

CV Assignment 5, BoW & VGG

BoW

Bow is implemented according to the guidance of the handout:

1. feature points on a grid: used meshgrid to generate equally spaced grid based on the arguments
2. histogram of oriented gradients: first, iterate over all points. For each point, iterate over the pixels centered on the pixel with range defined by the cells. Then get the gradient and discretize the gradient direction into the 8 bins using arctan function and the x, y coordinates of the gradient (also corrected by adding 2π if the vector is pointing to the other half of the space). Then the histogram is constructed by concatenating all the histograms for all the cells to get the feature vector.
3. codebook construction: used the previously written functions to find the feature for each image, and used k-means clustering to get the reduced set.
4. BoW representation: implemented a simple procedure to find the counts of assigning each feature to each center.
5. create BoW histograms: this called the procedures implemented above.

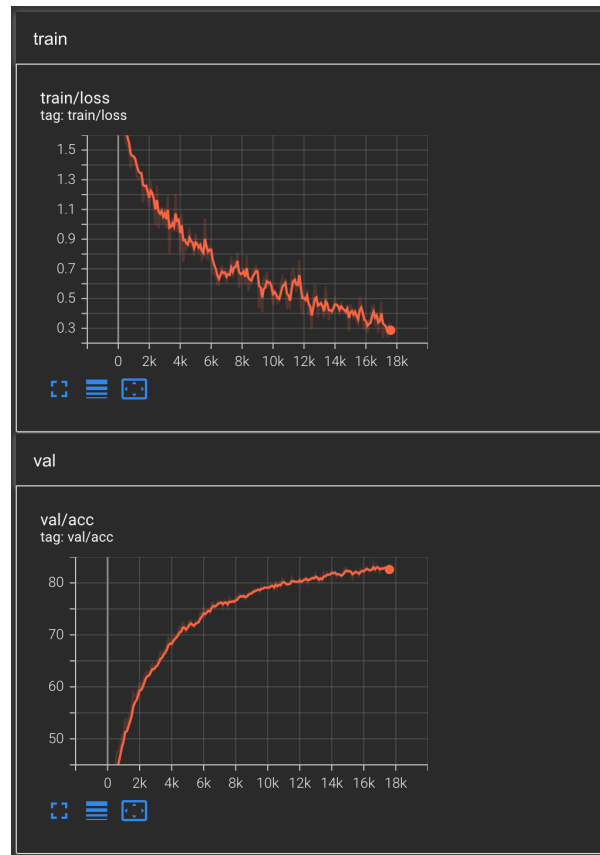
Setting the number of clusters to be 20 and the number of iteration to 60, the result is shown below:

```
(cv) ~ exercise5_object_recognition python3 bow_main.py
creating codebook ...
100% | 100/100 [00:36<00:00, 2.76it/s]
number of extracted features: 10000
clustering ...
creating bow histograms (pos) ...
100% | 50/50 [00:17<00:00, 2.90it/s]
creating bow histograms (neg) ...
100% | 50/50 [00:18<00:00, 2.73it/s]
creating bow histograms for test set (pos) ...
100% | 49/49 [00:16<00:00, 2.97it/s]
testing pos samples ...
test pos sample accuracy: 0.8571428571428571
creating bow histograms for test set (neg) ...
100% | 50/50 [00:17<00:00, 2.81it/s]
testing neg samples ...
test neg sample accuracy: 0.98
```

where the test accuracies for the pos and neg are: 85% and 98%.

VGG

The architecture of the VGG model is implemented strictly following the guidance of the handout. The final linear layers have a hidden dimension equal to the size of the argument. The training is done with 50 epochs and the lose (acc) is plotted below:



Running the test script, we get an accuracy of 81%.

```
(cv) ~ exercise5_object_recognition python3 test_cifar10_vgg.py --model_path=runs/81913/last_model.pkl
[INFO] test set loaded, 10000 samples in total.
79it [00:26, 2.95it/s]
test accuracy: 81.73
```