USC Viterbi
School of Engineering

DEPARTMENT
Electrical and Computer Engineering

EE 569

Homework #6:
Successive Subspace Learning Competition with CIFAR-10 Classification

Name: Zhiwei Deng
Date: 5/02/2020

Contents

1. PROBLEM 1 .... **3**

1.1 FEEDFORWARD-DESIGNED CONVOLUTIONAL NEURAL NETWORKS (FF-CNNS) .... **ERROR! BOOKMARK NOT DEFINED.**

1.1.1 Saab Transform Summary .... ***Error! Bookmark not defined.***

1.1.2 Comparison between FF-CNN and BP-CNN .... ***Error! Bookmark not defined.***

1.2 SUCCESSIVE SUBSPACE LEARNING (SSL) .... **ERROR! BOOKMARK NOT DEFINED.**

1.2.1 SSL Methodology and Comparison with DL .... ***Error! Bookmark not defined.***

1.2.2 Modules Functions *16*

1.2.3 Neighborhood Construction and Subspace Approximation .... ***Error! Bookmark not defined.***

1.2.4 Label-Assisted Regression (LAG) .... ***Error! Bookmark not defined.***

2. PROBLEM 2 .... ERROR! BOOKMARK NOT DEFINED.

2.1 BUILDING A PIXELHOP++ MODEL .... **ERROR! BOOKMARK NOT DEFINED.**

2.1.1 Abstract and Motivation .... ***Error! Bookmark not defined.***

2.1.2 Approach and Procedures .... ***Error! Bookmark not defined.***

2.1.3 Experimental Result .... ***Error! Bookmark not defined.***

2.1.4 Discussion .... ***Error! Bookmark not defined.***

2.2 ERROR ANALYSIS .... **ERROR! BOOKMARK NOT DEFINED.**

2.2.1 Abstract and Motivation .... ***Error! Bookmark not defined.***

2.2.2 Approach and Procedures .... ***Error! Bookmark not defined.***

2.2.3 Experimental Result .... ***Error! Bookmark not defined.***

2.2.4 Discussion .... ***Error! Bookmark not defined.***

REFERENCES 19

1. Problem 3

1.1 Motivation and Logics

1.1.1 Motivation

Smaller Neighborhood Construction: Inspired by the VGG network, which uses multiple smaller kernels to simulate the performances of large kernels. Two smaller neighborhood construction PixelHop++ models are constructed. While since the window size are smaller, the depth model should be increased to extract as many features as possible. The first motivation of this proposal is that smaller windows like 3 by 3can keep more local information of one image than a large window like 5 by 5, these local information pieces might construct a more robust embedding for the image classification. On the other hand, this process can create more features than 5 by 5 window, which means it can extract more information than large window size. Also, smaller window sizes can help to reduce the model size of each PixelHop unit.

Prevent Overfitting: As stated above, the smaller window sizes provide a more informative model with more features. Which means it is harder for the classifier in Module 3 to prevent overfitting problem. Also, computation complexities of Saab transform, regression and feature selection are much heavier with too many features. So, besides the Saab transform, it’s necessary to reduce the feature numbers of the PixelHop++ unit. Firstly, I increase the threshold of the PixelHop++ units to filter more features out. Another way to do that is to decrease the Ns number respective to the training sample numbers. Furthermore, these two methods can also reduce the model size of the SSL system. Although since the SSL model is deeper, the parameters number is inevitable to be larger.

Number of Centroids: Numbers of centroid of each class represents the number of clusters of every label. As discussed in HW6, some classes in CIFAR-10 can be very misleading like cat-dog, bird-airplane etc. These misleading images share some similar features which might make the model harder to distinguish from each other. If the number of centroids of one class is too few, which means the samples are divided only into several clusters, might make the centroids of different classes too close. Which will lower the accuracy of the model. However, if this value is too large, which means the samples are divided into many clusters, might make the parameter number increases a lot and damage the efficiency. Also, the model size will increase rapidly with the number of centroids. So, in this assignment, I increase every classes' number of centroids in LAG process to 10. By the analysis of HW6, numbers of centroids of classes of 'cat', 'bird', 'dog', 'deer' is increased to try to eliminate the confusion and try to construct a more precise model by improving the LAG process.

Image Padding: From the original PixelHop++ model, we can see that the input images are not padded, they are sent to the model directly with 32 by 32 scale. The 2-pixel wide boundary pixels are only scanned once by the neighborhood construction process, while other pixels are scanned 3 times. To prevent losing this boundary information, we can use a reflection padding to enlarge the input images with 1-pixel width since using the 3 by 3 window size. Then the input image shape changes from (32, 32, 3) to (34, 34 ,3).

Classifier: In Module 3, SSL uses common Machin Learning methods to classify the LAG features. In HW6 Problem 2, the Random Forest is recommended to use. However, by the error analysis we can see that the images' resolution of CIFAR-10 is very low, which means the noises of the whole dataset is much higher than some high-resolution dataset. While the experience of RF tuning proves that the RF is easily to be overfitted when the data contains high level of noises. So, it might be wise to use SVM to replace the Random Forest. SVM is more robust since it can filter some redundant information of the features and make the decisions by the key vectors. However, the computational complexity of SVM is much higher than RF, which means the training time should be longer.

1.1.2 Logics

Based on the motivations stated above, I propose two larger SSL model with 4 PixelHop++ units and 5 PixelHop++ units, respectively. The model's diagram of them are shown as below in Figure 1 and Figure 2. For the 4 PixelHop++ unit model, the first two units' window size is 5, and the last two units' window size is 3. For the 5-PixelHop++ Unit Model, all PixelHop Units' window sizes are 3 by 3. For the both models, the max-pooling operations are only applied on the first two PixelHop++ units, since the dimensions of the first two units are much larger than the later two. It is necessary to do the dimension reduction by the max-pooling. Also, this arrangement is suitable for the model to make the widths of every inputs of each unit are all even.

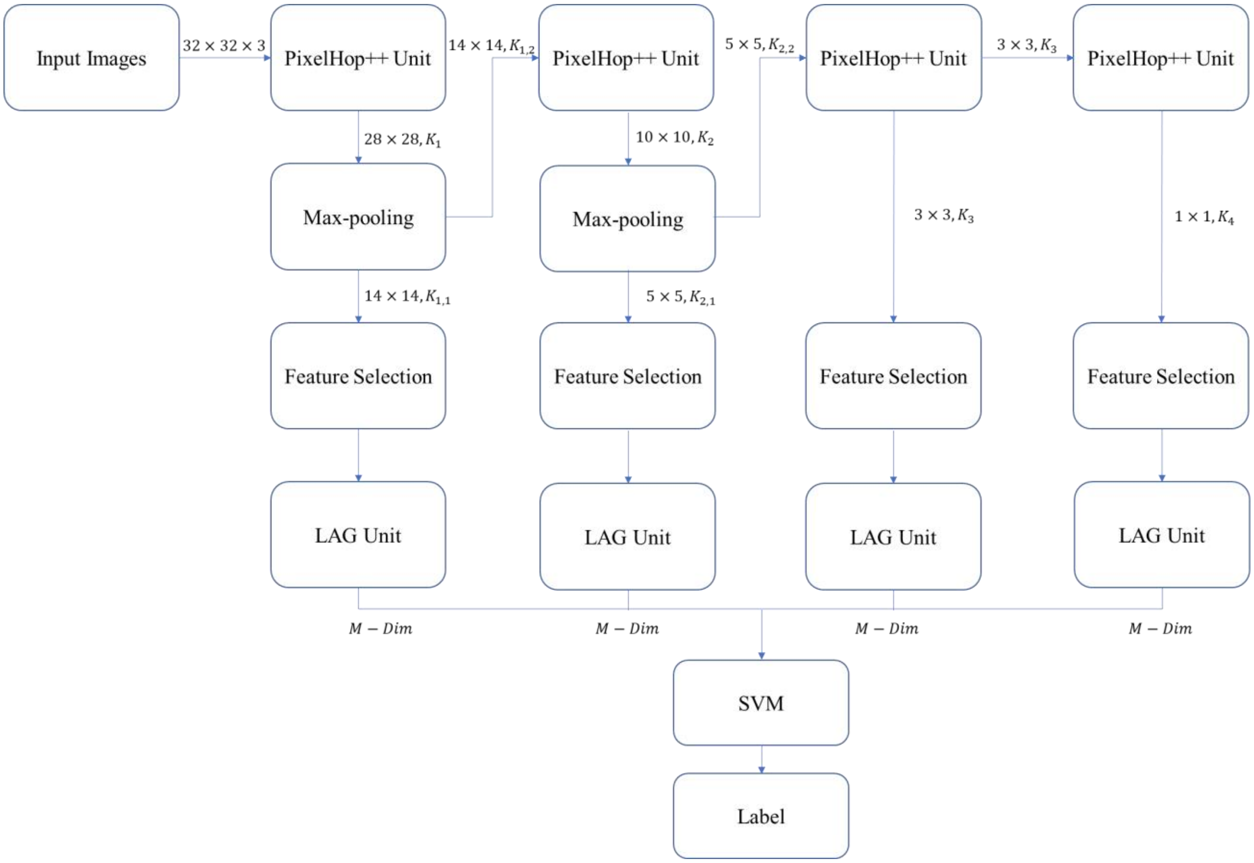


Figure 1. 4-PixelHop++Unit Model Architecture.

As to the hyper-parameters of to models are shown as below in Table 1.

Table 1. Hyper-parameters of Each Model

Hyper-parameter	**4-PixelHop++ Units**	**5-PixelHop++ Units**
Spatial Neighborhood size	(5,5) and (3,3)	(3,3)
Stride	1	1
Max-pooling	(2x2) -to- (1x1)	(2x2) -to- (1x1)
TH1, TH2	0.0012, 0.00012	0.0012, 0.00012
Number of selected features	1000, 250	1000, 250
α in LAG units	10	10
Number of centroids per class	10, 5	10
Classifier	SVM	SVM

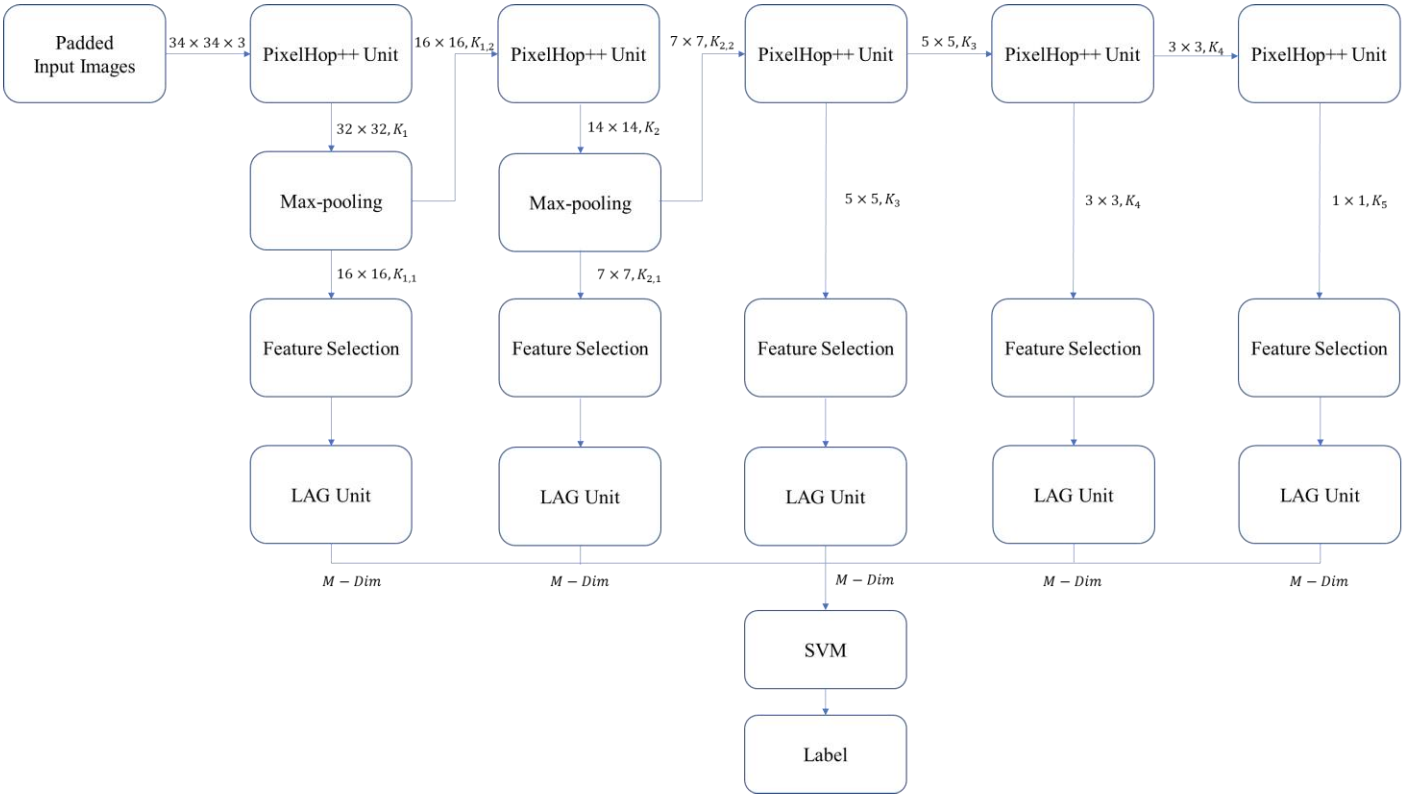


Figure 2. 5-PixelHop++Unit Model Architecture.

Source of Improvements: From the analysis above, we can conclude several reasons for the improvements of new models. Firstly, the two models' architectures are deeper than the baseline in HW6, which means that more features are extracted by PixelHop units. More specifically, for 5-PixelHop++ Unit model, the features dimension of each unit is 4864, 4606, 6275, 2970, 304 respectively with TH1=0.01, TH2=0.0001. While in the original model, the number is 8232. 6850, 530 respectively. We can see that the new model extracted more than 3000 features than the original model. These features can be used to strengthen the classifier to do a better job. Furthermore, for 5-PixelHop++ Unit model, it considers the boundary pixels' information as same important as the center pixels.

Secondly, the number of centroids of certain classes has been increased, which means the unsupervised clustering should be more precise than the original model. For example, some clusters of cat and dog classes are very close in high dimensional space, and the original model cannot separate one from another since the number of centroids are limited. However, for the new models, cat and dog samples can have more centroids to form more clusters, which gives them more flexibility of distinguishing them with each other. What need to mention is that only the 4 most confusing classes in the 4-PixelHop++ model is increased, which are 'cat', 'dog', 'bird' and 'deer'.

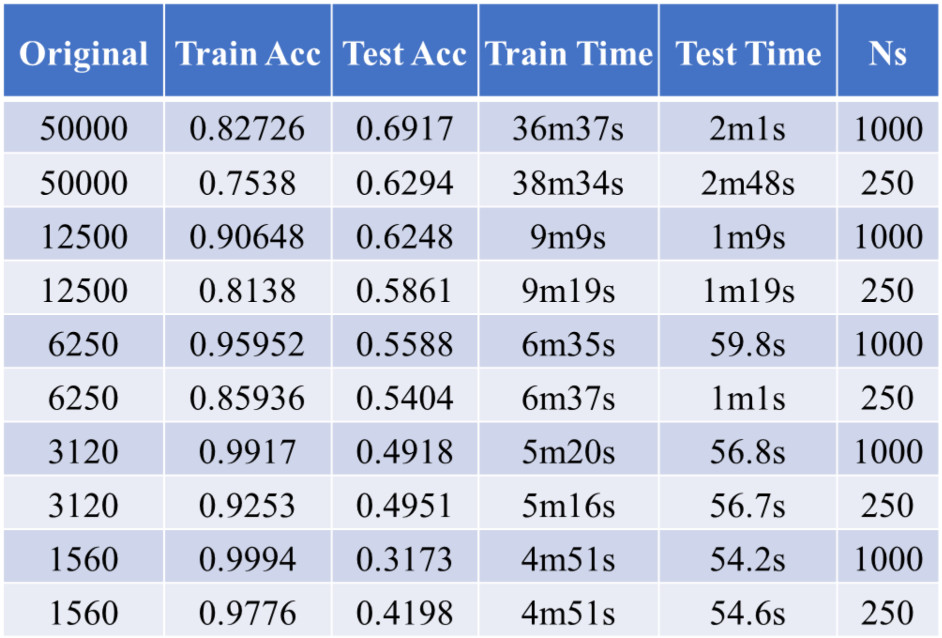
Thirdly, the SVM should be more robust than RF to the noisy data like low resolution images in CIFAR-10 dataset. The detailed comparisons between RF and SVM are going to be illustrated in the next section.

1.2 Classification Accuracy and Model Size

1.2.1 Accuracy

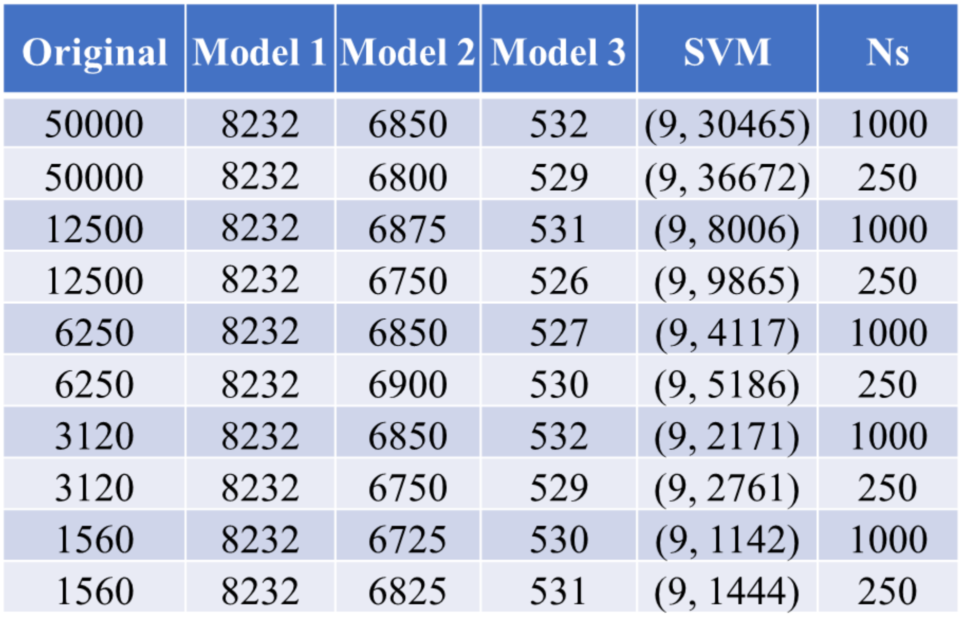
The original model's performance is shown as below in Table 2 and 3.

Table 2. Original Model's Accuracy with Different Ns



Original	Train Acc	Test Acc	Train Time	Test Time	Ns
50000	0.82726	0.6917	36m37s	2m1s	1000
50000	0.7538	0.6294	38m34s	2m48s	250
12500	0.90648	0.6248	9m9s	1m9s	1000
12500	0.8138	0.5861	9m19s	1m19s	250
6250	0.95952	0.5588	6m35s	59.8s	1000
6250	0.85936	0.5404	6m37s	1m1s	250
3120	0.9917	0.4918	5m20s	56.8s	1000
3120	0.9253	0.4951	5m16s	56.7s	250
1560	0.9994	0.3173	4m51s	54.2s	1000
1560	0.9776	0.4198	4m51s	54.6s	250

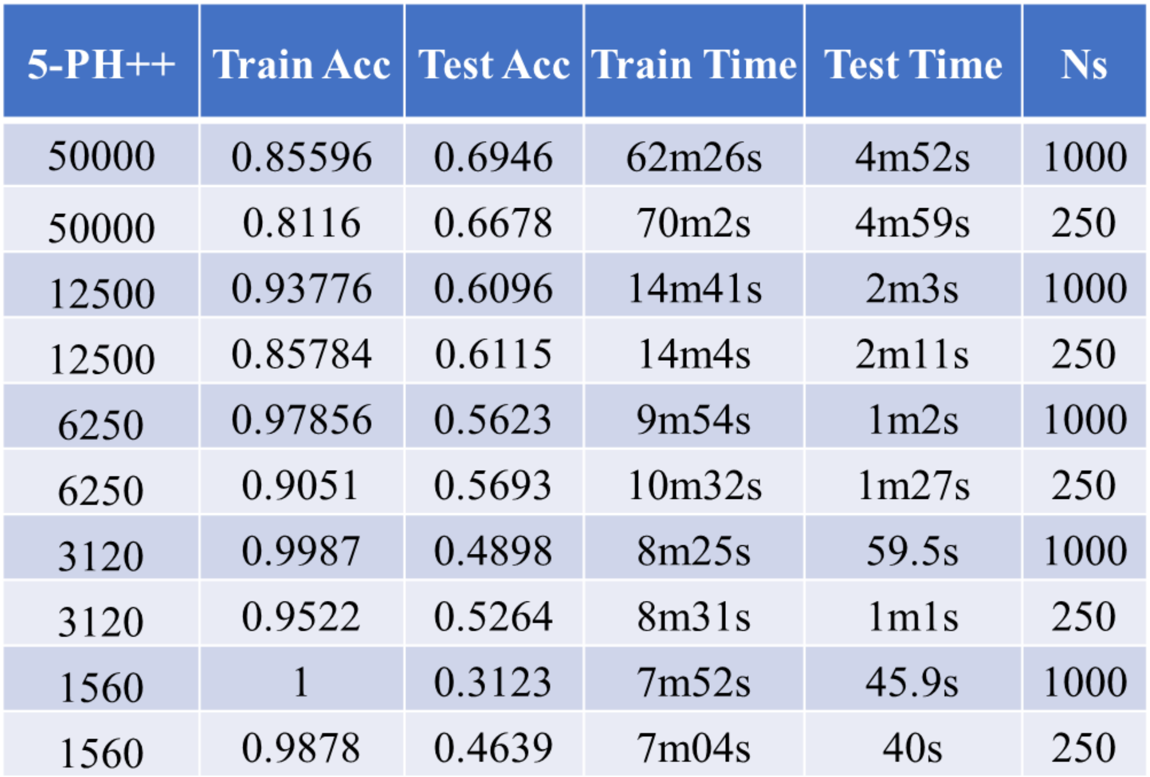
Table 3. Original Model's Model Size with Different Ns



Original	Model 1	Model 2	Model 3	SVM	Ns
50000	8232	6850	532	(9, 30465)	1000
50000	8232	6800	529	(9, 36672)	250
12500	8232	6875	531	(9, 8006)	1000
12500	8232	6750	526	(9, 9865)	250
6250	8232	6850	527	(9, 4117)	1000
6250	8232	6900	530	(9, 5186)	250
3120	8232	6850	532	(9, 2171)	1000
3120	8232	6750	529	(9, 2761)	250
1560	8232	6725	530	(9, 1142)	1000
1560	8232	6825	531	(9, 1444)	250

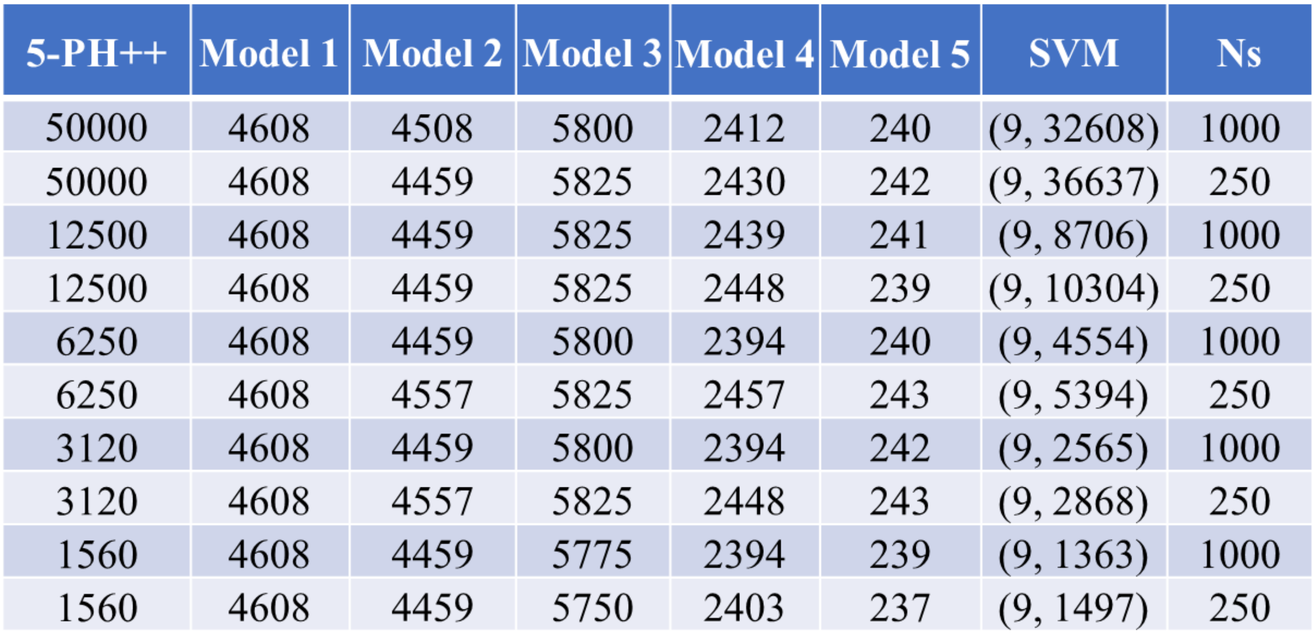
The 5-PH model's performance is shown as below in Table 4 and 5.

Table 4. 5-PH Model's Accuracy with Different Ns



5-PH++	Train Acc	Test Acc	Train Time	Test Time	Ns
50000	0.85596	0.6946	62m26s	4m52s	1000
50000	0.8116	0.6678	70m2s	4m59s	250
12500	0.93776	0.6096	14m41s	2m3s	1000
12500	0.85784	0.6115	14m4s	2m11s	250
6250	0.97856	0.5623	9m54s	1m2s	1000
6250	0.9051	0.5693	10m32s	1m27s	250
3120	0.9987	0.4898	8m25s	59.5s	1000
3120	0.9522	0.5264	8m31s	1m1s	250
1560	1	0.3123	7m52s	45.9s	1000
1560	0.9878	0.4639	7m04s	40s	250

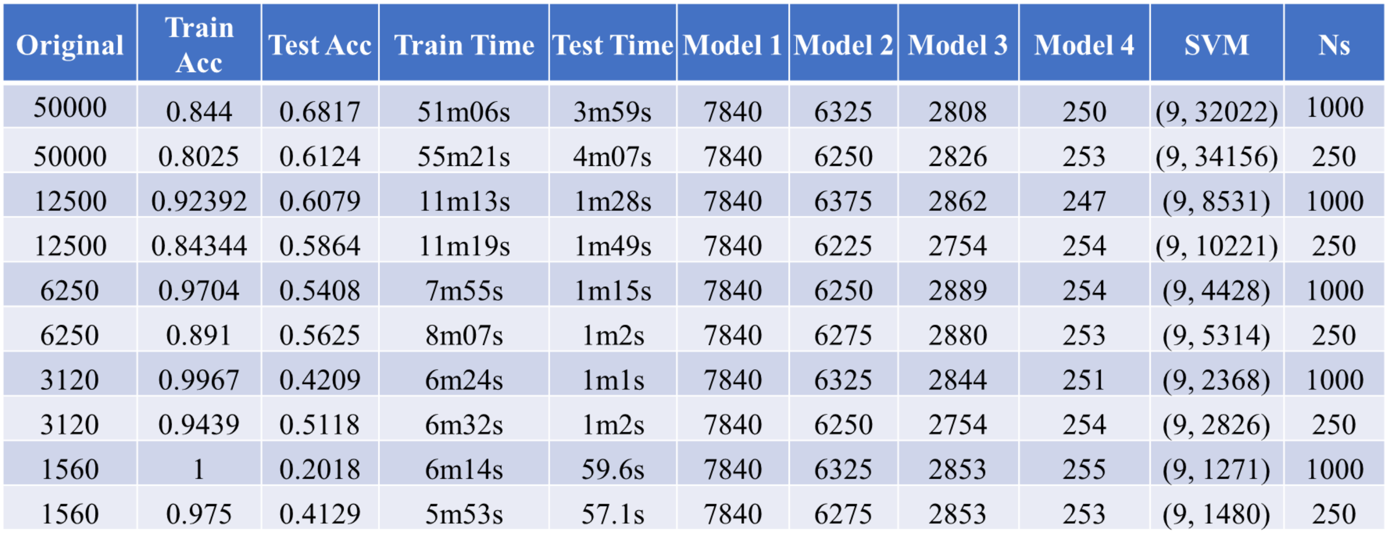
Table 5. 5-PH Model's Model Size with Different Ns



5-PH++	Model 1	Model 2	Model 3	Model 4	Model 5	SVM	Ns
50000	4608	4508	5800	2412	240	(9, 32608)	1000
50000	4608	4459	5825	2430	242	(9, 36637)	250
12500	4608	4459	5825	2439	241	(9, 8706)	1000
12500	4608	4459	5825	2448	239	(9, 10304)	250
6250	4608	4459	5800	2394	240	(9, 4554)	1000
6250	4608	4557	5825	2457	243	(9, 5394)	250
3120	4608	4459	5800	2394	242	(9, 2565)	1000
3120	4608	4557	5825	2448	243	(9, 2868)	250
1560	4608	4459	5775	2394	239	(9, 1363)	1000
1560	4608	4459	5750	2403	237	(9, 1497)	250

The 4-PH model’s performances and model sizes are shown as below in Table 6.

Table 6. 4-PH Model’s Accuracy and Sizes with Different Ns



Original	Train Acc	Test Acc	Train Time	Test Time	Model 1	Model 2	Model 3	Model 4	SVM	Ns
50000	0.844	0.6817	51m06s	3m59s	7840	6325	2808	250	(9, 32022)	1000
50000	0.8025	0.6124	55m21s	4m07s	7840	6250	2826	253	(9, 34156)	250
12500	0.92392	0.6079	11m13s	1m28s	7840	6375	2862	247	(9, 8531)	1000
12500	0.84344	0.5864	11m19s	1m49s	7840	6225	2754	254	(9, 10221)	250
6250	0.9704	0.5408	7m55s	1m15s	7840	6250	2889	254	(9, 4428)	1000
6250	0.891	0.5625	8m07s	1m2s	7840	6275	2880	253	(9, 5314)	250
3120	0.9967	0.4209	6m24s	1m1s	7840	6325	2844	251	(9, 2368)	1000
3120	0.9439	0.5118	6m32s	1m2s	7840	6250	2754	254	(9, 2826)	250
1560	1	0.2018	6m14s	59.6s	7840	6325	2853	255	(9, 1271)	1000
1560	0.975	0.4129	5m53s	57.1s	7840	6275	2853	253	(9, 1480)	250

Since the accuracies and model sizes of 4-PixelHop++ Unit model are both under the performances of the original model. So, the follow comparisons and analysis process are only between 5-PH model and the original model.

Although from Table 4 and 2, we can only see a slightly improvement on test accuracy. But another thing needs to mention is that the training accuracy is higher. Which means the whole model fits the data better. This is due to the increase of the number of centroids. An image can use a higher dimensional vector to represent itself. It gives the classifier more flexibility to fit the data more precisely. Also, based on the tables above, I think the overfitting problem is generally existed in SSL models. So, reducing the features dimensions might be helpful. But I don’ have enough time for fine tuning the parameters of each model. It can be regarded as a future work.

Comparison: From the Tables above, we can see that the accuracy of 5-PH model is generally better than the original model, especially when the selected features are few. And it can reach the highest testing accuray in all of these three models.

The weak supervision accuracy comparisons between the original model and the 5-PH model are shown as below Figure 3 and Figure 4 with Ns =250 and 1000 respectively.

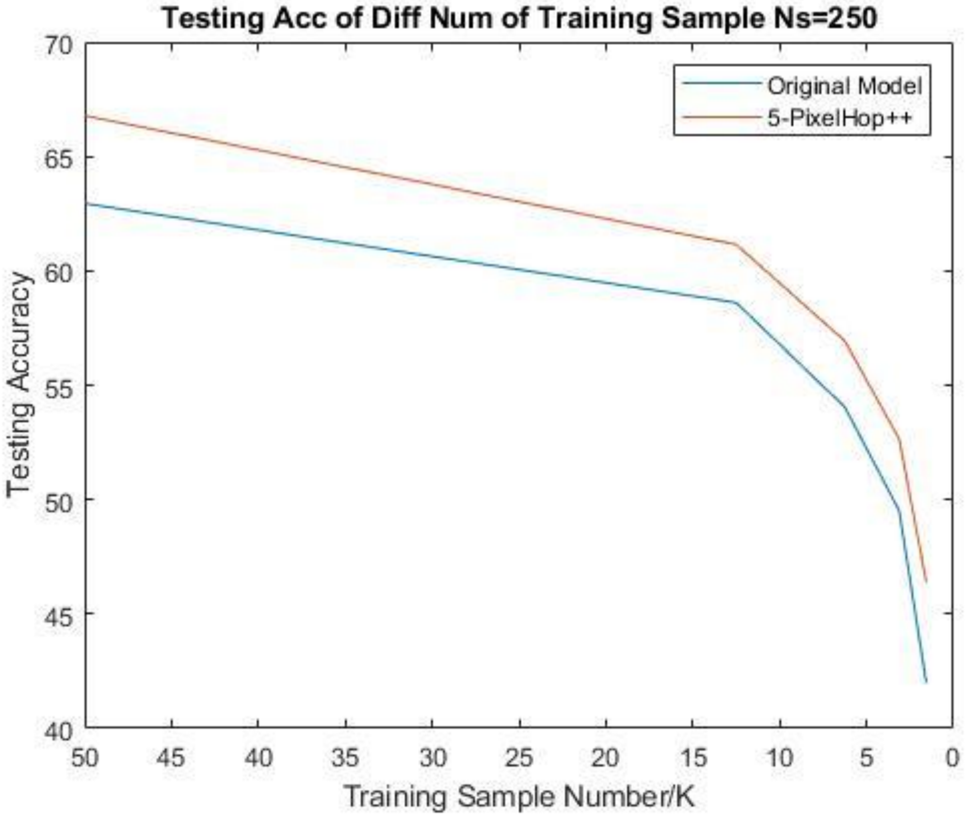


Figure 3. Accuracy of original model and the 5-PH Ns=250

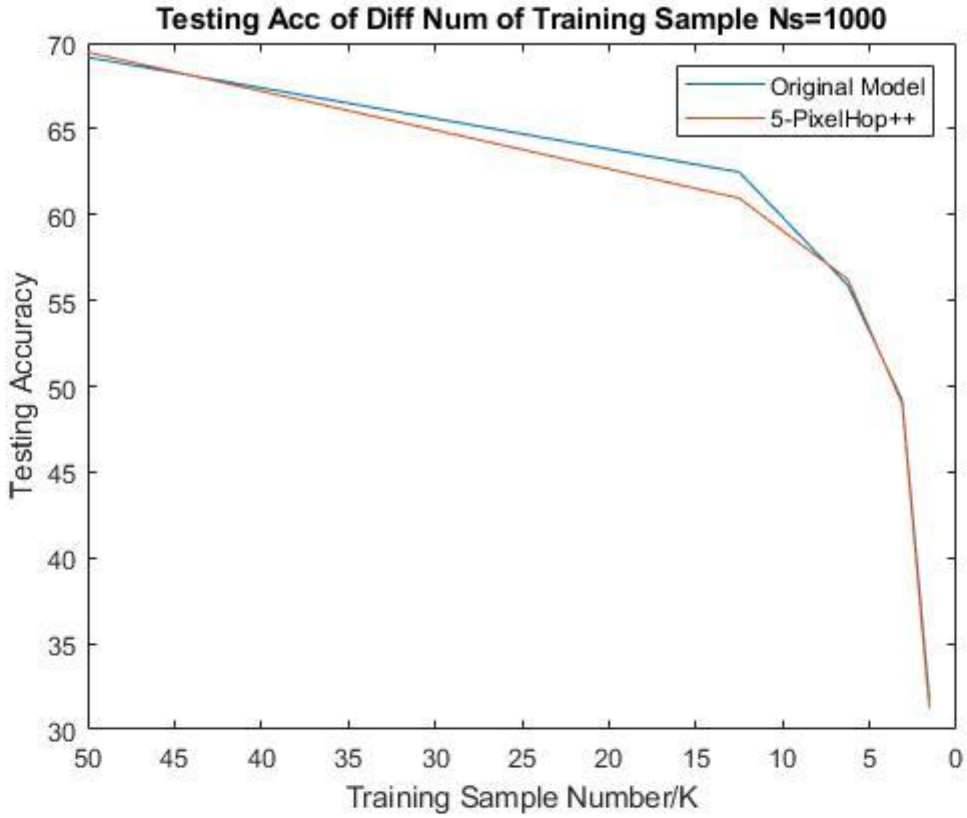
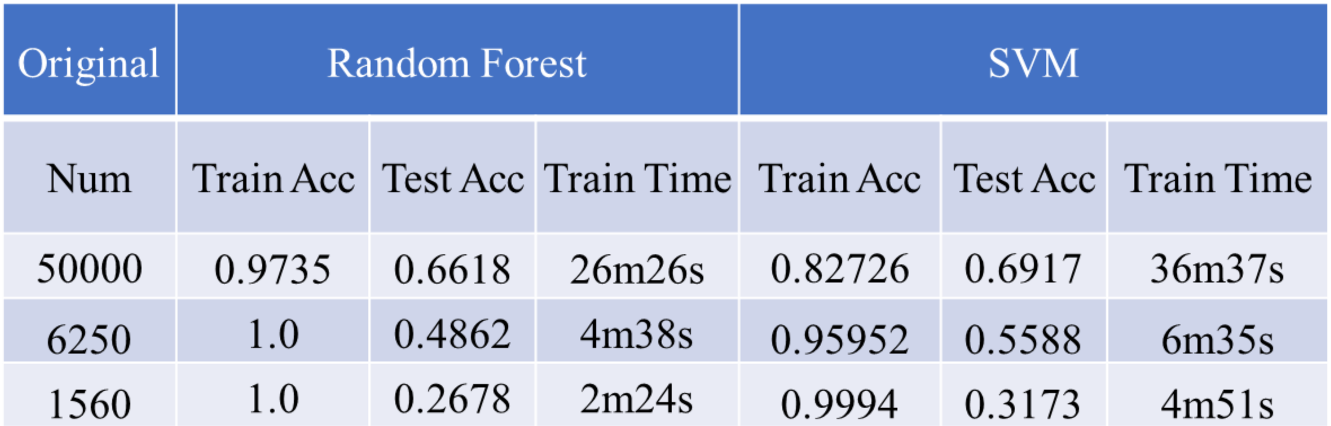


Figure 4. Accuracy of original model and the 5-PH Ns=1000

From Figure 3 and Figure 4, we can see that the accuracies between 5-PH and the original model are quiet similar wheng the Ns is large. But 5-PH can reach a slightly higher best performance with the whole dataset training. However, the advances of 5-PH is obvious when the selected features are fewer. From Figure 3, we can see that when the Ns is small, since 5-PH contains more layers, which makes it can give more information about the images. Thus, the accuracy of the model would not drop drasticly. From the graph above, we can see that the robustness of 5-PH is better than the original model since it can make more precise prediction with fewer image features. But this improvement sacifices the running time, from the table details we can see that the model of 5-PH takes longer time to train than the original model. But with smaller trianing datasets, the training time of two models are closer, and 5-PH model performas better when the seleceted features are few. So, 5-PH model might be more suitable for the CV tasks with smaller datasets and featureless images, like texture classification (might be).

For the classifier, the accuracies of the original model using the RF and SVM as the classifier are shown as the Table 7 below.

Table 7. Random Forest vs SVM



Original	Random Forest			SVM		
Num	Train Acc	Test Acc	Train Time	Train Acc	Test Acc	Train Time
50000	0.9735	0.6618	26m26s	0.82726	0.6917	36m37s
6250	1.0	0.4862	4m38s	0.95952	0.5588	6m35s
1560	1.0	0.2678	2m24s	0.9994	0.3173	4m51s

From the table above, we can easily seen that RF is more time-efficient, however, the highest test accuracy is lower than SVM. Also, from the training accuracy data of Random forest, we can say that the classifier has been overfitted. So, although SVM is more time consuming, I still think it is much better to use it as classifier in Module 3 over RF.

Comparison with CNN: In HW5, I implemented a CNN with the weak supervision performance as Figure 6. The degradation of CNN and SSL model is quiet similar. They both lost about 40% testing accuracy when the training sample numbers decreased from 50k to 1.56k. However, the degradation of SSL might be more than CNN if the Ns is too large since the simple classfier of Module 3 is more likely to be overfitted than a fine-tuned CNN network. For other aspects, the training time on CPU of CNN is much larger than SSL model, which might take almost 22 hours for CNN to reach the accuracy shown below.

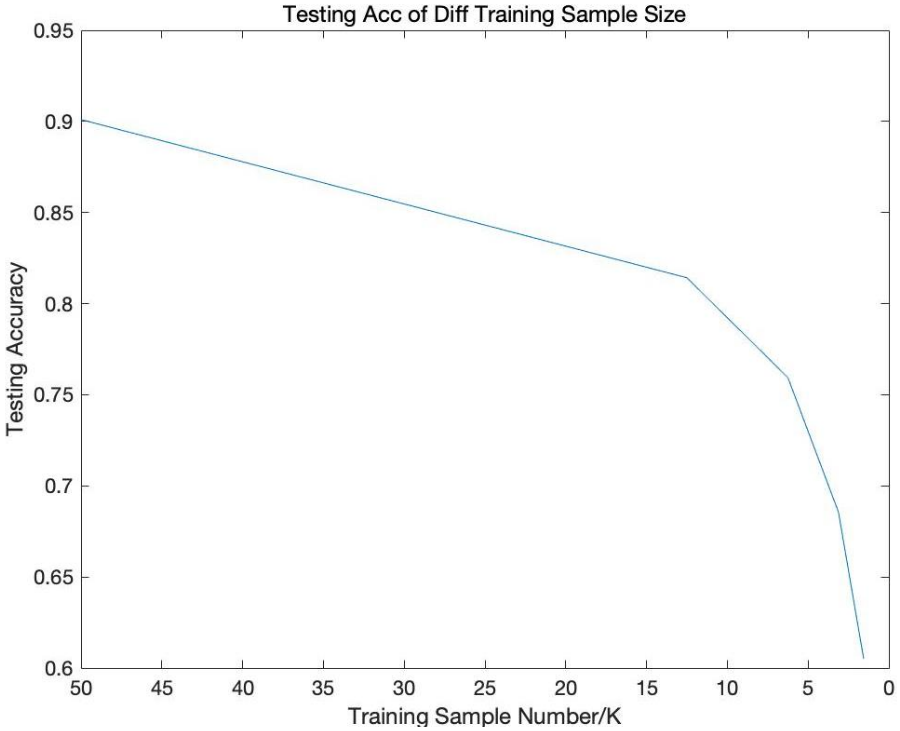


Figure 6. Weak Supervision of CNN in HW5 Problem 2

As the aspect of efficiency, the SSL model is much more efficient than CNN models, since it uses less paramters and can be excuted on CPU effectively. However, the CNN might have more flexibilities since the architecture are more independent. In conclusion, CNN is better than SSL in accuracy and flexibility since it is much more mature than SSL. But SSL is significantly more robust and efficient than CNN if the accuracy is not strictly required. And I think SSL can be improved by more tuning and modifications.

1.2.2 Model Size

The model sizes of each PixelHop++ system are shown as Table 8-9 below.

Table 8. Original PixelJop++ Model Parameters

Original	Parameter Num	Module Params	Without Classifier	Total
PixelHop++ Unit 1	3150	23175	149725	421143
PixelHop++ Unit 2	6825			
PixelHop++ Unit 3	13200			
LAG Units	126550	126550		
SVM	271418	271418	271818	

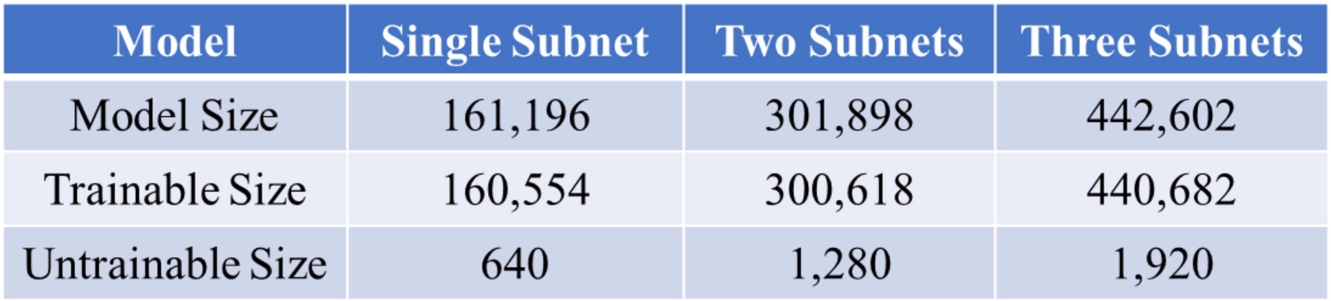
Table 9. 5-PixelHop++ Model Parameters

5-PH Model	Parameter Num	Module Params	Without Classifier	Total
PixelHop++ Unit 1	486	6345	430845	724317
PixelHop++ Unit 2	441			
PixelHop++ Unit 3	828			
PixelHop++ Unit 4	2430			
PixelHop++ Unit 5	2160			
LAG Units	424500	424500		
SVM	293472	293472	293472	

From Table 8 and 9, we can see that the new model's parameter is almost twice than the original one. But the parameters in module 1 is decrease while the parameters in module 2 is increased a lot. This is due to the small window size that used in Module 1 and the increments of the number of the centroids of each class, which will make the feature vector much longer.

Comparion with CNN: In HW5, the CNN's parameters number is about 440k as shown in Table 10. For the original PixelHop++ model, the parameter numbers are less than the CNN model with 3 subnet. However, this is achieved with no FC layers in CNN, which reduces the ntwork's parameters greatly. The parameters of convolutional layers are remained. So, in conclusion, the model size of 5-PixelHop++ is greater than the CNNs proposed in HW5 problem 2.

Tabel 10. CNN Model Sizes of 3 Models



Model	Single Subnet	Two Subnets	Three Subnets
Model Size	161,196	301,898	442,602
Trainable Size	160,554	300,618	440,682
Untrainable Size	640	1,280	1,920

Techinical Specs: PC Model—Macbook Pro 2018

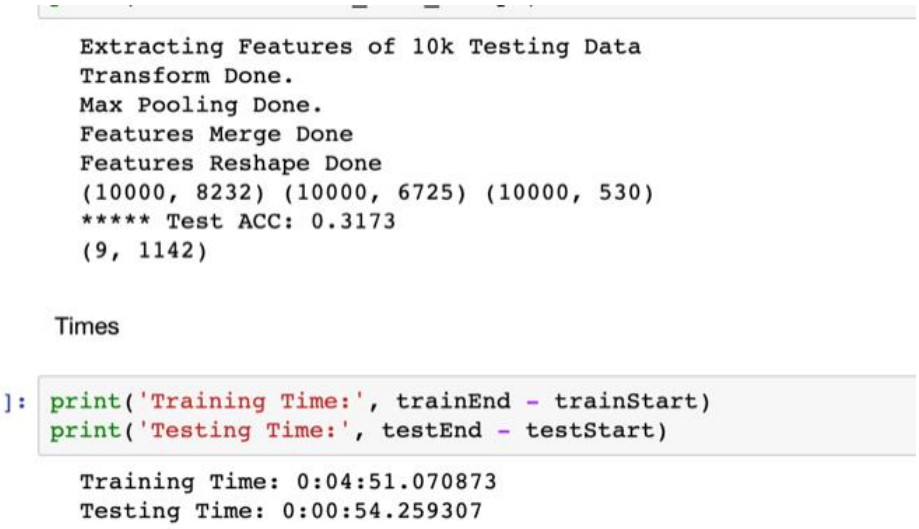
CPU—Intel Core-i5 with 2.3GHz

Core Num—4

RAM—16Gi

Running Time Prove Screenshots:

Original model Runing Time with Ns = 1000 and 1.56K Training Data



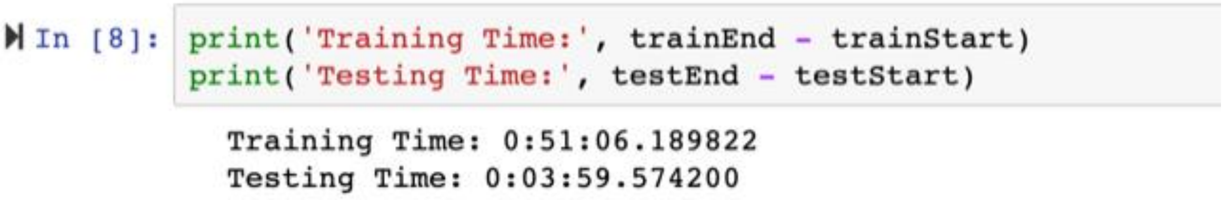
```
Extracting Features of 10k Testing Data
Transform Done.
Max Pooling Done.
Features Merge Done
Features Reshape Done
(10000, 8232) (10000, 6725) (10000, 530)
***** Test ACC: 0.3173
(9, 1142)
```

Times

```
]: print('Training Time:', trainEnd - trainStart)
   print('Testing Time:', testEnd - testStart)

Training Time: 0:04:51.070873
Testing Time: 0:00:54.259307
```

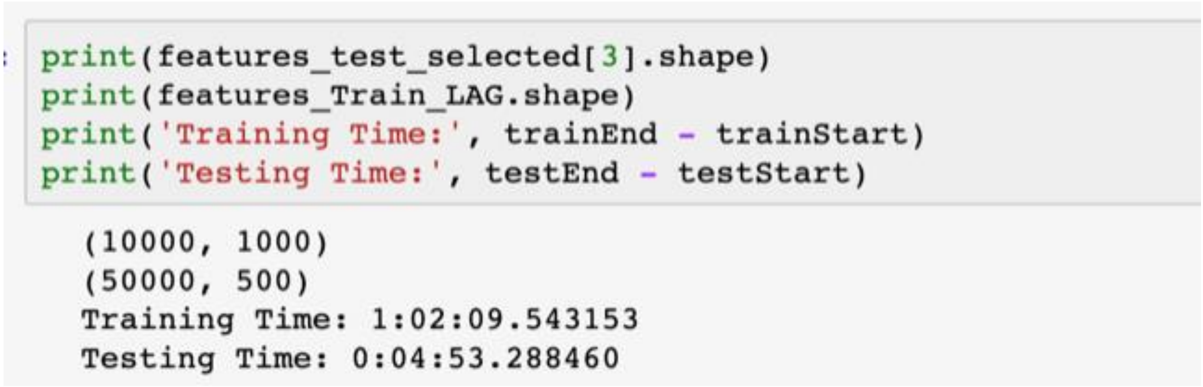
4-PH model Runing Time with Ns = 1000 and 50K Training Data



```
In [8]: print('Training Time:', trainEnd - trainStart)
        print('Testing Time:', testEnd - testStart)

Training Time: 0:51:06.189822
Testing Time: 0:03:59.574200
```

5-PH model Runing Time with Ns = 1000 and 50K Training Data



```
: print(features_test_selected[3].shape)
  print(features_Train_LAG.shape)
  print('Training Time:', trainEnd - trainStart)
  print('Testing Time:', testEnd - testStart)

(10000, 1000)
(50000, 500)
Training Time: 1:02:09.543153
Testing Time: 0:04:53.288460
```

References

[1] C.-C. Jay Kuo and Yueru Chen, "On data-driven Saak transform," Journal of Visual Communication and Image Representation, vol. 50, pp. 237–246, 2018.

[2] C-C Jay Kuo, Min Zhang, Siyang Li, Jiali Duan, and Yueru Chen, "Interpretable convolutional neural networks via feedforward design," Journal of Visual Communication and Image Representation, vol.60, pp. 346–359, 2019.

[3] Yueru Chen and C-C Jay Kuo, "Pixelhop: A successive subspace learning (ssl) method for object recognition," Journal of Visual Communication and Image Representation, p. 102749, 2020.

[4] Yueru Chen, Mozhdeh Rouhsedaghat, Suya You, Raghuveer Rao, C.-C. Jay Kuo, "PixelHop++: A Small Successive-Subspace-Learning-Based (SSL-based) Model for Image Classification," https://arxiv.org/abs/2002.03141, 2020

[5] Yueru Chen, Yijing Yang, Wei Wang, C.-C. Jay Kuo, "Ensembles of Feedforward-designed Convolutional Neural Networks", in International Conference on Image Processing, 2019