Deep Learning for Computer Vision (2018 Spring)

HW2

B03611026 郭冠軒

Problem 1

$$K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^\top \mathbf{x}')^2 = \Phi(\mathbf{x})^\top \Phi(\mathbf{x}')$$

Given $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$,

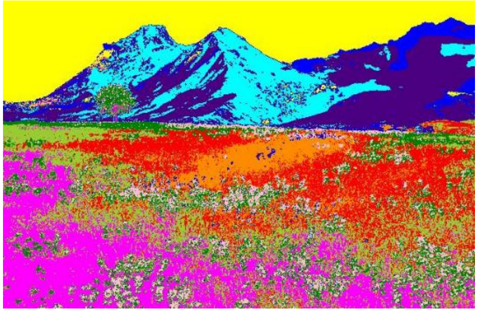
$$\begin{aligned}
K(\mathbf{x}, \mathbf{x}') &= (\mathbf{x}^\top \mathbf{x}')^2 \\
&= \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^\top \begin{bmatrix} x_1{}' \\ x_2{}' \end{bmatrix} \right)^2 \\
&= (x_1\, x_1{}' + x_2\, x_2{}')^2 \\
&= (x_1\, x_1{}')^2 + 2(x_1\, x_1{}'\, x_2\, x_2{}') + (x_2\, x_2{}')^2 \\
&= \begin{bmatrix} x_1{}^2 \\ \sqrt{2}\, x_1 x_2 \\ x_2{}^2 \end{bmatrix}^\top \begin{bmatrix} x_1{}'^{\,2} \\ \sqrt{2}\, x_1{}'\, x_2{}' \\ x_2{}'^{\,2} \end{bmatrix} \\
&= \Phi(\mathbf{x})^\top \Phi(\mathbf{x}')
\end{aligned}$$

$$\Rightarrow \Phi(\mathbf{x}) = \begin{bmatrix} x_1{}^2 \\ \sqrt{2}\, x_1 x_2 \\ x_2{}^2 \end{bmatrix}$$

Problem 2

(a) Color segmentation

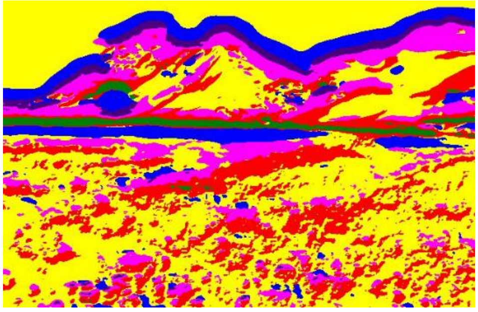
(i) RGB color feature.



(ii) Lab color feature.


(b) Texture segmentation

(i) Texture feature.

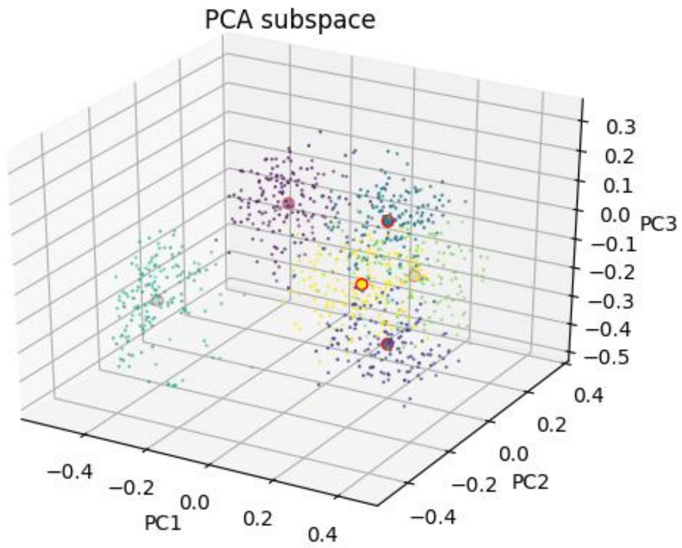
(ii) Texture feature & Lab color feature.


Problem 3

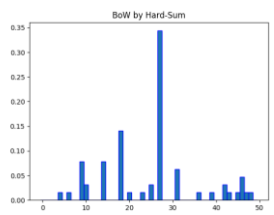
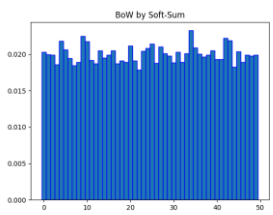
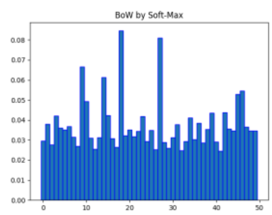
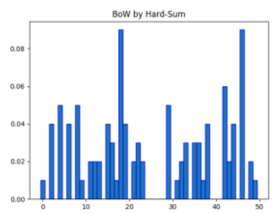
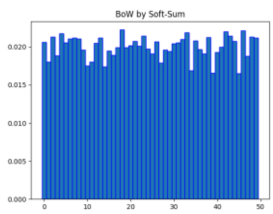
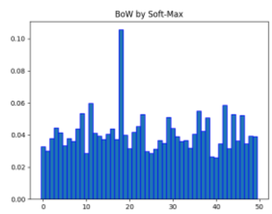
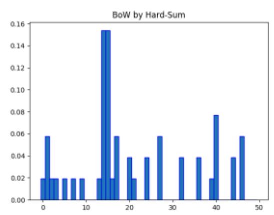
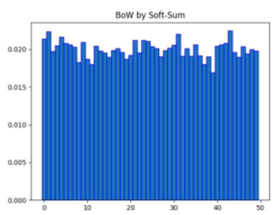
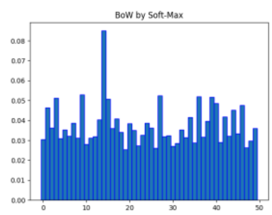
(a) Interest point detection. ("image_0006.jpg" from "train-10/Coast/" dataset)

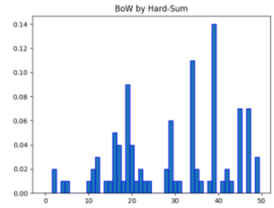
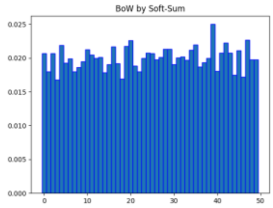
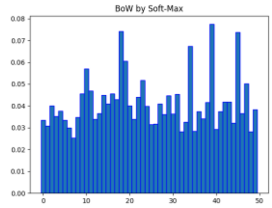
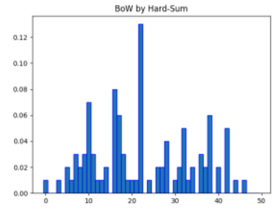
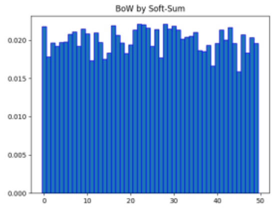
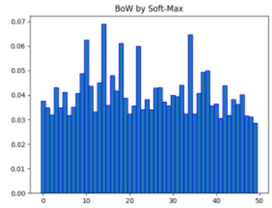
(b) Plot PCA subspace.

Here is the result of the PCA subspace in three dimension. The selected six clusters are plotted with six different colors of markers, and the markers of visual words are plotted with red edge color especially.



(c) BoW features of Hard-Sum, Soft-Sum, and Soft-Max.

BoW strategy	Hard-Sum	Soft-Sum	Soft-Max
Coast			
Forest			
Highway			

Mountain			
Suburb			

I expect that the "Soft-Max" strategy could result in better classification results.

We may observe several characteristics of "Soft-Max" from the BoW plot:

1. It takes almost every visual words into account (with different weights).
2. It has a few peaks in some specific visual words.

Although the "Hard-Sum" strategy has some peaks in specific visual words, it does not consider every visual words when deciding what the category of the picture is.

The "Soft-Sum" strategy does consider every visual words, but it does not have the visual words with outstanding weights.

According to the description above, we may say that the "Soft-Max" strategy have both of the merits of "Hard-Sum" and "Soft-Sum", and in my opinion, it could result in better classification results than the other two strategies.

(d)

(i) Use **Train-10** as training data and **Test-100** as testing data.

BoW strategy	Hard-Sum	Soft-Sum	Soft-Max
accuracy	0.468	0.482	0.416

The results are not as expected based on my observation in (c).

There are still lots of factors that may affect the classification accuracy, such as the scheme of picking interesting points (Hessian threshold in SURF), the diversity of training data (50 training images, 500 testing images in this task), the cluster numbers of visual words, the neighbor numbers of KNN, etc.

So, the classification results may vary while taking these factors into consideration.

(ii) Use **Train-100** as training data and **Test-100** as testing data.

BoW strategy	Hard-Sum	Soft-Sum	Soft-Max
accuracy	0.630	0.652	0.586

* Hessian threshold = 400, n_clusters = 70

The classification results improved as I changed the training data from **Train-10** to **Train-100**.

I have tried to change the parameters including Hessian threshold of SURF and the cluster numbers of visual words.

For Hessian threshold, I tried three different values: 200, 400, and 600. I got the best classification accuracy when Hessian threshold = 200. The smaller (but not too small) threshold value we set, the more interest points we will get. So, it was reasonable to get this kind of result.

Then, for cluster numbers of visual words (n_clusters), I also tried three different values: 30, 50, and 70. I got the best classification accuracy when n_clusters = 70, and the worst result occurred when n_clusters = 50. Unfortunately, I couldn't find any reason to explain the results.