[CV21] Assignment 5 - Di Zhuang

2 Bag-of-words classifier

2.1 Local feature extraction

2.1.1 Feature detection - feature points on a grid

In grid_points(), use np.linspace() and np.meshgrid() to compute a regular grid that fits the given image with granularity 10 by 10 and border 8.

2.1.2 Feature description - histogram of oriented gradients

In descriptors_hog(), for each grid point, first compute the upper right corners of all cells around it to locate the 16 cells. Next, for each cell, get the x gradient and y gradient of each pixel in the cell and compute the direction of the gradient of each pixel in the cell. Afterwards, compute a histogram for all directions of gradients computed for the pixels in the cell. Concatenate the histogram to the descriptor of the grid point. Finally, put descriptors of all grid points into a 2D array.

2.2 Codebook construction

In create_codebook(), for each image, compute the grid points and the corresponding histogram descriptors. Apply the K-means algorithm to cluster all descriptors from all images. The cluster centers are the visual words we need.

2.3 Bag-of-words vector encoding

2.3.1 Bag-of-words histogram

In bow_histogram(), for each descriptor of the image, compute its distance to all visual words in the codebook, assign the descriptor to the visual word to which its distance is the shortest. Increment the count for the corresponding visual word by 1. At the end we obtain a histogram that describes the occurrences of the visual words in the image.

2.3.2 Processing a directory with training examples

In create_bow_histograms(), for all images in the directory, compute their bag-of-words histogram.

2.4 Nearest neighbor classification

In bow_recognition_nearest(), for each test image, compute the distance between its bag-of-words histogram and that of all training images. Label the test image with the label of the training image whose bag-of-words histogram is the nearest to that of the test image.

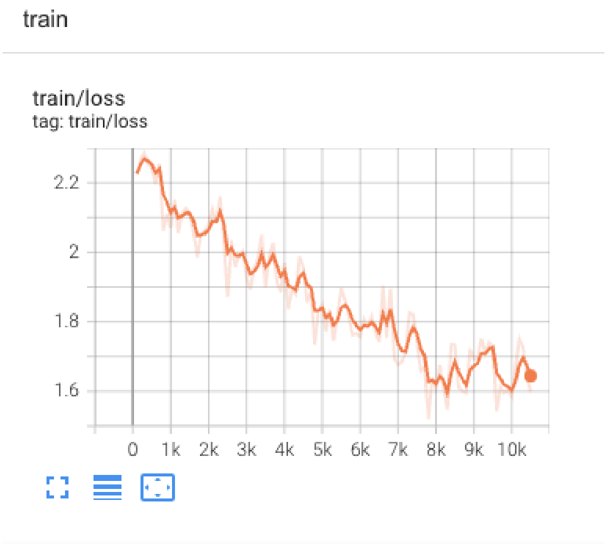
Result

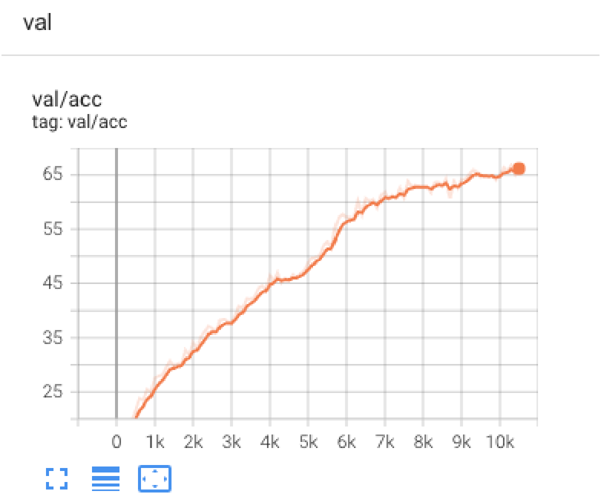
By testing several combinations of K and numiter, we decide to use K= 7 and numiter = 14. This gives the following result:

test pos sample accuracy: 0.9387755102040817
test neg sample accuracy: 0.94

3 CNN-based classifier (model trained for 30 epochs)

```
[INFO] test set loaded, 10000 samples in total.
79it [00:32,  2.41it/s]
test accuracy: 65.18
```


train
train/loss
tag: train/loss
1k
2k
3k
4k
5k
6k
7k
8k
9k
10k


val
val/acc
tag: val/acc
1k
2k
3k
4k
5k
6k
7k
8k
9k
10k