0.7428	0.7352	0.7752	0.8417	0.8071

4.Discussion

(1) The Image House should have a higher F-measure. Because it has a distinct boundary and less messy which makes it easy to do edge detection. Therefore the precision and recall rate will more higher.

(2) Precision and recall are two values widely used in the field of classification, which are used to evaluate the quality of the results. Precision calculates that the ratio of all "correctly detected item (TP") " number and all" actually detected (TP+FP) ". Recall calculates that the ratio of the number of all "correctly detected item (TP") "and the number of all" item (TP+FN) that should be detected ". A higher F measurement means a better performance. And either higher precision or recall cannot result in a good F. And we can prove that F measure reaches the maximum when precision is equal to recall if the sum of precision and recall is a constant. If $\mathrm{P}+\mathrm{R}=\mathrm{T}$ where T is a constant. And then $\mathrm{F}=\frac{2}{T}*P*(T-P)$. When $\mathrm{P}=\frac{\mathrm{T}}{2}$, which means $\mathrm{P}=\mathrm{R}$, F reaches maximum.

Problem 3: Salient Point Descriptors and Image Matching

(a) Extraction and Description of Salient Points

1.Abstract and Motivation

Feature extraction is a concept in computer vision and image processing. It refers to extract image information by computer and determine whether the current pixel belongs to the image feature. The result of feature extraction is to divide pixels on the image into different subsets. These subsets often belong to isolated points, continuous curves or continuous regions. The SIFT(Scale-invariant Feature Transform) and SURF(Speeded Up Robust Features) algorithm has been considered as the most effective and the most widely used feature extraction algorithm. SURF is the improved version of SIFT.

2.Approach and Procedures

SIFT:

The algorithm is a method to extract image feature based on descriptor and then do image matching. The procedure can be described as following:

(1) Simulate multi-scale features of image data by constructing Gaussian Pyramid using different Gaussian kernel.

(2) Use LoG (Laplacian of Gaussian) taking pace of DoG (Difference of Gaussian)to find the key point which is the local extremum value of surrounding 26 pixels.

(3) Remove the pixels which have unsymmetrical local curvature of the DoG, i.e. key points of low contrast and unstable edge response.

(4) Assign the direction parameter to each key point based on its surrounding points' gradient direction distribution character.

(5) Generate 128 dimensional descriptor of each key point.

SURF:

The most important feature of the SURF algorithm is the use of the Haar feature and the concept of integral image, which greatly accelerates the running time of the program.

(1) Detect area where have big color and gray-scale level difference with its surrounding.

(2) Generate a Hessian matrix to build Gaussian Pyramid of input image

(3) First determination of the position of key points (feature points) by using non maximum suppression(NMS).

(4) Assign direction to key points.

(5) Generate feature descriptor for SURF feature points.

3.Experimental Results

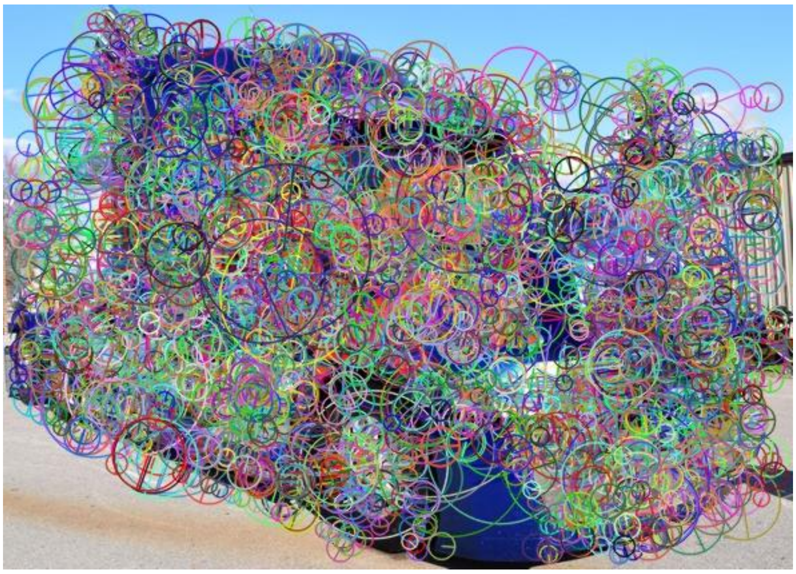
(1) SIFT

Optimus prime truck

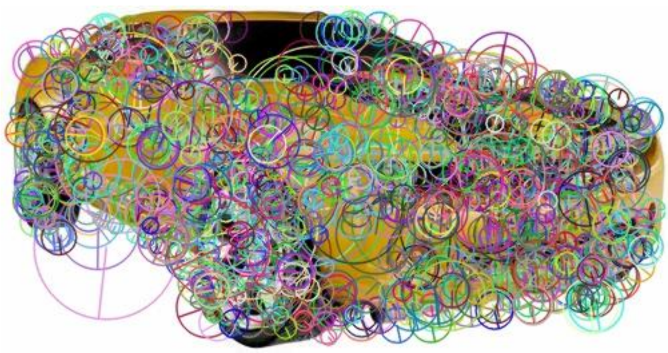
Bumblebee car:

(2) SURF

Optimus prime truck



Bumblebee car:



4.Discussion

Compare their results in terms of performance and efficiency according to their strength and weakness.

According to the result of SIFT and SURF, its obvious that SURF will find more key points than SIMT generally. But the key points of SIFT can changed by change the parameters, thresholds and contrast layers. To be specifically, when threshold gets larger the key points will be fewer. When layers gets larger the key points will be more. The key points that SIFT found is more precise than SURF, but SURF can work out the key points quickly than SIFT.

(b) Image Matching

1.Abstract and Motivation

After obtaining features of different image, we can do image matching between features.

2.Approach and Procedures

SIFT:

The matching of feature points is achieved by calculating the Euclidean distance of the two sets 128 dimensional feature points. The smaller the Euclidean distance is, the higher the similarity is. When the Euclidean distance is less than the threshold, it can be judged to be a successful match.

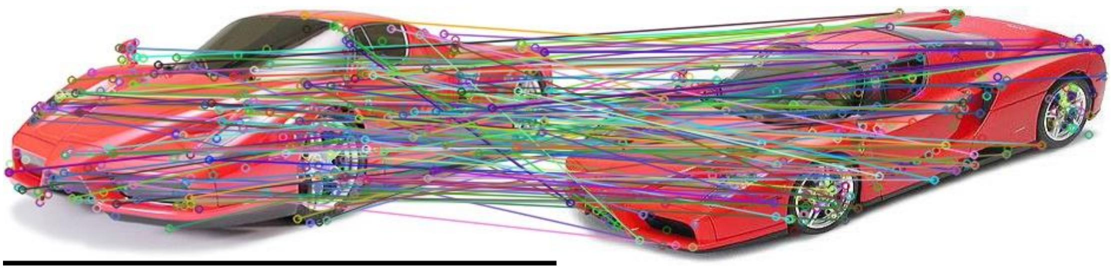
SURF:

In order to accelerate matching speed, SURF adds a new element to the feature vector, i.e. the Laplacian response sign of the feature point. When we detect key feature points, the positive and negative signs of the Hessian matrix's trace are recorded as a variable in the feature vector. In the feature matching, the computation time can be saved, because only the feature points with the same positive and negative sign can be matched, and no similarity calculation is performed for the feature points with different signs.

3.Experimental Results

(1) SIFT

Ferrari _1 and Ferrari _2

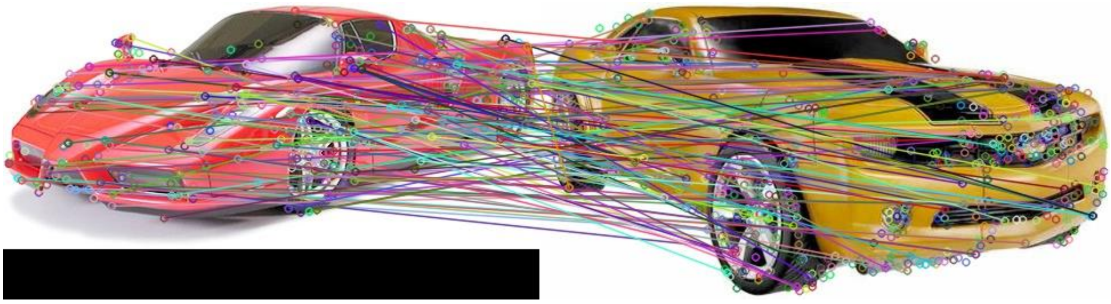


match points:383; good match points:148;

Ferrari _1 and Optimus prime truck

match points:383; good match points:98;

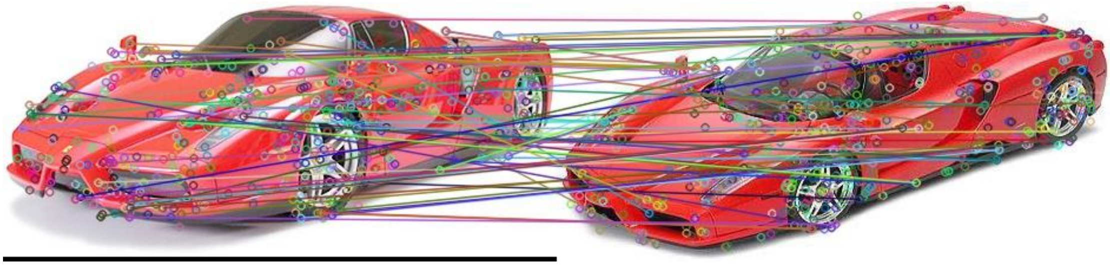
Ferrari _1 and Bumblebee car



match points:383; good match points:126;

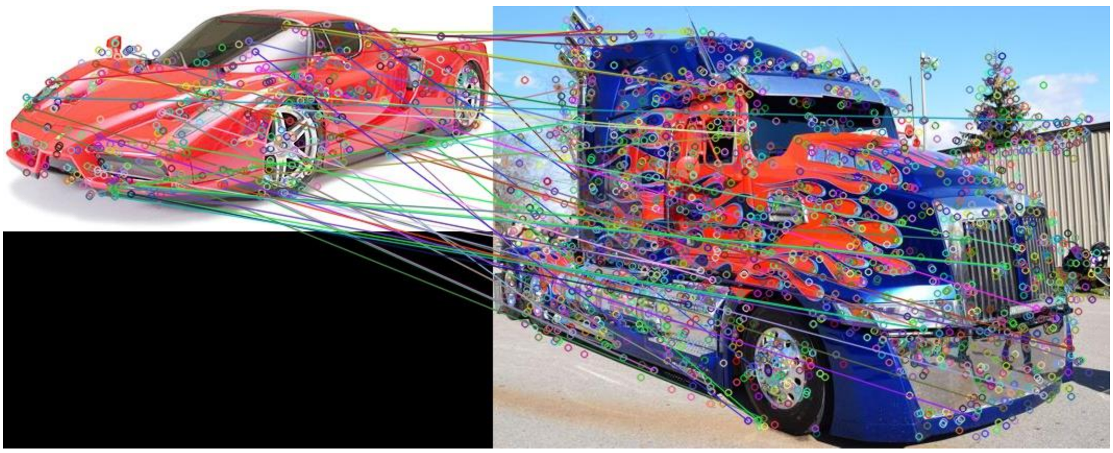
(2) SURF

Ferrari _1 and Ferrari _2



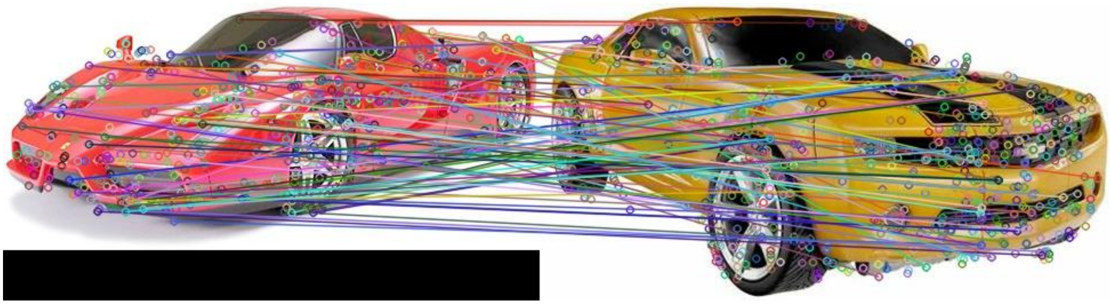
match points:573; good match points:66;

Ferrari _1 and Optimus prime truck



match points:573; good match points:52;

Ferrari _1 and Bumblebee car



[redacted]

match points:573; good match points:85;

4.Discussion

According to the matching result, it can be seen that SURF has more good matching points than SIFT. when match Ferrari1.raw and Ferrari2.raw, there will be more match points than other pairs, that means these two images are most similar. However, SIFT is worse than SURF due to it wrongly pairs some points. One obvious thing is that in the image of Optimus we can see lots of key points are matched wrongly to the tree near the Optimus. What is more, lots of key points wrongly matched to Ferrari's front window with other parts of Optimus.

(c) Bag of Words

1.Abstract and Motivation

In a huge collection of documents, every document can be represented as a N dimensional vector using Bag-of-words model, so that we can use computer to complete the classification of massive documents. In the image processing, we can consider an input image as a document with some 'feature words'.

2.Approach and Procedures

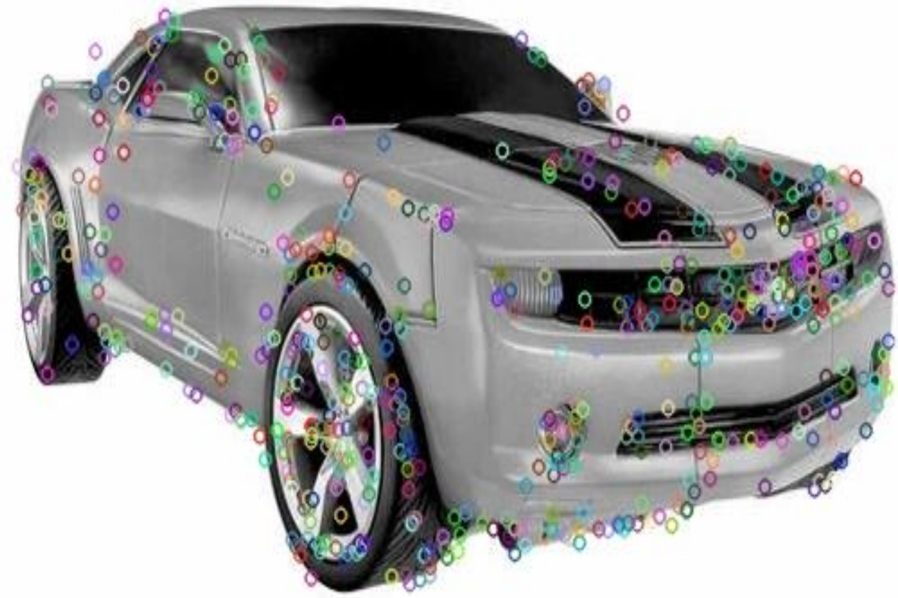
At first, we need construct word dictionary, which usually takes three steps: feature detection, feature representation and the generation of bag of word. So we can use the SIFT algorithm to extract the invariant feature points from the input image as a visual vocabulary. And then we use the K-mean clustering algorithm to construct a word dictionary based on similarity measurement among samples. Therefore, different input images can be represented by 'words'. Finally, the image can be represented by a N-dimensional vector by counting each words appearance histogram which can be a criterion of image similarity matching.

3.Experimental Results

(1)SIFT Feature:

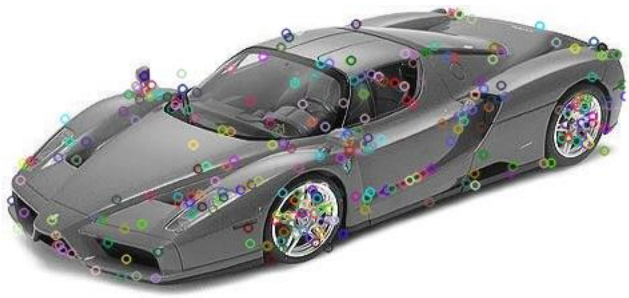
Optimus prime truck

Bumblebee car

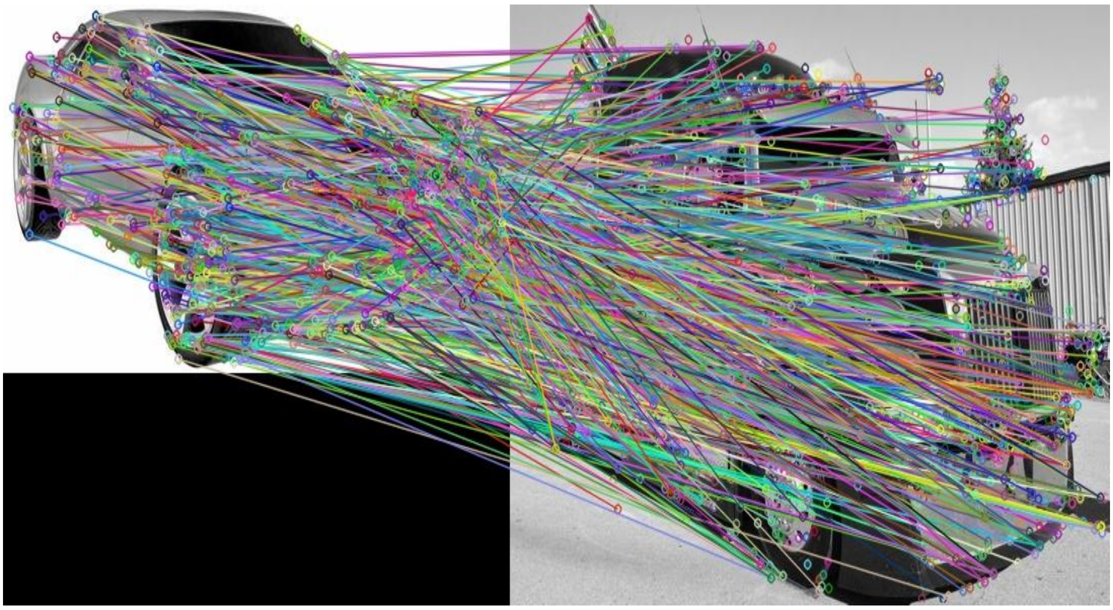


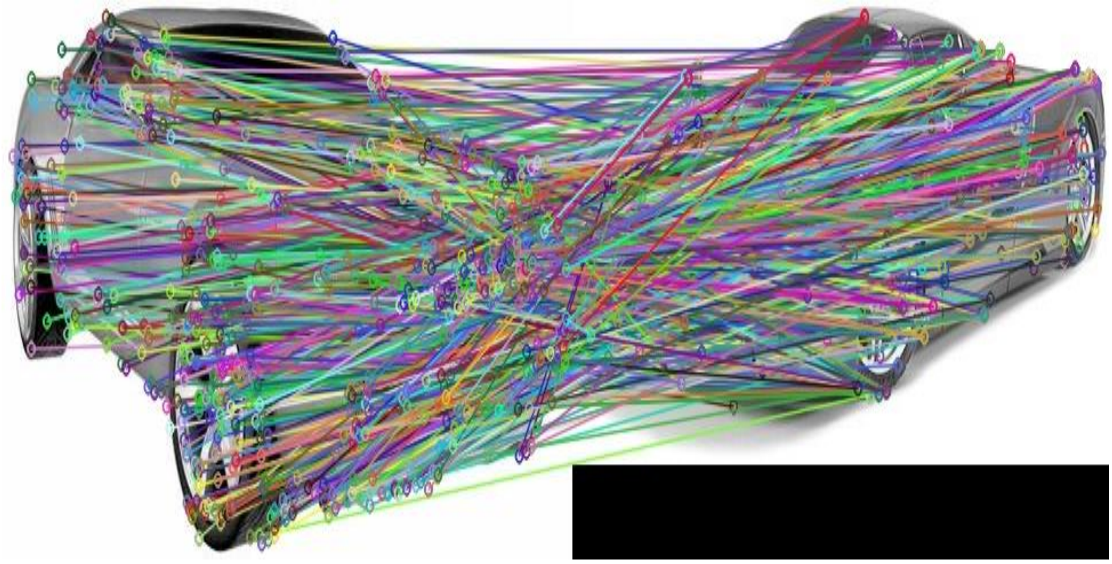
Ferrari_1

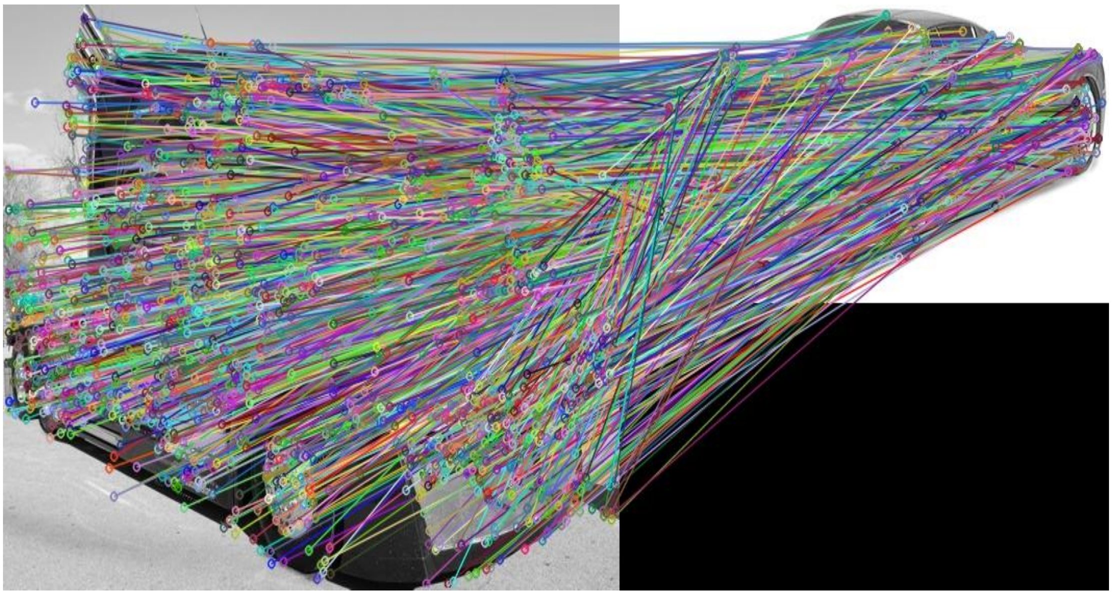
Ferrari_2



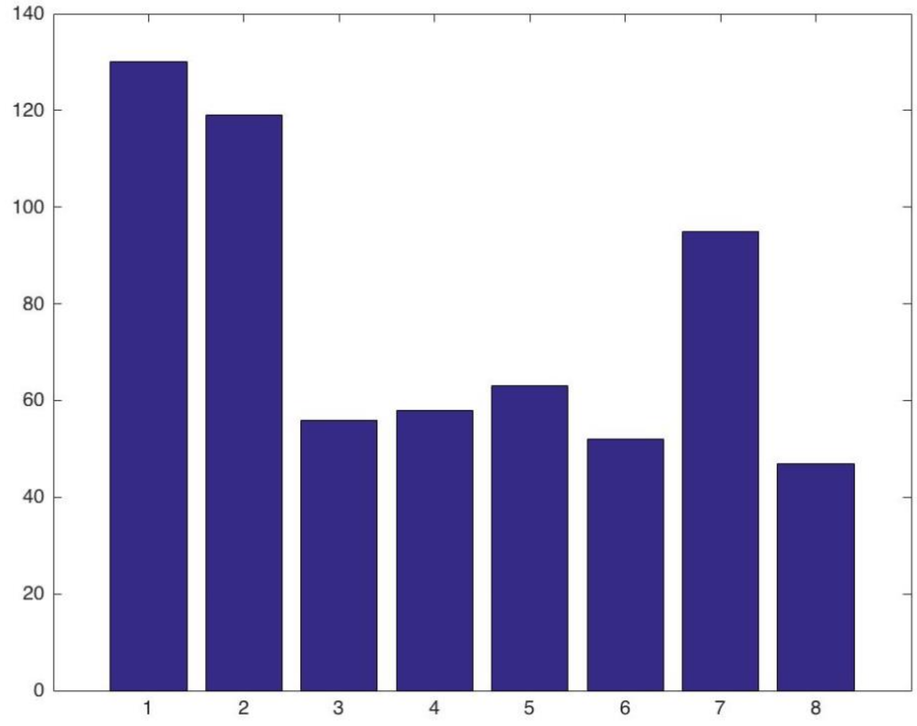
(2) Codebook



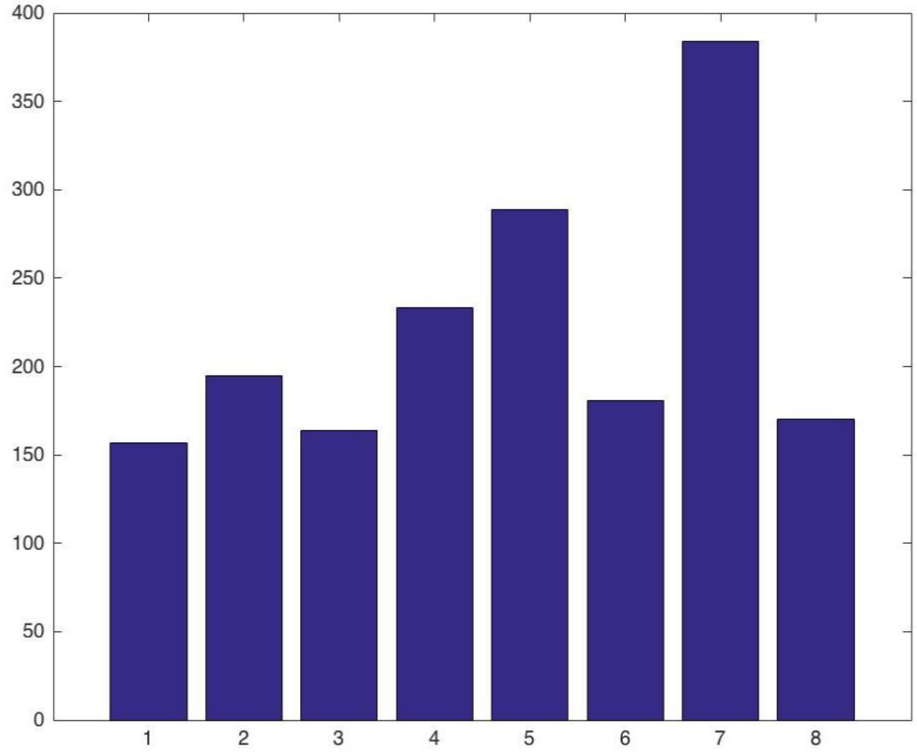




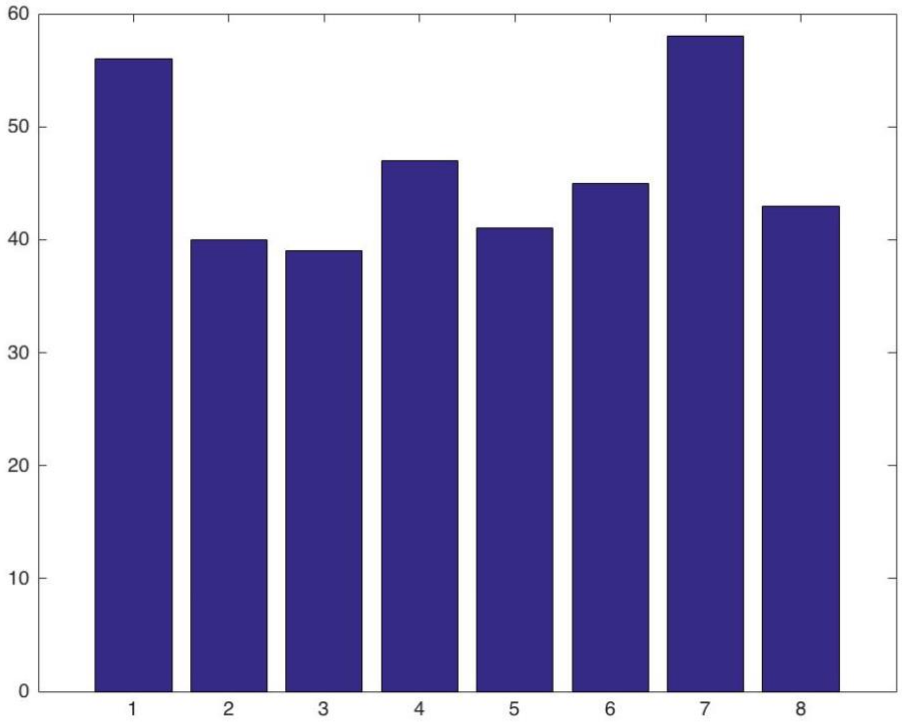
Optimus prime truck 8-features histogram



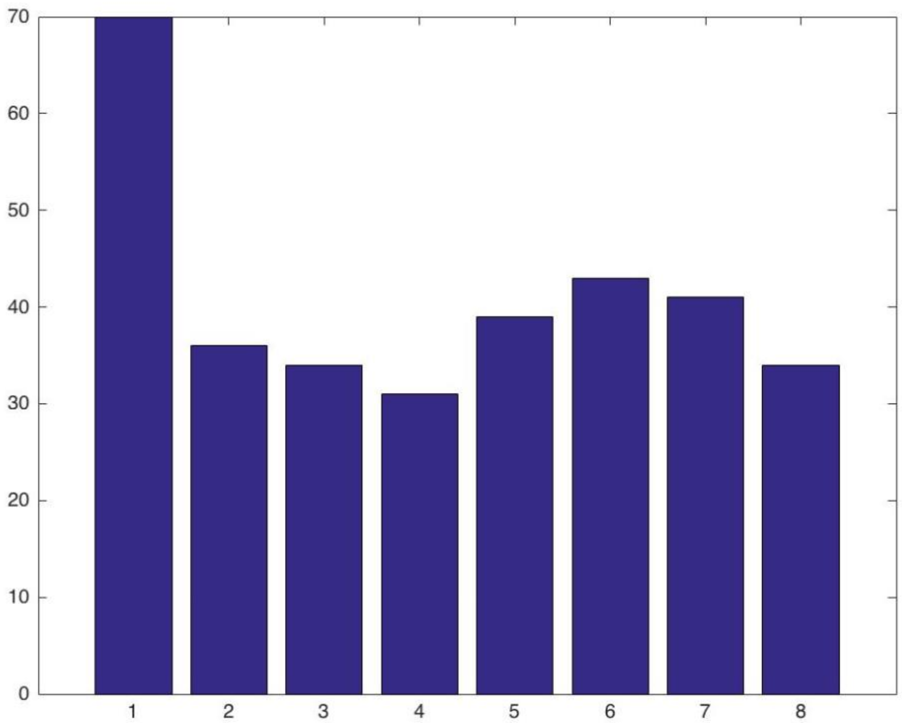
Bumblebee car 8-features histogram



Ferrari_1 8-features histogram



Ferrari_2 8-features histogram



(3) Matching Codewords

```
Feature Histogram of Bumblebee
130 119 56 58 63 52 95 47
Feature Histogram of Optimus
157 195 164 233 289 181 384 170
Feature Histogram of Ferrari
56 40 39 47 41 45 58 43
Feature Histogram of testing image
70 36 34 31 39 43 41 34
______________final result__________________
It Belongs to Ferrari
```

4.Discussion

According to my observation, this algorithm successfully recognizes Ferrari2 and cluster it with Ferrari together. By observe their 8-features histogram, its obvious that Ferrari1 and Ferrari2 are most similar.