EE569 Digital Image Processing

# EE569 Competition --- CIFAR10 Classification

Zheng Wen
zwen1423@usc.edu

## Motivation and Logic behind Design

In order to get a higher accuracy on the basic of PixelHop++, several models are tried. In the beginning, ensemble learning is considered. The diagram is shown in Fig. 1. Borrowed from the experience from deep learning, the min pooling is meaningless, so only max pooling and average pooling are used. 6 PixelHop++ unit are trained, 3 of them are using max pooling function to process the output while the other 3 are using average pooling. The results are put into feature selection module and LAG unit separately, then the output of these 6 LAG unit are concatenated together to form a long feature vector. In this way, more features are put into the classifier, and after feature selection, all of the chosen features are useful. But these refinement does not improve the accuracy anymore. After trying hard to change the hyperparameters of random forest and the feature selection part, this plan is discarded. Besides the unchanged classification accuracy, another reason to discard this plan is that the training time is much longer than the original system because there are much more features to be selected in Feature selection model, which is the most time consuming part in the whole diagram, in comparison with the original system. Another reason to discard this plan is that the number of parameters is doubled. The modest accuracy in this plan is 62.35% on test set.

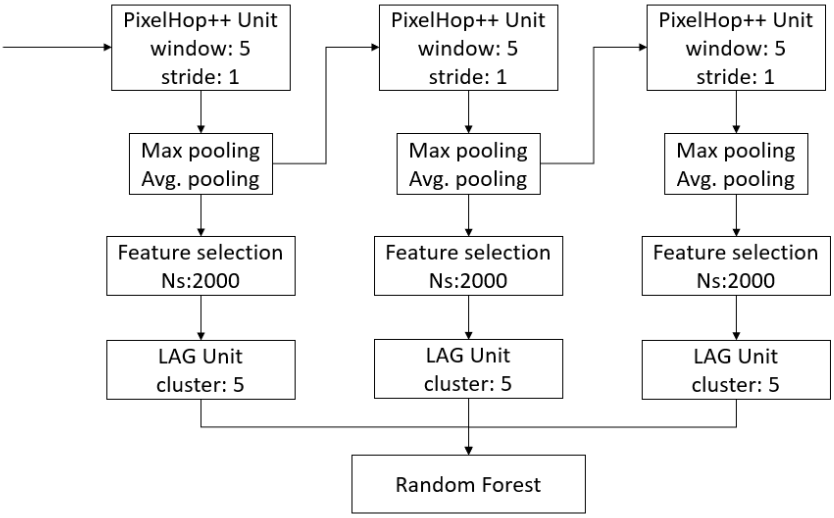



Fig. 1 First Trial

Another plan is to change the structure of PixelHop++ unit. In the original system, all of the three PixelHop++ unit uses 5*5 window with stride 1. In this way, the principle component could be extracted, and this setting could make the input of the last PixelHop++ unit is just the same size with window, the actual effect is the same as global max pooling, which is commonly used in the last layer of CNN. In order to get a finer look at the whole image, I tried to chose different window size in PixelHop++ unit. In my implementation, the window size of the first PixelHop++ unit is 7*7, because in this way, the anchor vector

could have a higher dimension, which will make the kernels more diverse. Besides, the first PixelHop++ unit actually take a look at the original image, if the window size is small, the PixelHop++ unit would more focus on local information, which is not suitable for PixelHop++ unit where the principle components of each patch are extracted. In my opinion, PixelHop++ unit need to extract more holistic information in the first layer because the local information may be similar among different classes, so I believe that 5*5 in the first layer is not enough. The window size in the second PixelHop++ unit is remained as 5*5, but the window size of the last PixelHop++ unit is set to 3*3 because the size of the input of the last PixelHop++ unit is 4*4, which could not accommodate a 5*5 window. But this plan also fails. After finetuning the hyperparameters of the random forest and the number of clusters in LAG unit, the best classification accuracy in this plan is 64.84%. The plot of this plan is shown in Fig. 2

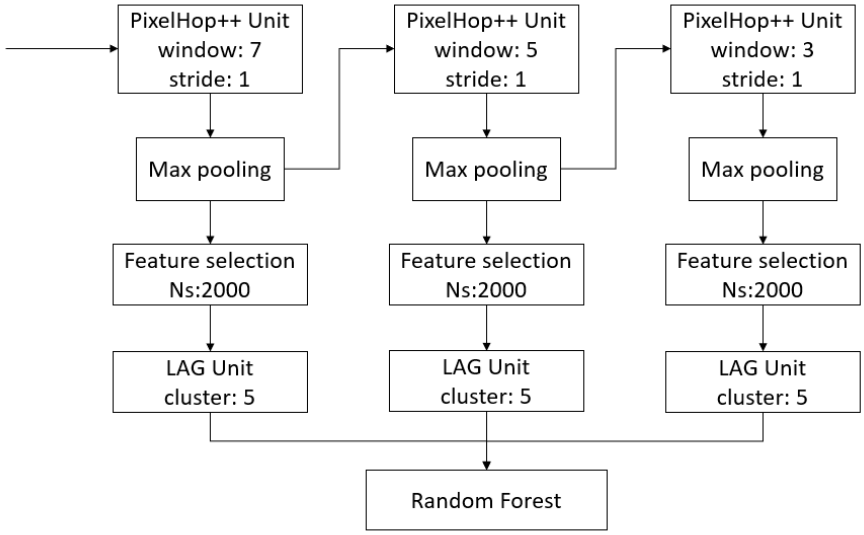



Fig. 2 Second Trial

So, the final plan in my implementation is shown in Fig. 3. In this implementation, the number of clusters in LAG unit is increased. Because after the failure above, I believe that the global-max-pooling like operation in the last PixelHop++ unit is critical for the whole process because the system could have a holistic view to the whole image in this way, so the structure of PixelHop++ unit is not changed in comparison with the original model. The only change is that the number of selected features in the Feature selection model is increased to 1500, and the number of clusters in the first two LAG unit is changed to 6. Because the dimension of input of the first two LAG unit is much larger than the third LAG unit, so if more clusters are offered in the first two LAG unit, the data points could be clustered finer, and thus the classification accuracy could be increased. Another changes in this model is that the classification method is support vector machine. Because SVM could have a better classification accuracy than using random forest. The classification accuracy depends largely on the initial random state and the number of estimators. After the number of estimators increases to some degree, the classification accuracy could not increase anymore as the number increases again. But the accuracy will still vary because its different random state. It is harder than SVM to adjust the hyperparameter even if the training time is shorter than SVM. The source of increased accuracy comes from the change of classification method, and the more selected features, as well as more clusters in LAG unit as illustrated above. One noticeable thing is that

more clusters in LAG unit is applied, the classification accuracy degraded when the number of training samples is extremely less, which is because the number of training samples is less, there is no need to use such much clusters in LAG unit, or the cluster could not represent the distribution better, there will be ambiguity among these clusters. Some other techniques like data augmentation are also tried, but they do not work.

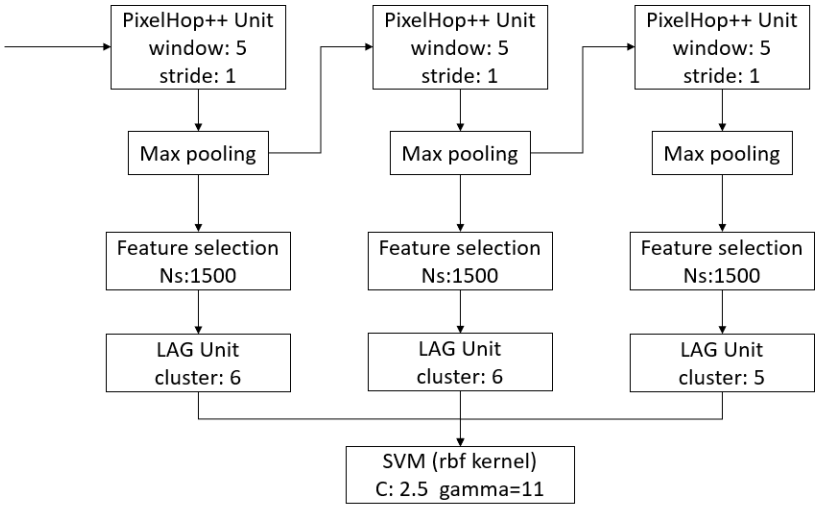



Fig. 3 Final Model

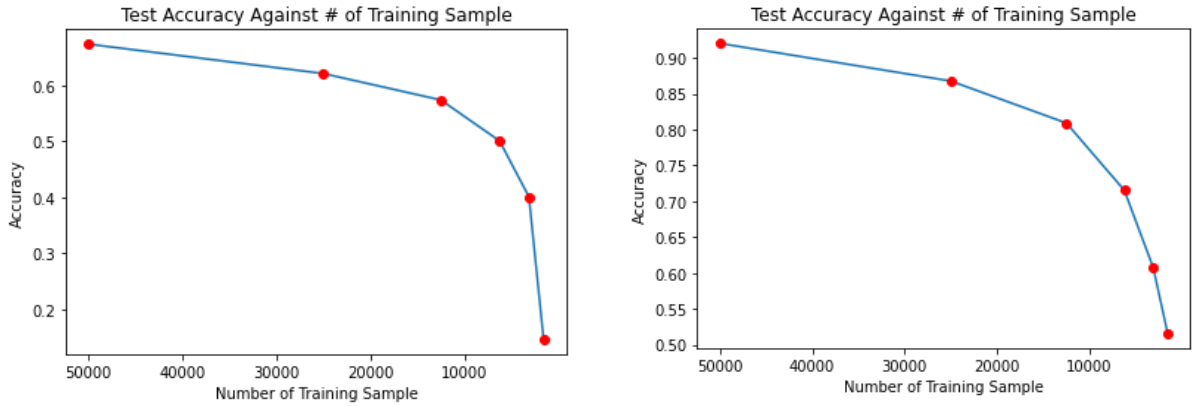
**Classification Accuracy, Model Size and Running Time**

| Number of training samples | 50000 | 25000 | 12500 | 6250 | 3120 | 1560 |
|---|---|---|---|---|---|---|
| C | 2.5 | 2 | 0.5 | 0.2 | 0.03 | 0.0005 |
| gamma | 11 | 11 | 5 | 1.3 | 0.55 | 0.0001 |
| Accuracy | 67.33% | 62.07% | 57.35% | 50.04% | 40.05 | 14.60% |
| Training time | 3865.6s | 1954.7s | 1194.8s | 913.9s | 835.1s | 794.1s |
| Inference time | 102.7s | 59.5s | 40.9s | 28.9s | 27.5s | 23.7s |

Table 1. Classification Accuracy and Ablation Study

In my implementation, the output of three PixelHop++ unit is a matrix with channel-last shape, these three unit have 41, 215 and 497 channels respectively, so in these three PixelHop++ units, there are 41 5*5*3 filters and 712 5*5 filters in total. The number of parameters here is 20875. The feature selection section stores the position of 1500 positions with smaller cross entropy, so in general, there is 1500 + 1500 + 1500 = 4500 parameters, but in my implementation, the last PixelHop++ unit only have 497 dimensions, so the number of parameters is 3497 in my setting. After feature selection by choosing the 1500 features with less cross entropy, the input dimension of each LAG unit is (50000, 1500), (50000, 1500) and (50000, 497), the output of each LAG unit are matrices with 60, 60, 50 dimensions in the second axis respectively, which

means the parameters in these three LAG unit is 90060, 90060 and 24900, the number of parameters in LAG unit is in total 205020. So, in total, there are **229392** parameters in the model. The classification accuracy on test set is **67.34**%, the according parameters of support vector machine is C=2.5, gamma=11, the kernel of SVM is Gaussian kernel. The classification accuracy and running time as the number of training samples degrades is shown in Table 1, the inference time denotes the total time evaluating the whole test set. The curve of accuracy is shown in Fig. 4(a).




(a) SSL (b) BP-CNN in HW5 P2

Fig. 4 Classification Accuracy

## Analysis and Comparison with BP-CNN

From the curve in Fig. 4, BP-CNN system has a better accuracy. This is because in the BP-CNN system in HW2 P2, there are a lot of advanced technique like learning rate schedule, cosine decay, data augmentation like cutout, and the advanced residual attention network which is a accumulative effect of research outcome in recent years, while in SSL, there is no so much techniques to choose from, it is a brand new machine learning based algorithm, if there are more techniques to choose from, I believe the performance will exceed BP-CNN. The performance of SSL degraded severely when 1/32 training samples are used, which is because of the increased number of clusters in LAG unit. Another reason is that the training samples are randomly selected among each class, the chosen data may not represent the feature of the class well. The number of parameters in BP-CNN is 396810, nearly twice in comparison with SSL system, the SSL system is much smaller than BP-CNN. Another advantage of SSL system is the faster training speed. When training with the whole training set, the SSL system takes no more than 2 hours on a single i7-8565U CPU, but training a BP-CNN in HW5 P2 takes nearly 5 hours on a Nvidia Tesla P100 GPU to get a better result. Besides, the BP-CNN is very sensitive to the choice of parameters, if the learning rate is not suitable, the whole system will degrade rapidly, which is not exist in SSL system. A shortcoming of SSL system is that in the training process, the algorithms takes up a lot of ROM, because my computer has a limit ROM, so a lot of ideas could not be carried out on my own computer. I also tried to use AWS p2.xlarge instance to assist my training process, but it is too costly, and some ideas are turned to be meaningless. Another shortcoming of SSL is the choice of classifier. Different classifiers will lead to different classification accuracy and different training time. I believe that if a unique classifier which is designed for SSL could be carried out, the performance will become more stable and pleasing.