EE569 Digital Image Processing

COMPETITION #2

CIFAR10 Classification using SSL

Issued: 4/8/2020 **Due: 5/3/2020**

Xu Kangyan

4876-0109-98

kangyanx@usc.edu

Problem 3: CIFAR10 Classification using SSL

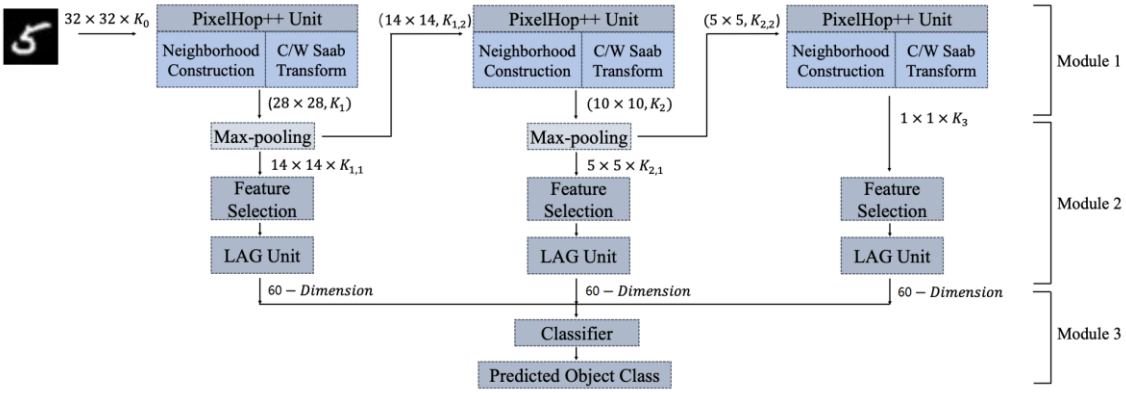


Figure 1. Block diagram of SSL model

I. Motivation and logics

1. Motivation

The hyper parameters in Module 1 are not changed. In Module 2, Number of selected features (Ns) and Number of centroids per class in LAG units are changed to reach better performance. Due to the limitation of calculation resource, only 3125 training images (instead of 50000) are used for Module 2 training and 3500 testing images are used for calculating test accuracy. There is no major change in the SSL model.

2. hyper-parameter

Spatial Neighborhood size in all PixelHop++ units	5x5
Stride	1
Max-pooling	(2x2) -to- (1x1)
Energy threshold for intermediate nodes (*TH1*)	0.001
Energy threshold for discarded nodes (*TH2*)	0.0001
Number of selected features (N_s) for each Hop	528
α in LAG units	10
Number of centroids per class in LAG units	6
Classifier	Random Forest (recommended)

Table 1. hyper-parameters

Multiple tests are done to determine the best parameter for Ns and Number of centroids in LAG.

```
Python Console
Using TensorFlow backend.
input feature shape: (6250, 32, 32, 3)
pixelhop2 transform
(6250, 28, 28, 42) (6250, 10, 10, 274) (6250, 1, 1, 528)
[[0.         1.44587748 0.         ... 0.10179067 2.26557493 0.21710318]
 [0.         0.19696179 1.88952725 ... 0.         0.75742221 1.04782381]
 [0.         0.34560697 0.         ... 0.         0.         0.        ]
 ...
 [0.         0.         0.09528495 ... 3.68140121 0.         0.        ]
 [0.12990583 0.02108835 0.25512547 ... 0.         1.27974927 0.        ]
 [0.00511505 0.         0.99494255 ... 0.43578355 1.67124298 0.        ]]
Time - 3130 sec
------- TRAINING FINISHED -------
```

Figure 2. Training Image Amount Selection

Even a selection of 6250 training images needs nearly 50 minutes for training. However, selection of 3125 images is much faster, but a slight change of this amount will cause a distinct time increase.

Selection Ns

```
Python Console
Using TensorFlow backend.
input feature shape: (3125, 32, 32, 3)
pixelhop2 transform
(3125, 28, 28, 42) (3125, 10, 10, 274) (3125, 1, 1, 528)
Ns_1 - 16464 Ns_2 - 13700 Ns_3 - 528
C:/Users/XBHOME/Desktop/Cmp2/Model_M2.py:148: DataConversionWarning:
  classifier.fit(feature_vector, y_train[0:train_num])
Time - 126 sec
------- TRAINING FINISHED -------

input feature shape: (3500, 32, 32, 3)
pixelhop2 transform
(3500, 28, 28, 42) (3500, 10, 10, 274) (3500, 1, 1, 528)
Accuracy on 3500 test images: 19.457 %
Time - 17 sec
------- TESTING FINISHED -------
```

```
Python Console
Using TensorFlow backend.
input feature shape: (3125, 32, 32, 3)
pixelhop2 transform
(3125, 28, 28, 42) (3125, 10, 10, 274) (3125, 1, 1, 528)
Ns_1 - 2000 Ns_2 - 1600 Ns_3 - 528
C:/Users/XBHOME/Desktop/Cmp2/Model_M2.py:148: DataConversionWarning:
  classifier.fit(feature_vector, y_train[0:train_num])
Time - 86 sec
------- TRAINING FINISHED -------

input feature shape: (3500, 32, 32, 3)
pixelhop2 transform
(3500, 28, 28, 42) (3500, 10, 10, 274) (3500, 1, 1, 528)
Accuracy on 3500 test images: 29.514 %
Time - 113 sec
------- TESTING FINISHED -------
```

```
Python Console
Using TensorFlow backend.
input feature shape: (3125, 32, 32, 3)
pixelhop2 transform
(3125, 28, 28, 42) (3125, 10, 10, 274) (3125, 1, 1, 528)
Ns_1 - 1200 Ns_2 - 1000 Ns_3 - 528
C:/Users/XBHOME/Desktop/Cmp2/Model_M2.py:150: DataConversionWarning:
  classifier.fit(feature_vector, y_train[0:train_num])
Time - 84 sec
------- TRAINING FINISHED -------

input feature shape: (3500, 32, 32, 3)
pixelhop2 transform
(3500, 28, 28, 42) (3500, 10, 10, 274) (3500, 1, 1, 528)
Accuracy on 3500 test images: 38.771 %
Time - 20 sec
------- TESTING FINISHED -------
```

```
Python Console
Using TensorFlow backend.
input feature shape: (3125, 32, 32, 3)
pixelhop2 transform
(3125, 28, 28, 42) (3125, 10, 10, 274) (3125, 1, 1, 528)
Ns_1 - 1000 Ns_2 - 1000 Ns_3 - 528
C:/Users/XBHOME/Desktop/Cmp2/Model_M2.py:148: DataConversionWarning:
  classifier.fit(feature_vector, y_train[0:train_num])
Time - 77 sec
------- TRAINING FINISHED -------

input feature shape: (3500, 32, 32, 3)
pixelhop2 transform
(3500, 28, 28, 42) (3500, 10, 10, 274) (3500, 1, 1, 528)
Accuracy on 3500 test images: 42.771 %
Time - 17 sec
------- TESTING FINISHED -------
```

Figure 3. Accuracy under different Ns Selection

Following Ns1, Ns2, Ns3 are selected: (16464, 13700, 528) (Top 50%), (2000, 1600, 528), (1200, 1000, 528), (1000, 1000, 528).

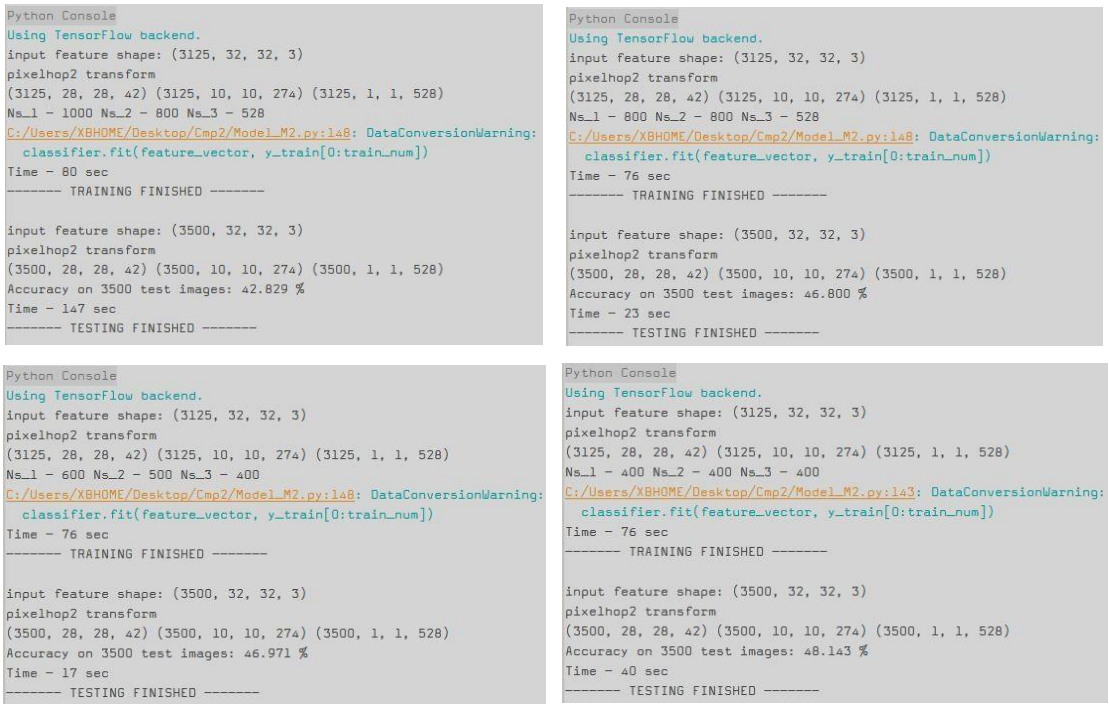


Figure 4. Accuracy under different Ns Selection

Following Ns1, Ns2, Ns3 are selected: (1000, 800, 528), (800, 800, 528), (600, 500, 400), (400, 400, 400).

It can be found that the **less** feature dimensions are kept, the higher the testing accuracy can be.

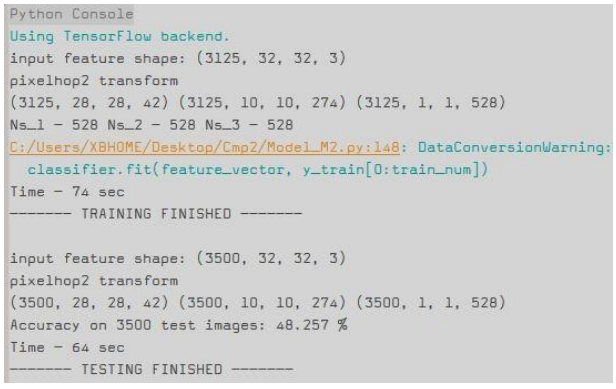
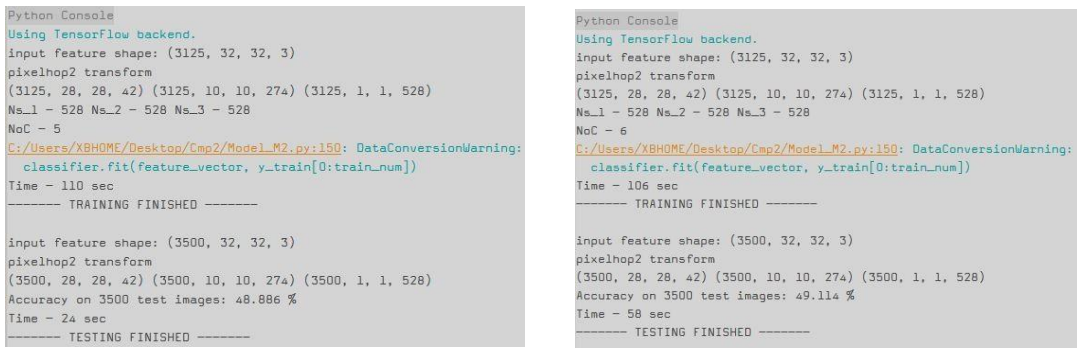
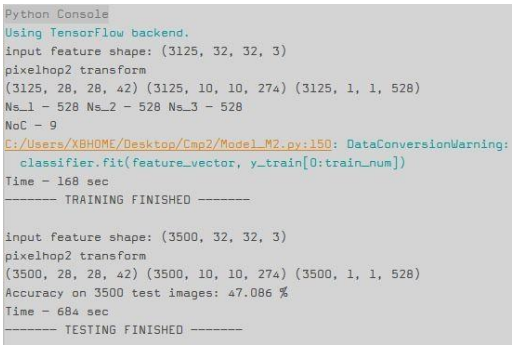


Figure 5. Best Ns Selection (528, 528, 528)

Selection Number of centroids in LAG





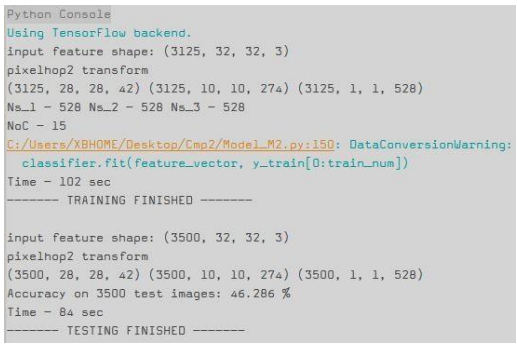


Figure 6. Accuracy under different Number of centroids Selection

Under the same Ns selection, 5, 6, 9, 15 centroids are tested, and 6 centroids reach highest accuracy.

3. Performance improvement

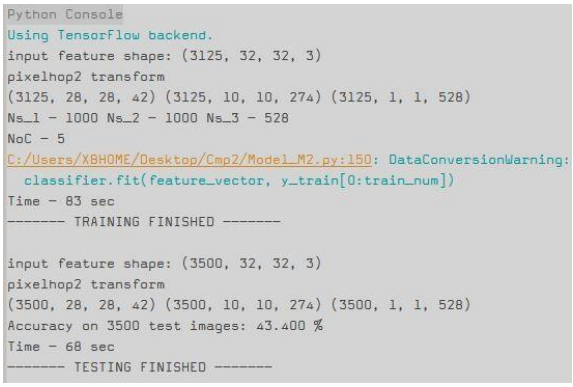
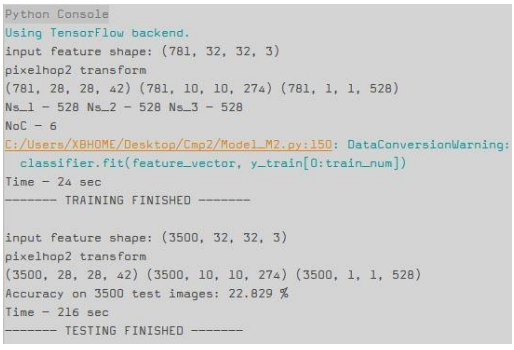
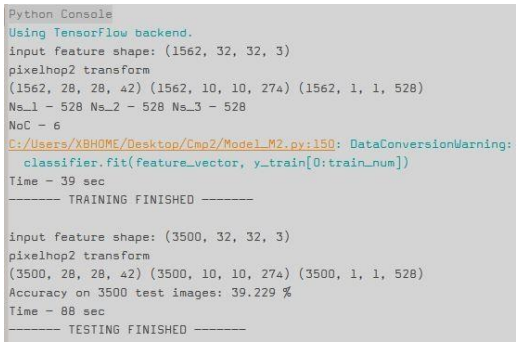


Figure 7. Reference Accuracy

Compared with reference testing accuracy (43.4%) in Problem 2(a), setting all the Ns with 528 dimensions and 6 centroids can get a higher accuracy (49.114% in Fig.6).

II. Classification accuracy with full and weak supervision





```
Python Console
Using TensorFlow backend.
input feature shape: (2343, 32, 32, 3)
pixelhop2 transform
(2343, 28, 28, 42) (2343, 10, 10, 274) (2343, 1, 1, 528)
Ns_1 - 528 Ns_2 - 528 Ns_3 - 528
NoC - 6
C:/Users/XBHOME/Desktop/Cmp2/Model_M2.py:150: DataConversionWarning:
  classifier.fit(feature_vector, y_train[0:train_num])
Time - 52 sec
------- TRAINING FINISHED -------

input feature shape: (3500, 32, 32, 3)
pixelhop2 transform
(3500, 28, 28, 42) (3500, 10, 10, 274) (3500, 1, 1, 528)
Accuracy on 3500 test images: 45.886 %
Time - 346 sec
------- TESTING FINISHED -------
```

```
Python Console
Using TensorFlow backend.
input feature shape: (3125, 32, 32, 3)
pixelhop2 transform
(3125, 28, 28, 42) (3125, 10, 10, 274) (3125, 1, 1, 528)
Ns_1 - 528 Ns_2 - 528 Ns_3 - 528
NoC - 6
C:/Users/XBHOME/Desktop/Cmp2/Model_M2.py:150: DataConversionWarning:
  classifier.fit(feature_vector, y_train[0:train_num])
Time - 106 sec
------- TRAINING FINISHED -------

input feature shape: (3500, 32, 32, 3)
pixelhop2 transform
(3500, 28, 28, 42) (3500, 10, 10, 274) (3500, 1, 1, 528)
Accuracy on 3500 test images: 49.114 %
Time - 58 sec
------- TESTING FINISHED -------
```

Figure 8. Testing Accuracy under different training image amount (781, 1562, 2343, 3125)

```
Python Console
Using TensorFlow backend.
input feature shape: (3500, 32, 32, 3)
pixelhop2 transform
(3500, 28, 28, 42) (3500, 10, 10, 274) (3500, 1, 1, 528)
Ns_1 - 528 Ns_2 - 528 Ns_3 - 528
NoC - 6
C:/Users/XBHOME/Desktop/Cmp2/Model_M2.py:150: DataConversionWarning:
  classifier.fit(feature_vector, y_train[0:train_num])
Time - 276 sec
------- TRAINING FINISHED -------

input feature shape: (3500, 32, 32, 3)
pixelhop2 transform
(3500, 28, 28, 42) (3500, 10, 10, 274) (3500, 1, 1, 528)
Accuracy on 3500 test images: 50.057 %
Time - 306 sec
------- TESTING FINISHED -------
```

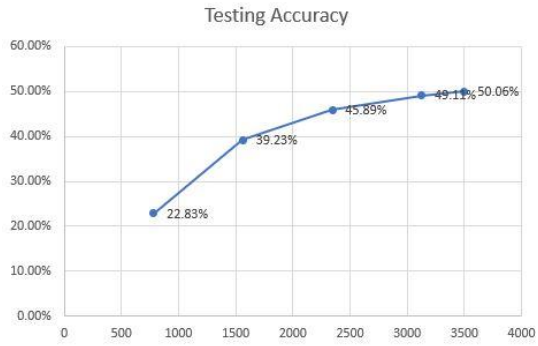


Figure 9. Highest Testing Accuracy (50.06% under 3500 training images) and Accuracy Curve

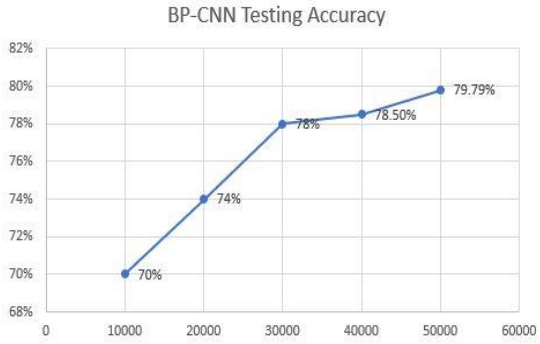


Figure 10. BP-CNN Training Accuracy Curve under different training image amount (10000, 20000, 30000, 40000, 50000)

Given different training images amount are used in current SSL model and former BP-CNN model, it is hard to compare. It can be found that the SSL model's testing accuracy began to converge to 50% (showing low slope) under only 3000 training images, while the BP-CNN model began to converge to 80% under 30000 training image.

III. Running time

System Configuration

Processor: Intel Core i5-9400 CPU @ 2.90 GHz

RAM: 8.00GB

OS: Windows 10

For the best training accuracy (50.06% for 3500 testing images) under 3500 training images, the training time is 276 sec for Module 2 and the inference time is 306 sec.

(However increasing either the amount of training or testing images will cause the local system crashing.)

IV. Model size

A. parameters of Saab filters: $K_1 = 42, K_2 = 274, K_3 = 528$.

Number of parameters: $(5 \times 5 \times 1) \times (42 + 274 + 528) = 21100$.

B. Regression matrix in LAG units: $60 \times (fearure_{dimension} + 1)$

Number of parameters: $3 \times 60 \times (528 + 1) = 95220$.

C. Classifier: $3\ Hops, M = 60, 10\ classes$.

Number of parameters: $(3 \times 60) \times 10 = 1800$.

The total parameter number is **118120**.

Compare to 395006 parameters in BP-CNN, This SSL model has a much smaller size.