EE569 Digital Image Processing

HOMEWORK #4

Texture Analysis and Segmentation / Image Feature Extractors

Issued: 3/4/2020 **Due: 3/22/2020**

Xu Kangyan

4876-0109-98

kangyanx@usc.edu

Problem 1: Texture Analysis and Segmentation

(a) Classification --- Feature Extraction

(b) Classification --- Classifier Explore K-means / Random Forest / SVM

(c) Texture Segmentation

(d) Advanced Texture Segmentation

I. Abstract and Motivation

(a) Classification

We use Laws Filters to do texture analysis. First subtract image global mean to reduce illumination effects. Then 5×5 Laws Filters (products of L5,E5,S5,W5,R5) are used to filter each texture image and get 25 filtered images. In classification part, average the absolute values through all the pixels in each filtered image to get each energy feature, 25 in total. This is a 25D Energy Feature Vector. Then replace 10 pairs (e.g. L5E5/E5L5) with their average to reduce the vector's dimension to 15D. Feature Vectors can be used for further work.

(b) Segmentation

In this part, a window is introduced. Given there is only one image for segmentation, first we also subtract image global mean and use Laws Filters to get 25 filtered images. Then the Energy Feature Vector is focused on each pixel, not the whole image. For each pixel, on each filtered image, we have a window (pixel as its center), and we average the absolute values through all the pixels in that window. Finally, each pixel has a 25D Energy Feature Vector. Same replacing operation (in (a)) is done to get 15D vector. We also use L5L5 to do feature normalization here, reduce each pixel's feature vector to 14D. Feature Vectors can be used for further work.

II. Approach, Procedures and Experimental Results

(a) Classification --- Feature Extraction

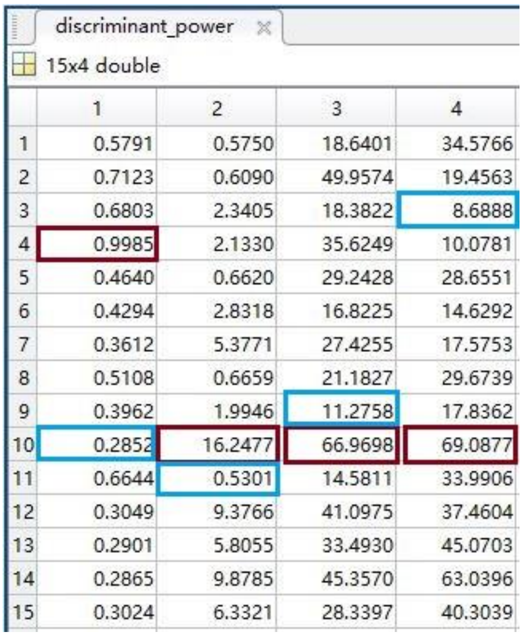
In this part, we have four types of feature: rice, grass, brick, blanket. 36 texture images are labelled, each texture has 9 images. Another 12 texture images are not labelled.

1. Feature Extraction: mirror reflection is used. Operation described in I (a).

2. Feature Averaging: Discriminant Power.

Here use the ratio of inter-class variance and intra-class variance to evaluate. The larger the ratio, the strongest the discriminant power this feature dimension has.

For each feature in each class, inter-class variance is the variance of 36 same-dimension features ($9\ image\ \times 4\ class$), while intra-class variance is the variance of 9 same-dimension features in same class. Given there is 15 dimension and 4 class, 60 ratios are calculated.



discriminant_power

15x4 double

	1	2	3	4
1	0.5791	0.5750	18.6401	34.5766
2	0.7123	0.6090	49.9574	19.4563
3	0.6803	2.3405	18.3822	8.6888
4	0.9985	2.1330	35.6249	10.0781
5	0.4640	0.6620	29.2428	28.6551
6	0.4294	2.8318	16.8225	14.6292
7	0.3612	5.3771	27.4255	17.5753
8	0.5108	0.6659	21.1827	29.6739
9	0.3962	1.9946	11.2758	17.8362
10	0.2852	16.2477	66.9698	69.0877
11	0.6644	0.5301	14.5811	33.9906
12	0.3049	9.3766	41.0975	37.4604
13	0.2901	5.8055	33.4930	45.0703
14	0.2865	9.8785	45.3570	63.0396
15	0.3024	6.3321	28.3397	40.3039

Figure 1. Ratio (Largest-R / Weakest-B)

Each dimension's Laws Filter: L5L5 L5E5/E5L5 E5S5/S5E5 E5E5 L5S5/S5L5 E5W5/W5E5 S5S5 L5W5/W5L5 E5R5/R5E5 W5W5 L5R5/R5L5 S5W5/W5S5 R5R5 W5R5/R5W5 S5R5/R5S5

3. Feature Reduction:

PCA is used in MATLAB to reduce dimension. Here PCA is realized via **SVD**. Three largest singular values and corresponding eigen vectors are used.

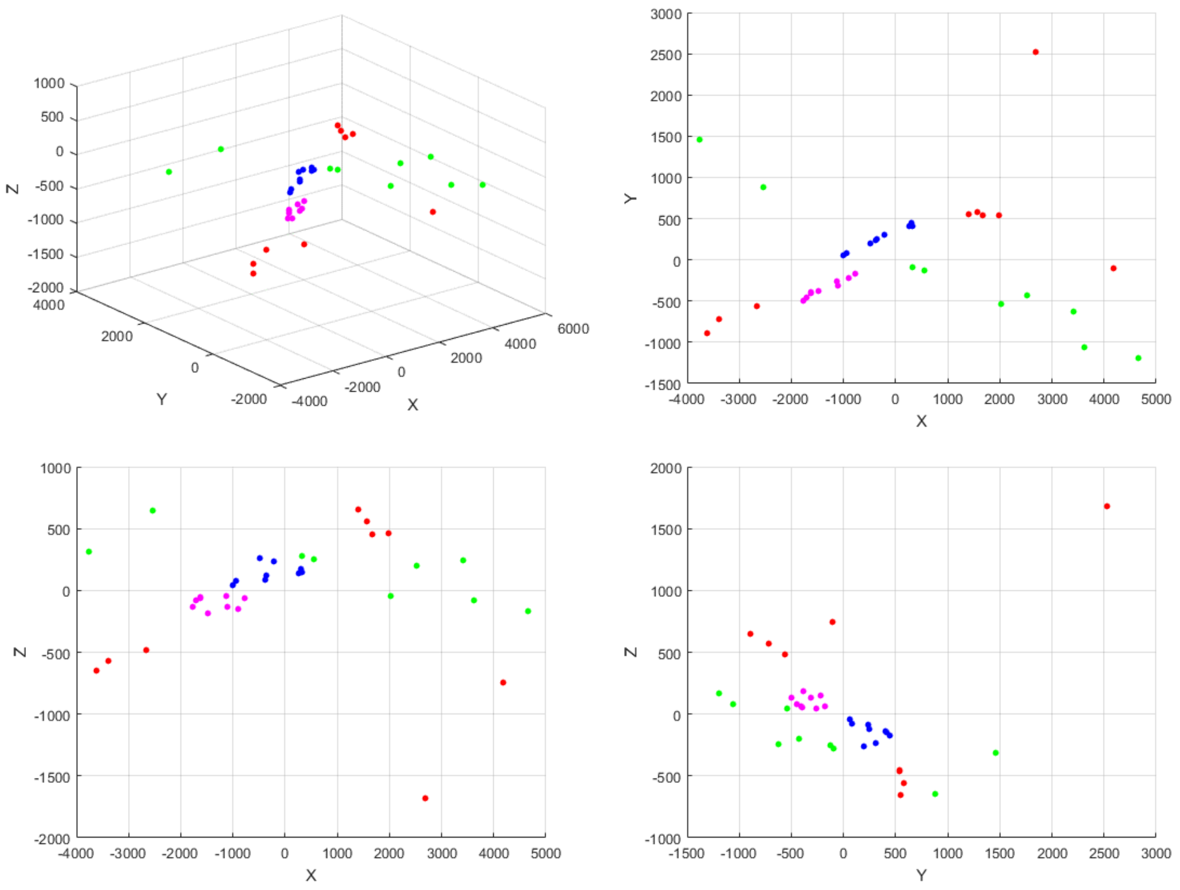


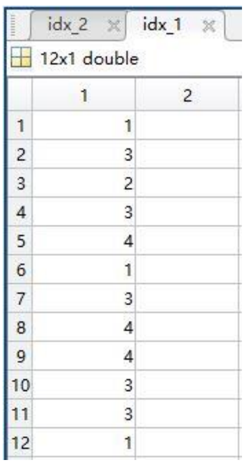
Figure 2. Reduced 3D Feature Vector in Feature Space

Blanket-R / Brick-G / Grass-B / Rice-M

We can see it's relatively easy to circle Rice Pattern and Grass Pattern, while Blanket Pattern and Brick Pattern are separate.

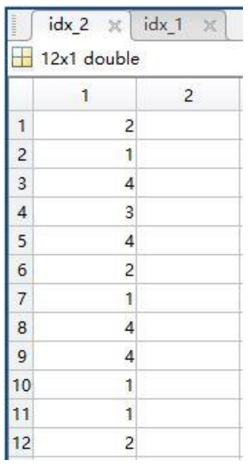
(b) Classification --- Classifier Explore K-means / Random Forest / SVM

1. Unsuperivsed: K-means cluster is used here on 15D and 3D vector.



12x1 double

	1	2
1	1	
2	3	
3	2	
4	3	
5	4	
6	1	
7	3	
8	4	
9	4	
10	3	
11	3	
12	1	



12x1 double

	1	2
1	2	
2	1	
3	4	
4	3	
5	4	
6	2	
7	1	
8	4	
9	4	
10	1	
11	1	
12	2	

Figure 3. K-means Result (left-15D / right-3D)

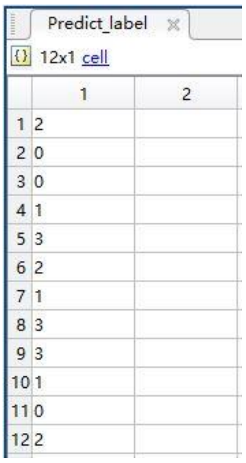
Use 0,1,2,3 to represent Blanket, Brick, Grass and Rice. 12 test images are labelled manually on the first row.

label	2	**0**	**0**	**1**	3	2	**1**	3	3	**1**	**0**	2
15D	1	**3**	**2**	**3**	4	1	**3**	4	4	**3**	**3**	1
label	2	**0**	**0**	**1**	3	2	**1**	3	3	**1**	**0**	2
3D	2	**1**	**4**	**3**	4	2	**1**	4	4	**1**	**1**	2

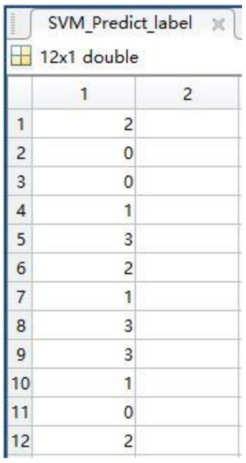
Both 15D and 3D has same results: Rice Pattern and Grass Pattern can be circled, while Blanket and Brick are mixed. The error rate is 0.5 .

I didn't implement K-means.

2. Superivsed: Random Forest and Support Vector Machine on 3D vectors.



```
Predict_label
12x1 cell
     1     2
1  2
2  0
3  0
4  1
5  3
6  2
7  1
8  3
9  3
10 1
11 0
12 2
```



```
SVM_Predict_label
12x1 double
     1     2
1    2
2    0
3    0
4    1
5    3
6    2
7    1
8    3
9    3
10   1
11   0
12   2
```

Figure 4. Supervised Result (left-RF / right-SVM)

Both results are 100% correct.

(c) Texture Segmentation

The operation is described in I (b).

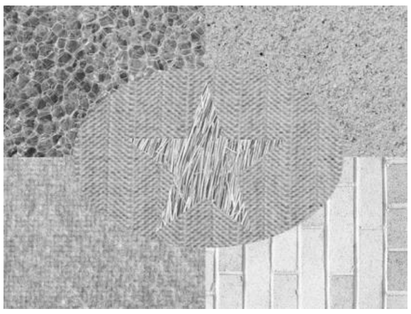


Figure 5. Image with Six Textures

Different window sizes are tested in the code and the results are below.

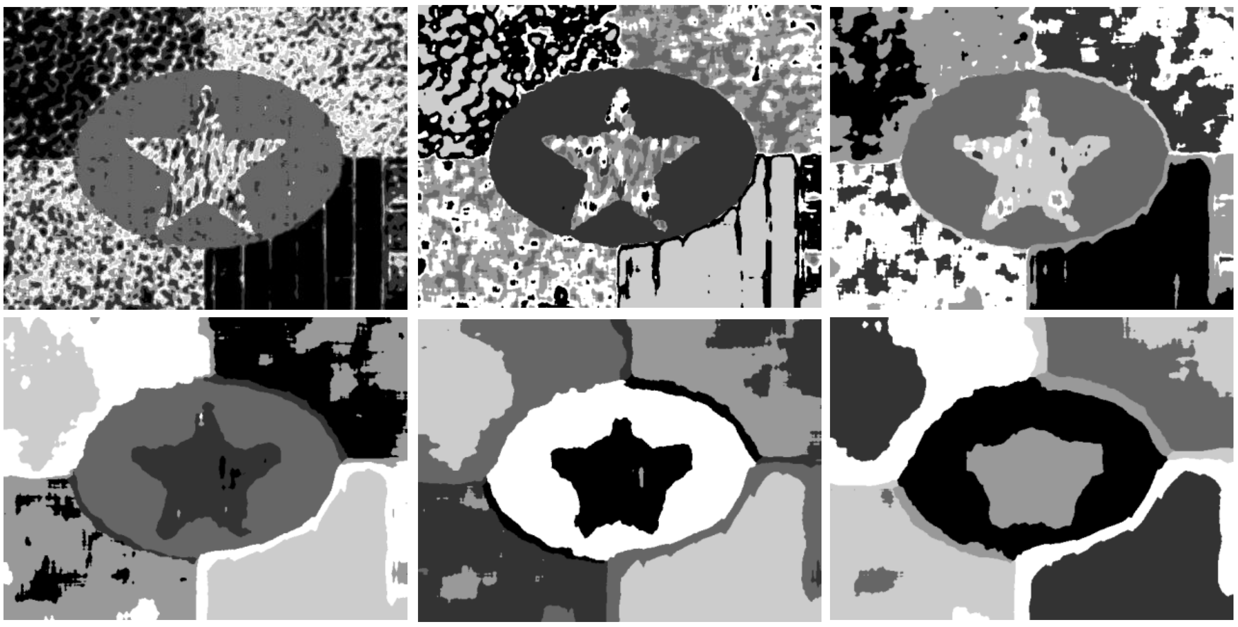


Figure 6. Image Segmentation

Each grey scale represents a certain texture the pixel belongs to.

The window size tested is **7**, **13**, **21**, **35**, **49**, **63**.

The result from window size **35** looks better than others. The area boundary is relatively clear and not out of shape.

(d) Advanced Texture Segmentation

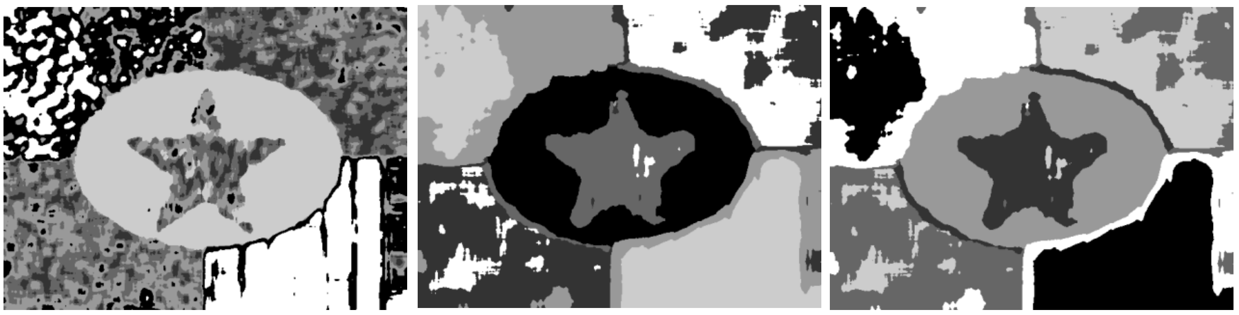


Figure 7. Image Segmentation

PCA is used. First image is 3D vector with window size 7, following are 3D with 35 and 6D with 35. Unfortunately, PCA seems have no change to the performance of segmentation.

Problem 2: Image Feature Extractors

(a) Salient Point Descriptor

(b) Image Matching

(c) Bag of Words

I. Abstract and Motivation

Scale Invariant Feature Transform (SIFT) can find interest points in images and it's useful for image mapping. Difference-of-Gaussian (DoG) is used as an approximation for Laplacian of Gaussian. Key point is selected when it is a local minimum or maximum in the 3D DoG scale space.

SIFT Descriptor is a 128 dimension vector (8 dimension vector per subblock).

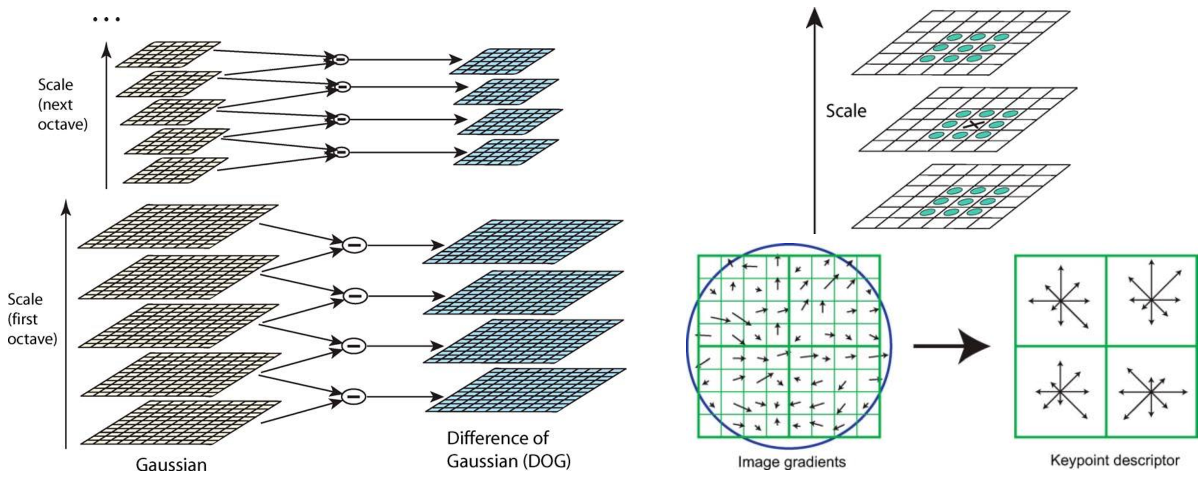


Figure 8. Basic Idea in SIFT

Normally, $k = \sqrt{2}$, $\sigma = 1.6$, scale = 5 and octaves = 4.

II. Approach, Procedures and Experimental Results

(a) Salient Point Descriptor

1. Robust to Image scale and rotation.

2. For Image Scale, we want to find locations that are invariant to scale change, so we can find stable features across all possible scales (Scale Space). Scale Space is defined as images convoluted by variable-scale Gaussian, then we can use DoG to approximate LoG, which is easier to find local extrema.

For Rotation, 128D Descriptor is calculated.

3. Robust to illumination: Normalize the descriptor vector, cap each element to 0.2 and normalize again.

4. LoG is hard to calculate, and DoG here is a close approximation.

5. Feature vector size is keypoint_num×128dim.

(b) Image Matching (VLFeat)

1. Key-point with largest scale

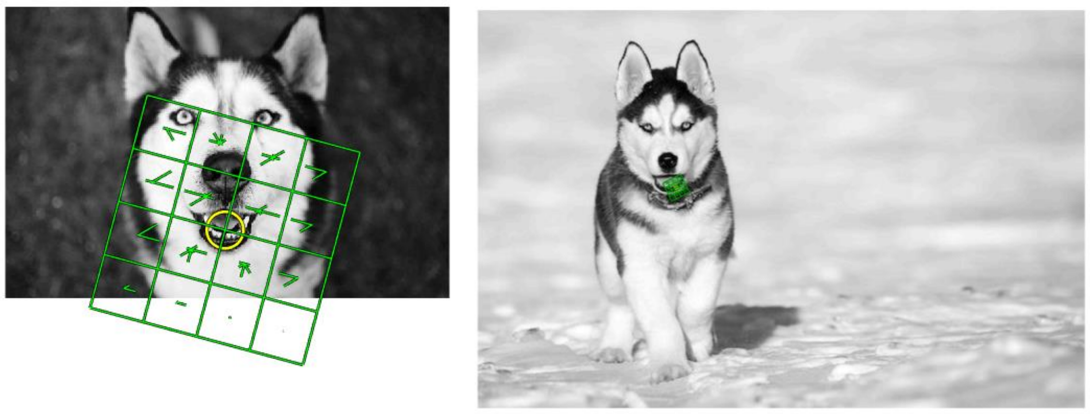


Figure 9. Key-point in Husky_3 and Husky_1

The largest scale in Husky_3 is 26.4549, orientation is 0.2626.

The closet neighboring key-point in Husky_1 has scale 1.7689, orientation is -2.0424.

L2-Norm is used for feature vector distance.

Descriptor vector is normalized.

2.SIFT pairs

The vl_sift( ) parameter is set as:

PeakThresh: 6, EdgeThresh: 12.1, Octaves:4, Levels: 5

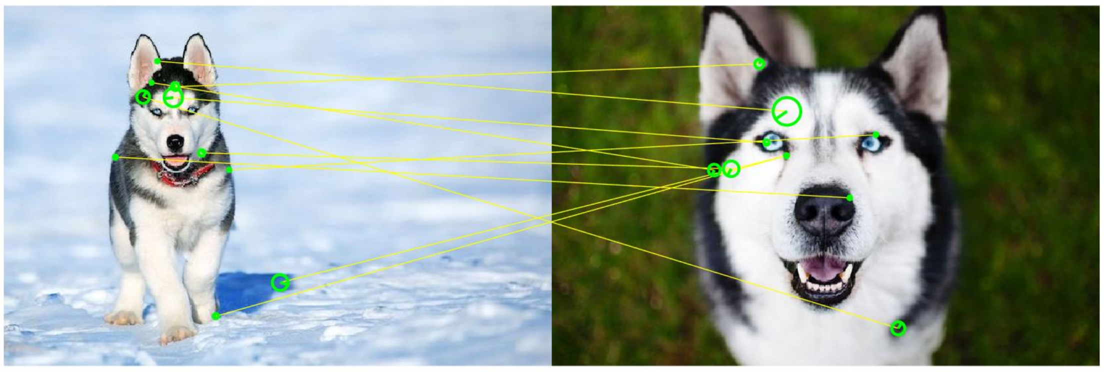


Figure 10. Pair between Husky_3 and Husky_1

The match is not very good.

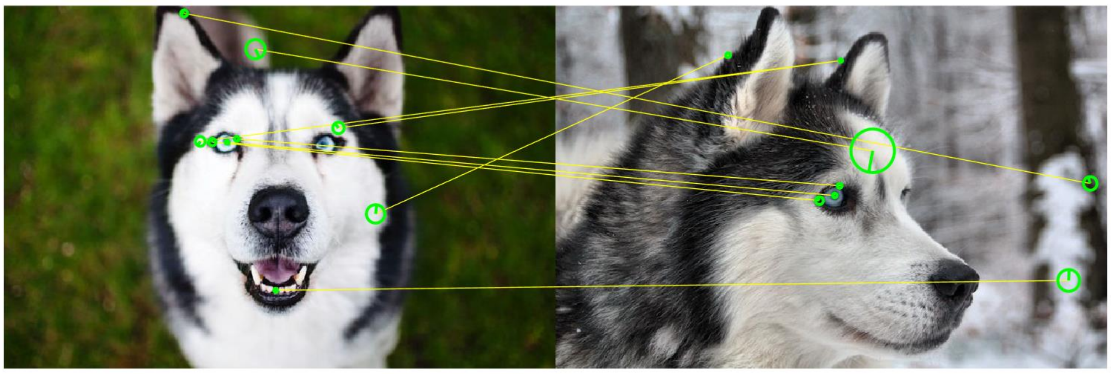


Figure 11. Pair between Husky_3 and Husky_2

The left eye matches good, other parts not good. Due to the view angle changing, it is likely to mismatch.

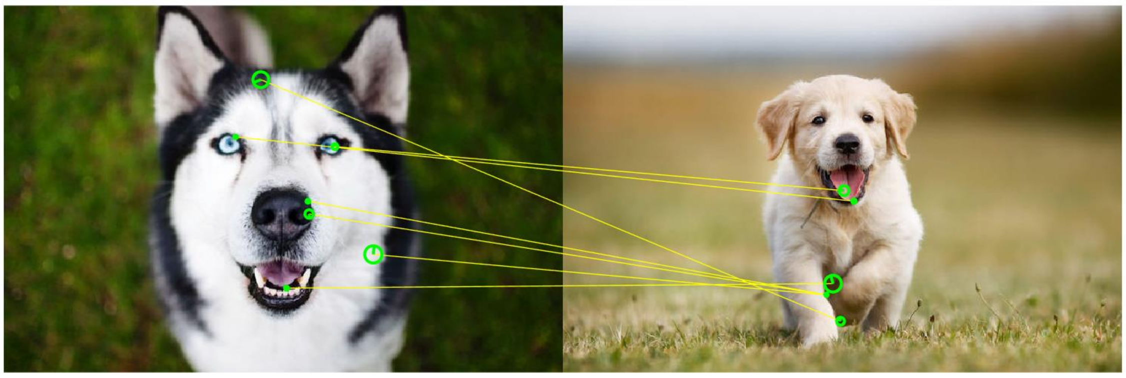


Figure 12. Pair between Husky_3 and Puppy_1

The match is not very good.

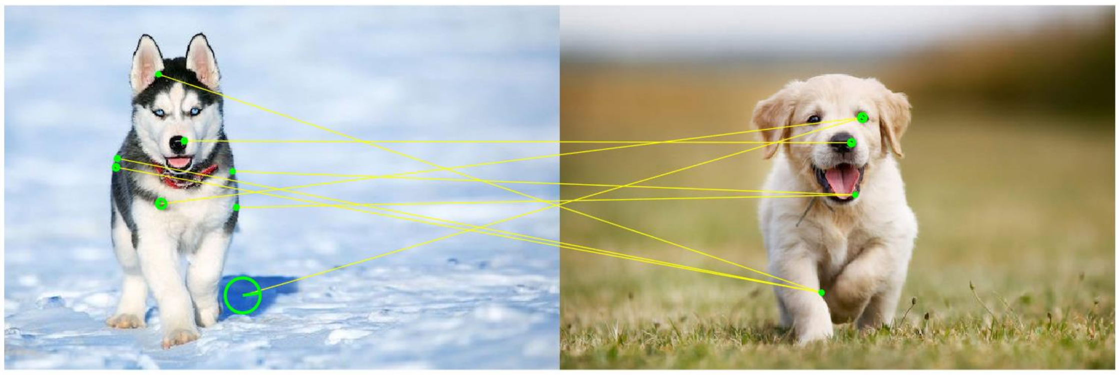


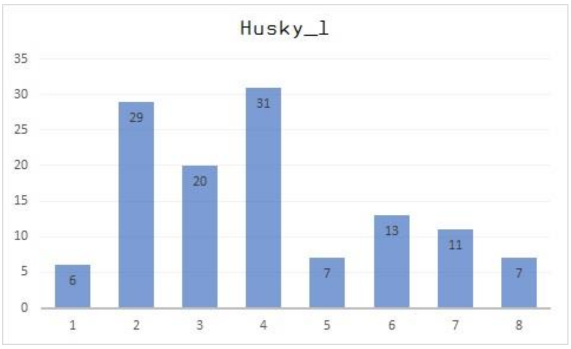
Figure 13. Pair between Husky_1 and Puppy_1

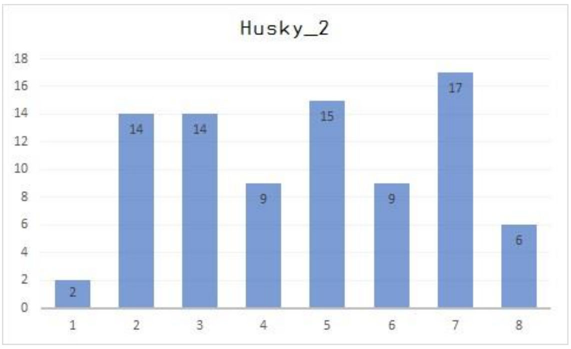
The nose matches good, other parts are not good.

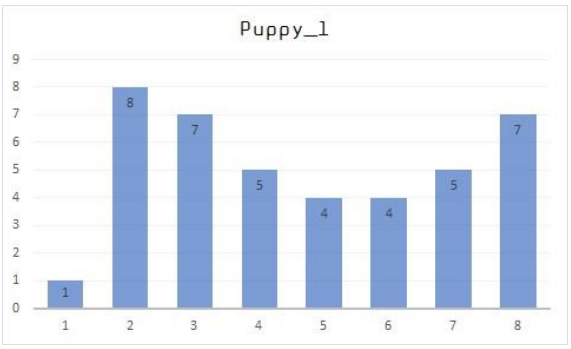
vl_ubcmatch( ) is used and the match effect is not good. The result is confusing. From human observation, Husky_1 and Puppy_1 are quite similar (esp. view angle, body shape), Husky_1 and Husky_3 are different on scale. The match should be better.

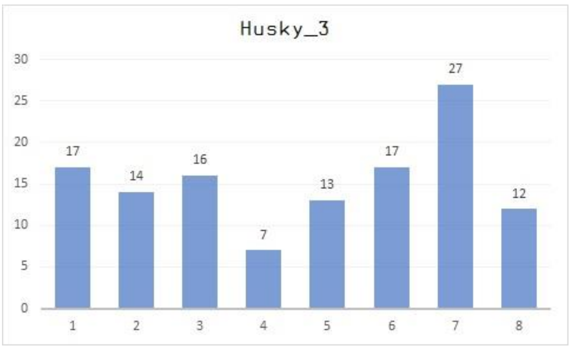
(c) Bag of Words

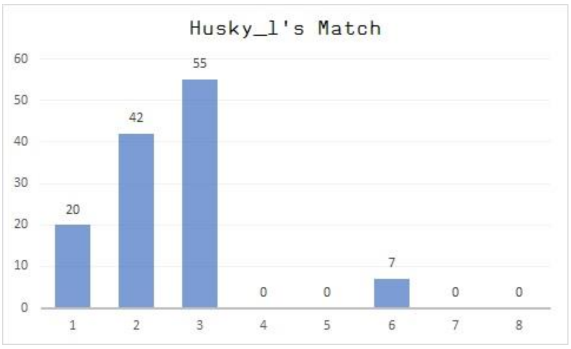
K-means (K=8) applied to each image’s SIFT feature vector.

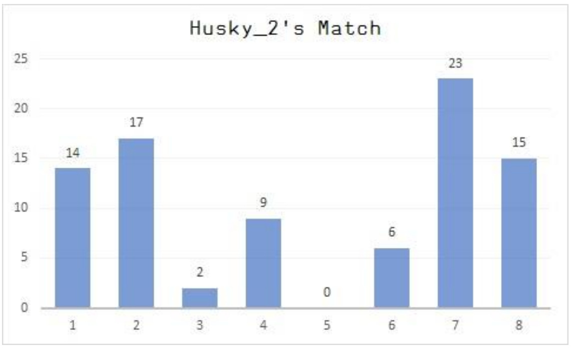


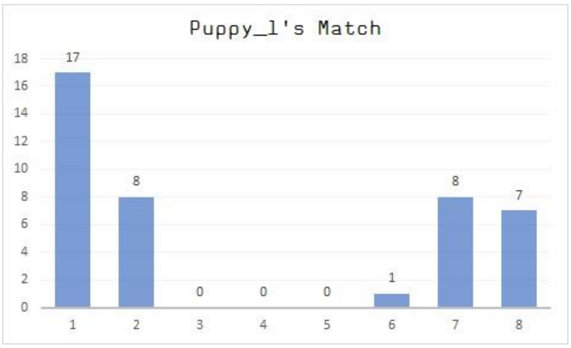












Puppy_1, Husky_1 and Husky_3 has similar histograms (bars from low to high if sorted). Puppy_1 has a large match on Husky_3’s word 1, Husky_1 has a large match on Husky_3’s word 3, which means Husky_3 has some parts similar to Husky_1 and Puppy_1.