Samwoo Seong
7953-6137-66
samwoose@usc.edu

EE569 Midterm 2 (HW6 P3)
May 1, 2020

1.

1.1 Motivation and logics behind my design

SSL model with PixelHop++ method can extract features that represent each class. Each hyper parameter plays different roles in the model. Therefore, we can improve a quality of the extracted feature (e.g., discriminant power of each feature) by properly tuning hyper parameters. To do so, we need to carefully consider their roles. [1], [2]

(1) Spatial Neighborhood size in all PixelHop++ units: this decides how closely we want to observe the neighborhood surrounding a target pixel. Therefore, we can capture rich attributes of the target pixel by employing various spatial neighborhood size. In our model, we select 3x3 for 3 units and 5x5 for other 3 units to obtain rich representative intermediate features.

(2) Stride: Since we desire to capture all details from input images, we make use of the smallest stride size which is 1

(3) Max-pooling: Max-pooling operation provides us 2 significant roles. First, we can have control over spatial dimension and second, we can inherit the strongest representation from the previous unit or module in the SSL structure. Therefore, we operate our model with (2x2) -to- (1x1) max pooling method.

(4) Energy threshold for intermediate node (TH1): This hyper parameter has an impact on number of intermediate nodes which will be processed at the next process. Since it could cause high dimension of channel (K) if the value is too low, we need to find a value that is high enough to control the dimension of K (i.e., get rid of redundancy in attributes), and low enough to retain rich representations at the next unit or module. Therefore, we work with 0.001 as value for TH1 in our model.

(5) Energy threshold for discarded nodes (TH2): This contributes to number of discarded nodes that barely have energy. Proper value of TH2 should be large enough to remove nodes that will not make any contribution to extracting high quality features, but small enough not to throw away nodes that have high energy. Thus, we choose 0.0001 for TH2 in our model.

(6) Number of selected features (Ns) for each Hop: This creates a way to link extracted features without super vision and classes. We should select a value that can pick features highly related to classes we have. In our model, 50 % of features that have the lowest cross-entropy values.

(7) α in LAG units: As value of α gets bigger, probability of a sample being in a cluster decreases rapidly meaning we consider distance between the sample and centroids importantly. Therefore, we need to choose a value that cover large enough surrounding area for the cluster, but not to large so that the area does not invade other clusters. We select 10 for value of α in our model.

(8) Number of centroids per class in LAG: It has two major roles. First of all, it gives a way to represent features from the previous modules and second, it provides a control over final feature dimension. The dimension after LAG unit will have an impact on the next module which is basically classification task. Therefore, we should choose a number of centroids that is large enough represent the formal rich feature, but small enough to reduce a possibility of overfitting caused by too high dimension of feature space when we have a small number of data samples. Thus, we choose 7 centroids in our model.

(9) Classifier: We can use any machine learning classifier at this stage. However, we apply Random Forest because it is one of most popular machine learning algorithms since it has ability to prevent overfitting and high run speed in general. Also, we select 400 estimators in Random Forest because it makes our classifier not only run fast enough, but also, restrict overfitting situation.

Hyper parameter	Chosen Value or Method
Spatial Neighborhood size in PixelHop++ units	3x3 for 3 units and 5x5 for the rest of units
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 (Ns) for each Hop	Top 50%
α in LAG units	10
Number of centroids per class in LAG units	7
Classifier	Random Forest
Number of Estimators in Random Forest	400

Table1. Summary of chosen hyper parameters.

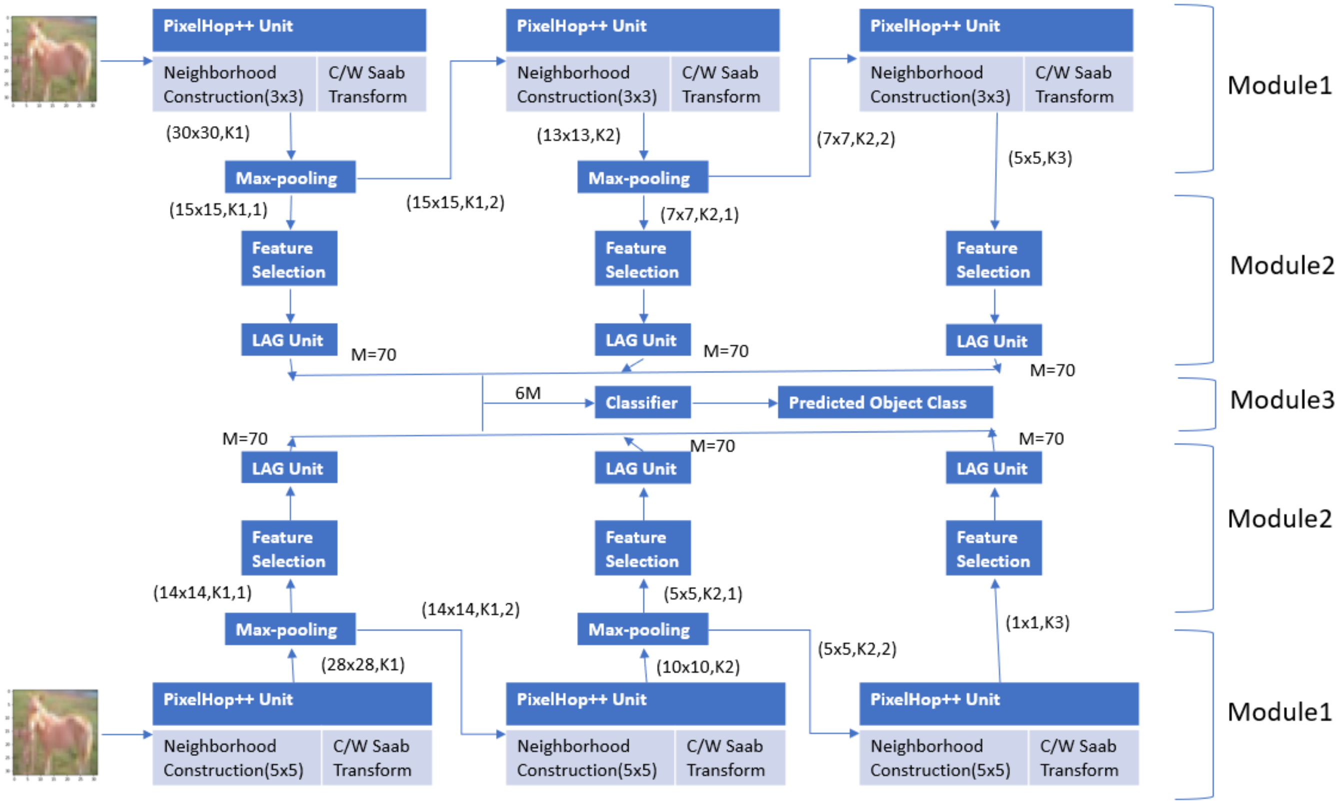


Figure 1. Architecture of SSL [1], [2], [3]

1.2 Experiment results

(0) Hardware Specifications

Memory	8GB RAM
Processor	CPU, Inter core i5

Table2. Hardware Specifications

(1-1) Best Accuracy

#NOTE

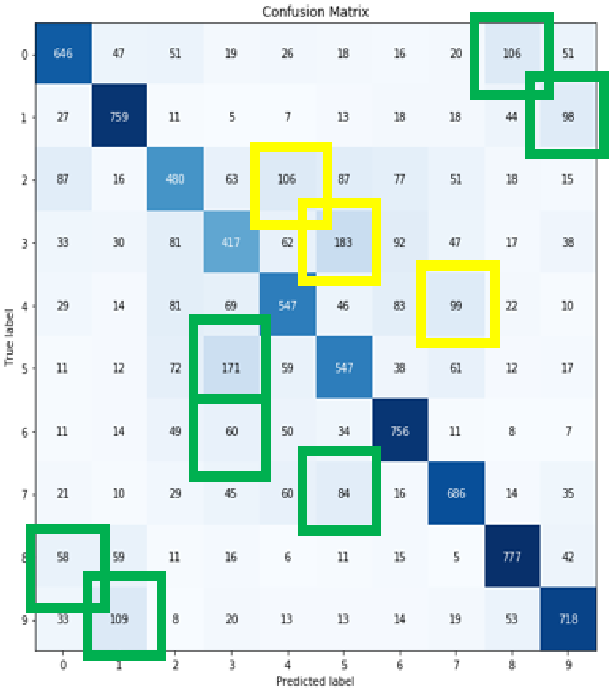
Since I was experiencing severe memory issues during pixelhop2.fit() and pixelhop2.transform(), maximum number of data samples are 6250 train images, and 20000 train images respectively. In the future work, I better use cloud web service for bigger number of data samples.

63.33% on test set

Note: Best accuracy on test set at HW6: 59.67 %

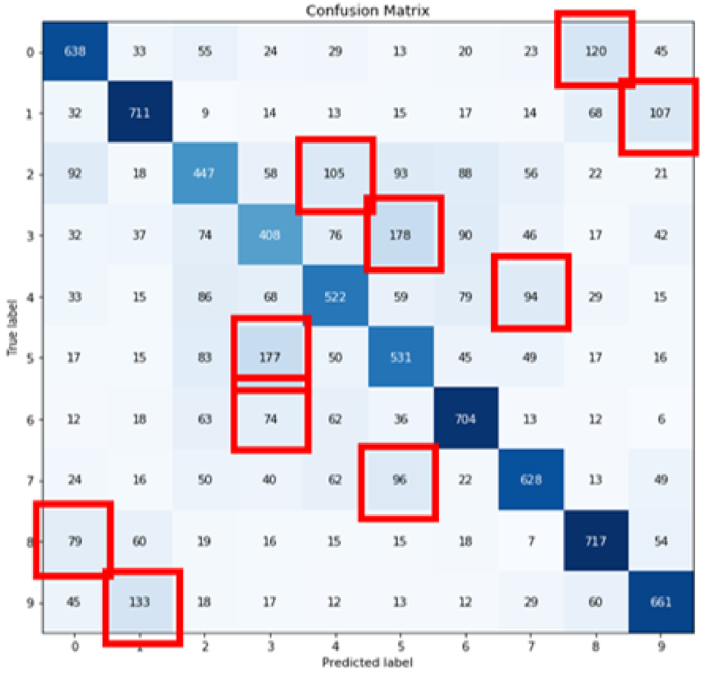
=>3.66% of improvement in terms of accuracy on test set

(1-2)



True label \ Predicted label	0	1	2	3	4	5	6	7	8	9
0	646	47	51	19	26	18	16	20	106	51
1	27	759	11	5	7	13	18	18	44	98
2	87	16	480	63	106	87	77	51	18	15
3	33	30	81	417	62	183	92	47	17	38
4	29	14	81	69	547	46	83	99	22	10
5	11	12	72	171	59	547	38	61	12	17
6	11	14	49	60	50	34	756	11	8	7
7	21	10	29	45	60	84	16	686	14	35
8	58	59	11	16	6	11	15	5	777	42
9	33	109	8	20	13	13	14	19	53	718

Figure 2-1. Confusion matrix in modified SSL model.



True label \ Predicted label	0	1	2	3	4	5	6	7	8	9
0	638	33	55	24	29	13	20	23	120	45
1	32	711	9	14	13	15	17	14	68	107
2	92	18	447	58	105	93	88	56	22	21
3	32	37	74	408	76	178	90	46	17	42
4	33	15	86	68	522	59	79	94	29	15
5	17	15	83	177	50	531	45	49	17	16
6	12	18	63	74	62	36	704	13	12	6
7	24	16	50	40	62	96	22	628	13	49
8	79	60	19	16	15	15	18	7	717	54
9	45	133	18	17	12	13	12	29	60	661

Figure 2-2. Confusion matrix in original SSL model. (HW6 Pr2)

Legend:
1.Green box: confusion level is somewhat reduced.
2.Yellow box: confusion level is somewhat raised.

(2) Training time

Module	Training Time
Module1(i.e., PixelHop2.fit())	280.12 + 246.83 = 526.95 sec = 8.78 mins
Module2(i.e., Feature Selection, LAG)	(309.56 + 270.80 + 167.80) + (99.16 + 242.14 + 167.80) = 1257.26 sec = 20 mins
Module3(i.e. Random Forest)	399.73 + 408.73 = 808.46 sec = 13 mins

Table3. Training time at each module

(3) Inference time

Module	Inference Time
Module1(i.e., PixelHop2.transform())	(171.06+158.91+147.86) + (81.48+82.50+83.78) = 725.59 sec = 12.09 mins for 20000 images
Module2 (i.e., Feature Selection, LAG.transform)	(0.13+ 0.29 + 0.0108) + (0.17 + 0.24 + 0.01) = 0.85 sec
Module3(i.e. Random Forest.prediction)	1 sec

(4) Test accuracies as a number of training samples drops

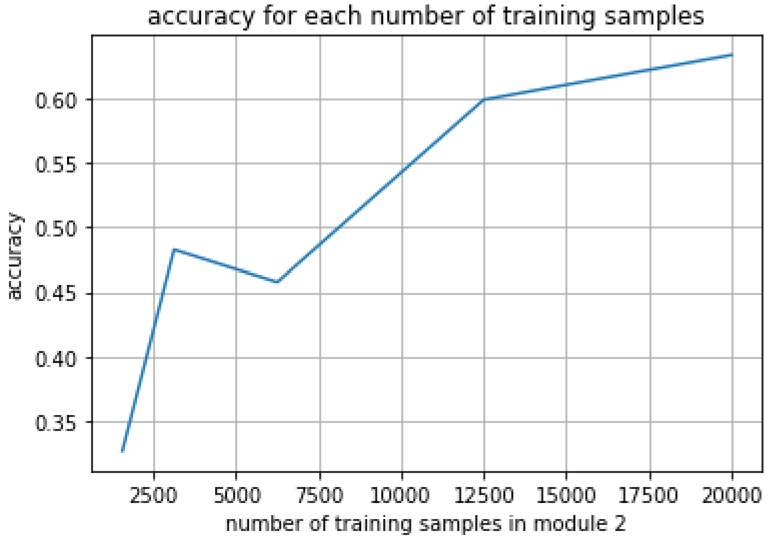


Figure 3. Test accuracies as a number of training samples drops in SSL model

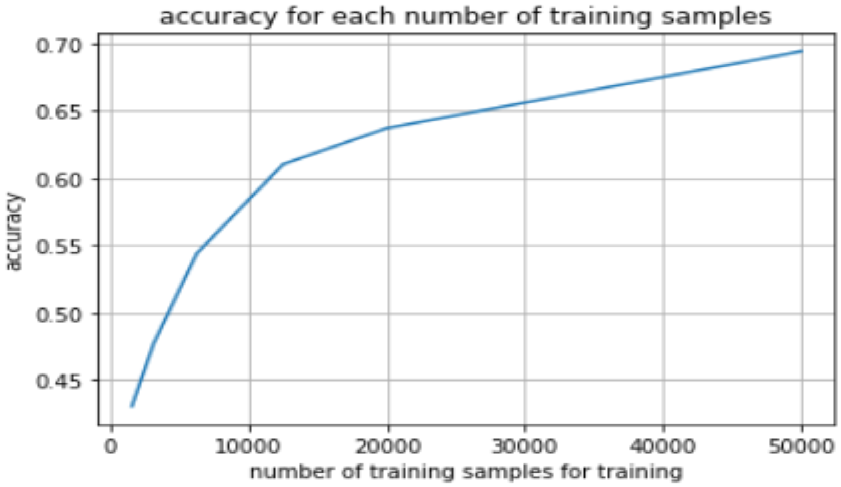


Figure 4. Test accuracy as a number of training samples drops in CNN model

(5) Model size

<Model size for SSL model with 5x5 size of window>

Module1

=>1st PixelHop++ unit has 5x5 size of window, 3 channels => 5x5x3 = 75 parameters

=>2nd PixelHop++ unit has 5x5 size of window, 42 channels => 5x5x42 = 1050 parameters

=>3rd PixelHop++ unit has 5x5 size of window, 278 channels => 5x5x278 = 6950 parameters

Module2

=>1st LAG unit has 70 dimension of output, n1 features after feature selection

= > 70 x (4116+1) = 288,190

=>2nd LAG unit has 50 dimension of output, n2 features after feature selection

= > 70 x (3412+1) = 238,910
=>3rd LAG unit has 50 dimension of output, n3 features after feature selection
= > 70 x (265+1) = 18,620

Module3
Random Forest Classifier has been used.
Number of estimators: 400
Bootstrap = true

<Model size for SSL model with 3x3 size of window>
Module1
=>1st PixelHop++ unit has 3x3 size of window, 3 channels => 3x3x3 = 27 parameters
=>2nd PixelHop++ unit has 3x3 size of window, 19 channels => 3x3x19 = 171 parameters
=>3rd PixelHop++ unit has 3x3 size of window, 96 channels => 3x3x96 = 864 parameters

Module2
=>1st LAG unit has 70 dimension of output, n1 features after feature selection
= > 70 x (2137+1) = 149,660
=>2nd LAG unit has 50 dimension of output, n2 features after feature selection
= > 70 x (2352+1) = 164,710
=>3rd LAG unit has 50 dimension of output, n3 features after feature selection
= > 70 x (3250+1) = 227,570

Module3
Random Forest Classifier has been used.
Number of estimators: 200
Bootstrap = true

=>Total number of parameters of the model used in HW6 Pr3 = 553,795 + 543,002 = 1,096,797

1.3 Discussion
-Observation on accuracy improvement and its sources
We could have improved performance of our model by adding more units with different size of spatial neighborhood construction and putting more seeds in LAG units. As we explain earlier,

the reason the accuracy has been approved is that more units and more seeds yield more distinguishable features when it comes to classification task.
Also, we can see this design approach could have somewhat lower confusion among certain classes in confusion heat map comparison at figure 2-1 and 2-2.

-Observation on training and inference time
It takes only 41 minutes for training which is much faster than training time for CNN-based model we designed in homework 5 problem 2 (e.g., CNN model takes 3 hours to finish its training). At module 1, the inference time seems quite long as it takes about 12 minutes. However, it means it takes only 0.03 second for one image which is very short even though we need to perform classification in real time. Therefore, SSL based model will have huge benefits when we are building a classifier on embedded systems because it can save time for training.

-Opinion on designing SSL structure
Since we know the exact roles of all hyper parameters and components in SSL model, we can spot where to modify with reasonable explanation. Increasing number of units certainly enhance performance of SSL model, but it surges model size rapidly. Additionally, it was still hard to try as many as hyper parameters because of my lack of code efficiency and time.

-Robustness
As we can see in figure 3, even though number of samples is reduced significantly, our model still maintains about 50% accuracy until the number of samples become 1532. The reason why it shows slower decreasing accuracy rate compared to CNN based model is that module 1 doesn’t need any supervision meaning it is independent on labels. Therefore, even though training set becomes smaller, our model can perform reasonably until some point. Additionally, we need to keep something in our mind during design when number of samples decreases that we should carefully pay attention to Ns and number of seeds in LAG units because they are indirectly and directly related to dimension of final feature space. Therefore, small enough value for Ns and number of seeds in LAG units should be chosen. In our model, we make use of 1000 for all features (when number of features is less than 1000 before feature selection) for Ns and 5 seeds for LAG units. Furthermore, if I could have used 10000 training images for unit 1 and had more time to figure out optimized value for Ns and number of seed in LAG units, then decreasing rate in figure 3 should be slower than decreasing rate in figure 4 as the number of training sample in module 2 becomes smaller.

-discussion about hyper parameter selection and future work.
In this work, I couldn’t explore as many hyper parameters as I wanted to due to time constraint. In the future work, we can investigate more on Ns, number of seeds in LAG units, TH1, and TH2 while following principle of each component. Also, I need to improve programming skill, so that I can write up code more memory-efficiently.

References

[1] 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.

[2] 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
[3] CIFAR-10 dataset (online) Available:
https://samyzaf.com/ML/cifar10/cifar10.html