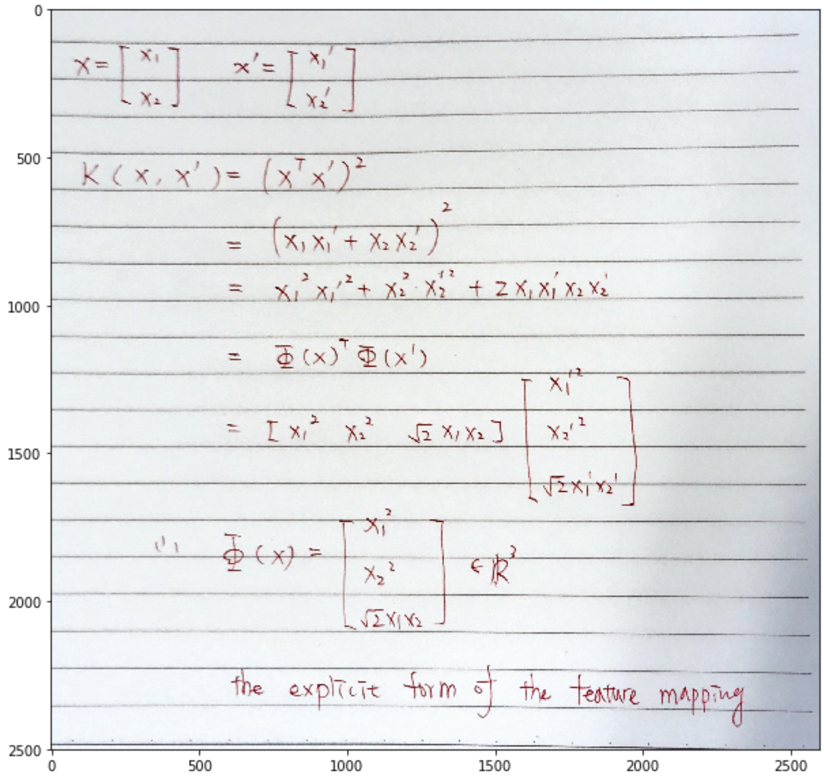
DLCV HW#2 - d05921027 張鈞閔

Out[51]: Click here to toggle on/off the raw code.

Problem 1

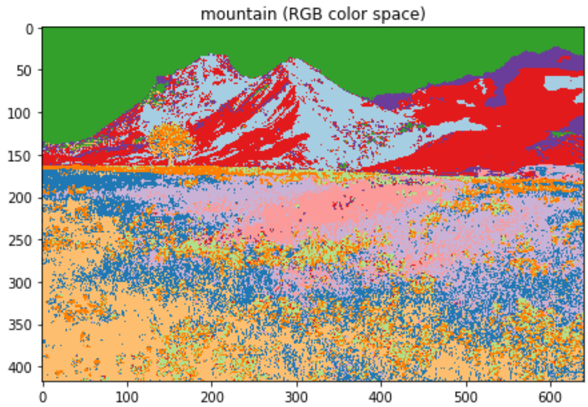
Out[4]: <matplotlib.image.AxesImage at 0x121bb9ef0>

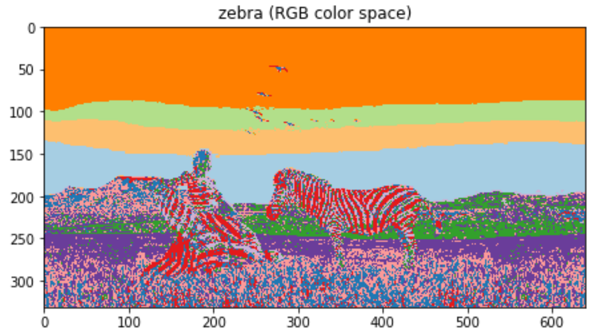


Problem 2

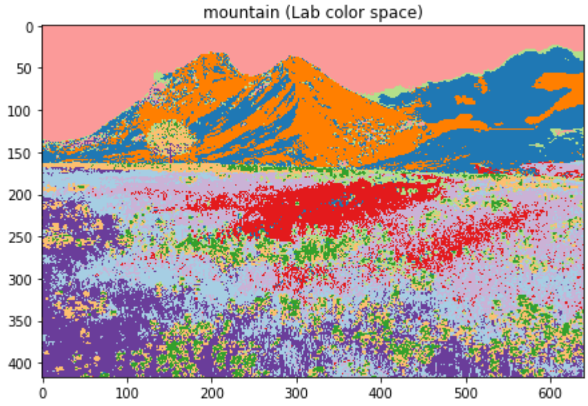
(a) color segmentation

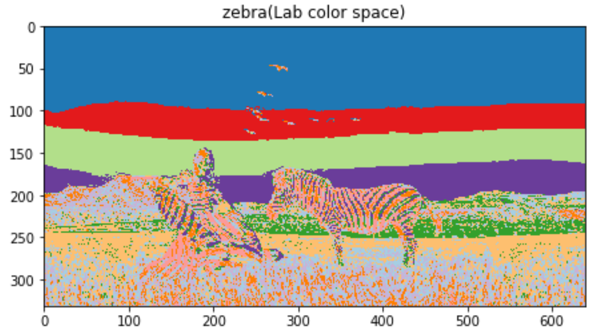
(a-i) RGB color space





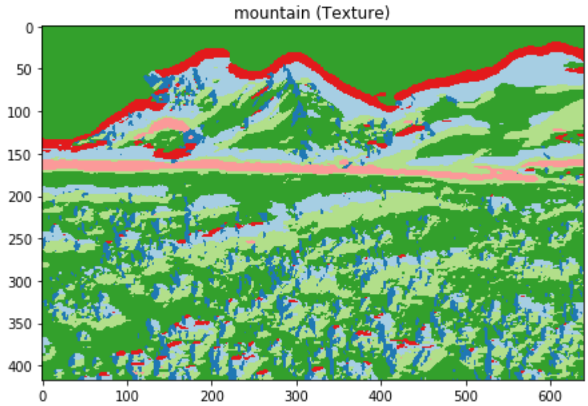
(a-ii) Lab color space

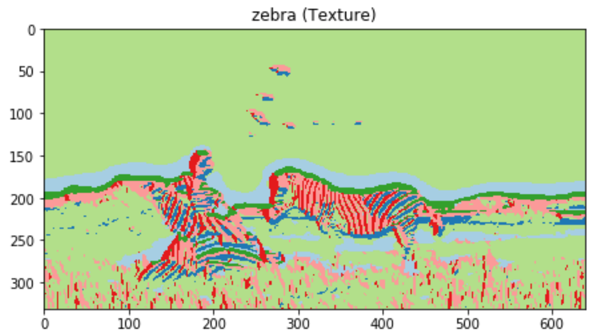




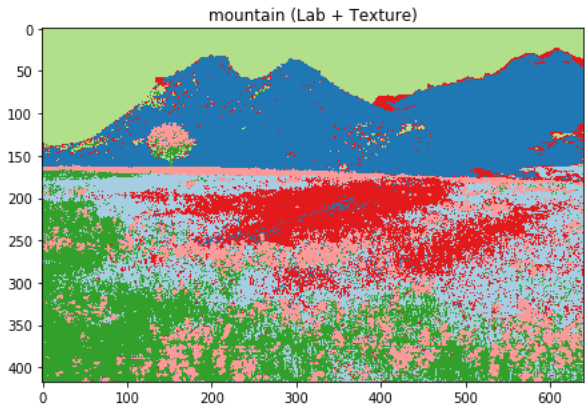
(b) texture segmentation

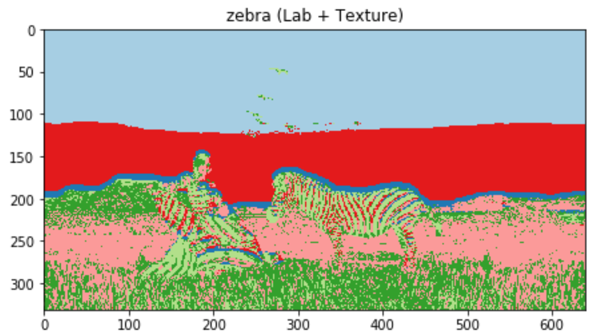
(b-i) texture only





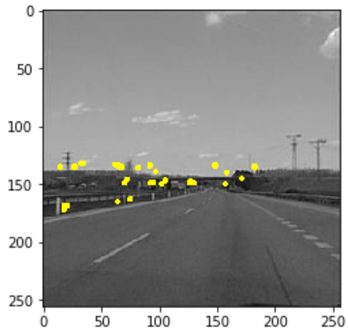
(b-ii) RGB+texture





Problem 3

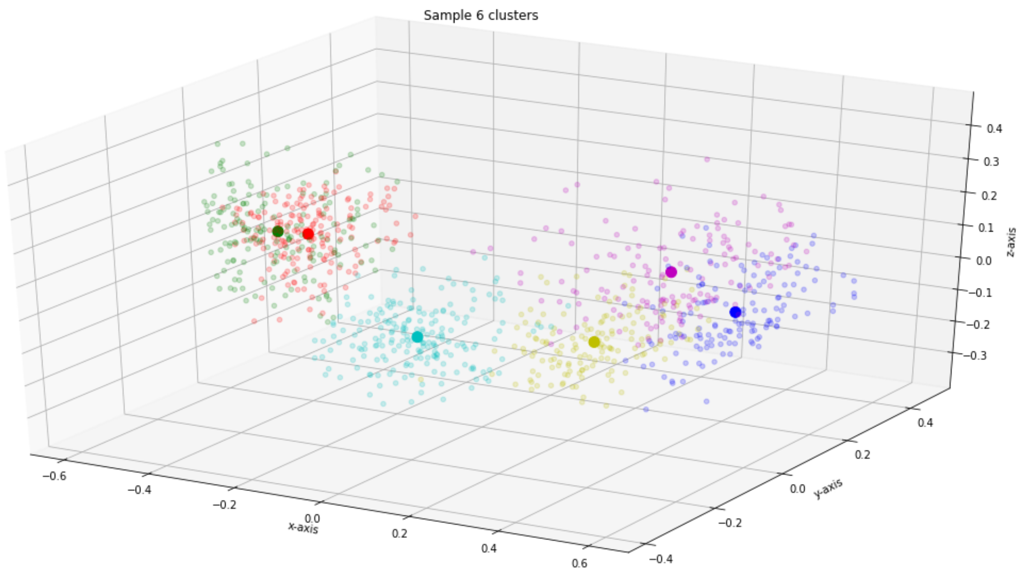
(a) randomly sample one image and extract its top 30 SURF interest points



(b) Visual words (KMeans) on the Train-10 dataset and Visualize in PCA features

```
interest points (9332,)
descriptors: (9332, 64)


pca descriptors: (9332, 3)
```



(c) Bag of visual words: Hard-Sum, Soft-Sum, Soft-Max

I think **SoftMax** is a better BoW strategy for classification because, for an image, SoftMax preserves the maximum likely description in the each visual word. Moreover, using soft labels benefits the smoothness of decision boundaries, which is good for generalization.

(d-i) kNN on Test-100

```
Trained kNN on Train-10 dataset.

Test-100 interest points: (94492,)
Test-100 descriptors: (94492, 64)


Test-100 accuracy of HardSum:  0.46
Test-100 accuracy of SoftSum:  0.366
Test-100 accuracy of SoftMax:  0.496
```

(d-ii) Train on Train-100 and test on Test-100

```
Train-100 interest points (102521,)
Train-100 descriptors: (102521, 64)

C=100 clusters as visual words
Maximum iteration=5000
The number of k=10 nearest neighbor



Trained kNN on Train-100 dataset.

Test-100 Accuracy of HardSum:  0.544
Test-100 Accuracy of SoftSum:  0.578
Test-100 Accuracy of SoftMax:  0.558
```