HOMEWORK#6 Report

Issued: 4/14/2021 **Due: 11:59PM, 4/30/2021**
Name: Siyu Li **ID:2455870216**
E-mail: lisiyu@usc.edu

Problem 1: Origin of Green Learning (GL) (35%)

(a) Feedforward-designed Convolutional Neural Networks (FF-CNNs) (20%)

(1) Summarize the Saab transform with a flow diagram. Explain it in your own words.

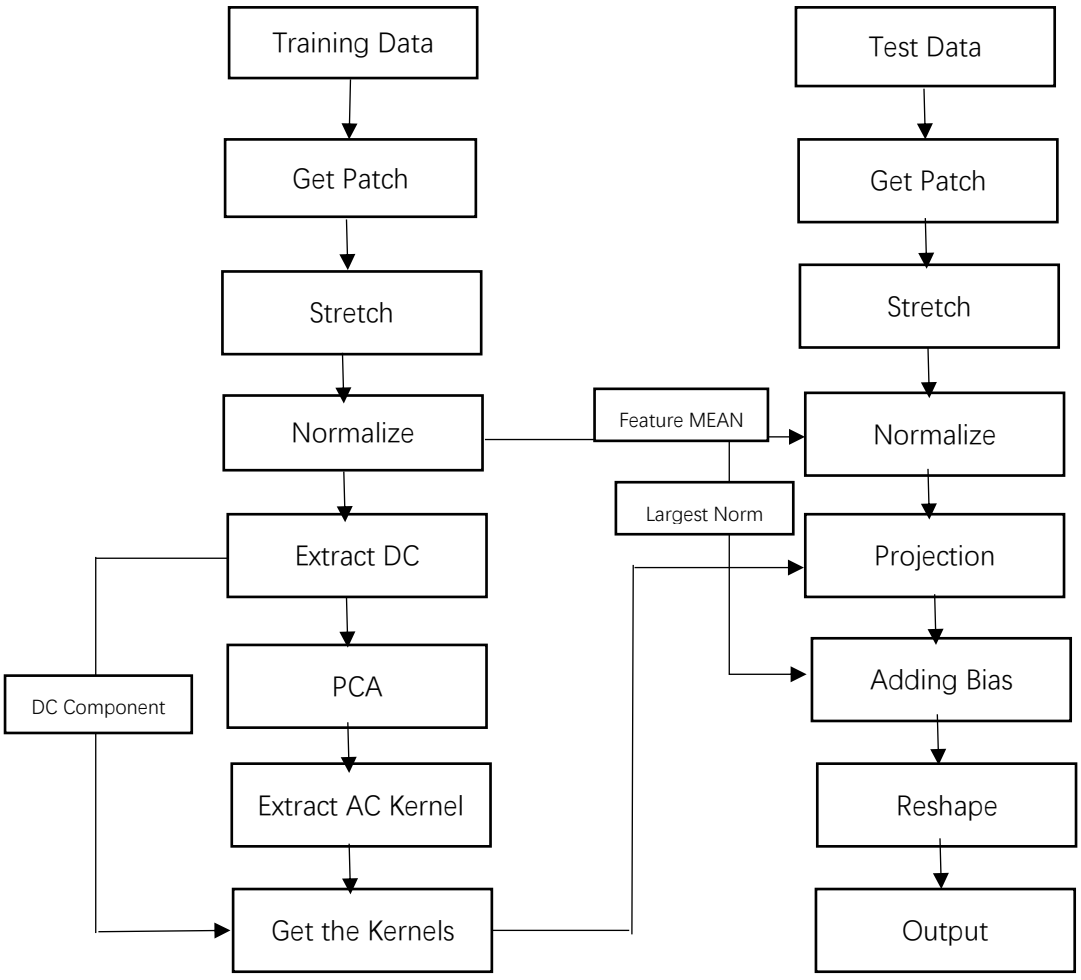


Fig 1.1 Training and Testing Flow Diagram of Saab Transform

Saab transform is an effective method of feature extraction and dimension reduction

and it originates from Saak **[1]** transform which performs feature extraction and eliminates the sign confusion issue in CNN caused by the asymmetric nature of activation functions. However, Saak transform doubles the feature map size so it is not computationally efficient. In Saab transform, a bias term is added to each layer of PCA result, which seems a more efficient way.

The flow diagram that describes the Saab transform is in **Fig 1.1**, the left side shows the cascade procedures of training process and the right side shows the procedures of testing process. The central part represents conveying the information acquired in the training process to the testing process.

In the process of Saab transform, the first step is extracting features and getting kernels from training data. The training image is scanned and traversed with a certain window and stride and then we get the patch. The patches are put together and then stretched into a one-dimensional vector. Data normalization is applied by removing the mean from features and then AC-DC decomposition is implemented. Then, PCA is applied on the AC part to extract features and these principal component feature vectors are called AC anchor vectors. Finally, the kernels are acquired by putting AC anchor vector with DC vector and then extracting the mean form corresponding patches.

Meanwhile, in the testing process, the test data is converted to patches and transformed into vectors by implementing the same procedure as training process. Then the stretched vectors are normalized by using the feature mean extracted from training data. And then they are projected into the kernels acquired in the training part and bias which is the largest norm from the training part is added to each projection element to address the issue of sign confusion. Finally, the feature vectors are reshaped to the transformed spatial dimension.

(2) Explain similarities and differences between FF-CNN and backpropagation-designed CNN (BP-CNNs)

Similarities: Both FF-CNN **[2]** and BP-CNN have very similar structure and they are used to do classification. They both have the feature extraction part and classification part. The feature extraction part in BP-CNN is implemented by cascading several layers of convolutional layers and pooling layers to acquire features from multiple resolutions and achieve certain invariance while FF-CNN also uses some unsupervised learning methods except for filtering and pooling. And both of them have classification part where BP-CNN uses fully-connected layers and sigmoid functions while FF-CNN can use some traditional statistical classification method such as SVM or Xgboost.

Differences: FF-CNN is a one-pass method which is based on the statistics from the previous layer while BP-CNN needs lots of iterations to get the parameters feedback from the minimization process of loss function.

From the perspective of feature extraction part, this part in FF-CNN is unsupervised, which means its kernel are built by projection where principal components of AC features are extracted from the input data. And DC kernels are usually discarded because it has little information. The unsupervised and statistical natural of feature

extraction part in FF-CNN make it possible to get the kernels in only one pass because you do not need to change the same kernel over and over again. Meanwhile, the kernel acquiring process in BP-CNN is a progressive one which relies on the gradual minimization process of loss function during a huge number of epochs.

Another major difference is that FF-CNN has the capability to solve the sign confusion problem while BP-CNN does not. Because a bias term is added to the output feature map and it can make the cascade of PCA layers useful.

In the supervised classification part in FF-CNN, to fully exploit the extracted features to improve the performance, they are clustered by k-means into different clusters whose number is much larger than the real number of labels and the labels assigned based on different clusters are called pseudo label. This can be justified that in a certain class of object there can be some intra-class different, which means a class can be classified into some different sub-class again. For example, the parent class ship can include sub-class such as cargo ship, towing ship and aircraft carrier. This method will be much better than the case that the huge number of features are connected directly to the layer generating final classification result. And in its classification part, most of the time is consumed on calculating the inverse matrix in LAG for the latter regression.

For BP-CNN overall, it a highly supervised method so its requirement on data is much higher than that of FF-CNN and it also has a lower generalization ability if the disparity between training set and test set is slightly larger. And for the computational cost, FF-CNN can be training much faster owning to its one-pass property and less parameters needed to be trained while in BP-CNN, due to its backpropagation nature which requires the feedback from minimization of loss function to a huge number of parameter positions over and over again in each iteration to looking for the optimal parameters.

(b) PixelHop and PixelHop++ (15%)

(1) Explain the SSL methodology in your own words. Compare Deep Learning (DL) and SSL.

SSL **[3]** is a new and computationally efficient machine learning method. The most innovation part of the feature extraction part is that unsupervised learning such as K means rather than supervised parametric backpropagation model in DL is applied to extract features. This will yield better computational efficiency as it is a one-pass unsupervised learning process and it also relies less on then quantity of training data. Its feature selection part also extracts feature on multiple scales by pooling to make full use of the input data by increasing the receptive field and it can also bring some invariance property. It can also make better use of the training set by neighbor construction. Moreover, the dimension of extracted features can be reduced by methods such as Saab or Saak transforms in order to extract more useful features and get better generalization capability meanwhile the kernel number in DL cannot be reduced to perform feature reduction.

And in the classification part, label assist regression is applied to do dimensional

reduction by putting pre-classified labels in the intermediate process to avoid connecting too many features directly to the final classifier. The number of this kind of pseudo labels will be much larger than that of the real labels at the end. This can be justified that in real world scenarios for a certain class of object there can be some intra-class different, which means a class can be classified into some different sub-class again. With the dimensional reduction by LAG, the small input of final classifier can lead to less parameter number in the final classifier. The setting of final classifier can also be flexible as we can use SVM, Linear Regression, Xgboost and other traditional statistical machine learning method rather than neuro network only in DL.

For the comparison between DL and SSL, the most important thing is that SSL has an unsupervised learning-based feature extraction mechanism and it can also perform feature reduction on the features to maintain useful ones and discard less useful ones. This can make SSL a one-pass and thus more computationally efficient method than DL. In DL, the number and size of kernels cannot be adjusted and due to the highly supervised nature, it needs a huge number of parameters and iterations to be trained. Moreover, in the backpropagation mechanism, the process of parameter updating is very vulnerable and unstable as gradient explosion, gradient decay and sign confusion may occur. While in SSL, the feature extraction part is fully non-parametric one which evades these problems and provide stability in training and better generalization performance.

When it comes to the common part, both DL and SSL can extract features from multiple scale if designed properly and use convolution operation to do extraction.

And for the difference in classification part, the application of LAG in SSL can reduce the dimension of output effectively and make the model size smaller. Its classifier setting is also more flexible as traditional method such as SVM and Xgboost can be used rather than neuro network only.

In general, SSL has better computational efficiency and better generalization ability. It requires less training data. However, under some special case that the data is very rich and the disparity between training data and test data is minimal, DL can achieve better result on both training set and test set owning to its highly supervised nature.

(2) What are the functions of Modules 1, 2 and 3, respectively, in the SSL framework?

The first module cascades Pixelhop **[3]** or Pixelhop++ **[4]** unit with pooling layer and unsupervised method such as Saab transform is used to project features from previous patches to lower dimension. The purpose of this cascading method is to extract information from multiple scales to enlarge the receptive field in order to explore the information in the input data thoroughly. The pooling layer between pixelhop units plays a role of both eliminating redundancy and achieving multiple scale property. With a fixed window size, the rescaling of original image exactly equals to change the window size that can decide the receptive field. And there is also neighbor construction process in pixelhop part, this will help exploit the information of neighbor connection

better.

The second module is aggression part or feature selection part and LAG part which performs feature reduction by assigning pseudo labels before final classification. The aggression part picks some statistical value to give further representation of feature map, making it easier to do classification. Feature selection part selects the most discriminant part in the extracted features by cross entropy criteria. The LAG has already been discussed in detail previously.

The third module is primarily used for the final classification. The features process by the previous module is concatenated together to be sent into the final classifier such as Random Forest, Xgboost or SVM to get the final classified labels.

(3) Explain the neighborhood construction and subspace approximation steps in the PixelHop unit and the PixelHop++ unit and make a comparison. Specifically, explain the differences between the basic Saab transform and the channel-wise (c/w) Saab transform.

The neighborhood construction is a method that collects neighbor information of a pixel by a window with certain size and some reshape process to extract more information. It is the same in both pixelhop and pixelhop++ method.

For the subspace approximation part, Saab is used in pixelhop while channel-wise Saab (cwSaab) is used for pixelhop++ for projecting high dimensional features to low dimensional feature space to do feature reduction. This is the only difference.

In Saab, the feature map is processed as a whole part. The patches are collected within the entire feature map. And PCA is implemented altogether to calculate the anchor vectors. The kernels are selected by given threshold and the input feature map will convolve with these kernels to acquire the output feature map.

However, in cwSaab, the transform is performed channel-wise. For each input feature map, the kernels are computed by channel instead of regarding all channels as a whole. There are two thresholds in PixelHop++, TH1 is used to determine which channel can be split further. Channels whose value is beyond TH1 are sent to next Saab transform, while those below TH1 are remained as leaf node or discarded. The TH2 is used to determine whether they will be remained as leaf node or just be discarded. The advantages of cwSaab are lower computational complexity because the number of parameters in cwSaab transform is smaller than that in Saab. Moreover, the subspace constructed by these principal components is orthogonal, which also justifies the cwSaab.

Problem 2: MNIST & Fashion-MNIST Classification (65%)

(a) Building the PixelHop++ Model (35%)

I. Experimental Results

In the training process of module 1, 10000 datapoints are used after the process of getting balanced dataset due to the limitation of computer memory. And only Hop3 features extracted from module 2 are used for training the final Xgboost classifier in module 3. In the training of module 2, 60000 datapoints are used to provide enough information to feed the module 3. The training of both module 1 and module 2 is unsupervised so it does not need labels in the training set. All the parameter settings are taken from table 1 in the homework6.pdf. TH1 is set to 0.005 and TH2 is set to 0.001. In the Xgboost classifier, the parameters are respectively n_jobs=-1, objective='multi:softprob',max_depth=6,n_estimators=100,min_child_weight=5,gamma=5,subsample=0.8,learning_rate=0.1,nthread=8,colsample_bytree=1.0.

(1) (2):

Dataset	Training Time (s)	Training Set Accuracy (60000 data points)	Test Set Accuracy (10000 data points)	K1	K2	K3	Model Complexity Based on Number of Parameters
Mnist	283.4	0.98595	0.9649	24	109	128	6625
Fashion-mnist	133.6	0.9084	0.8586	23	58	70	3775

Table 2.1

Function to calculate model complexity of Pixelhop++ based on cwSaab: (K1+K2+K3)*25

(3):

TH1 Value Mnist	K1	K2	K3	Model Complexity	Test Set Accuracy
0.1	24	47	22	2325	0.9115

0.01	24	106	110	6000	0.9626
0.005	24	109	128	6525	0.9649
0.001	24	109	131	6600	0.9679
0.0005	24	109	131	6600	0.9679

Table 2.2

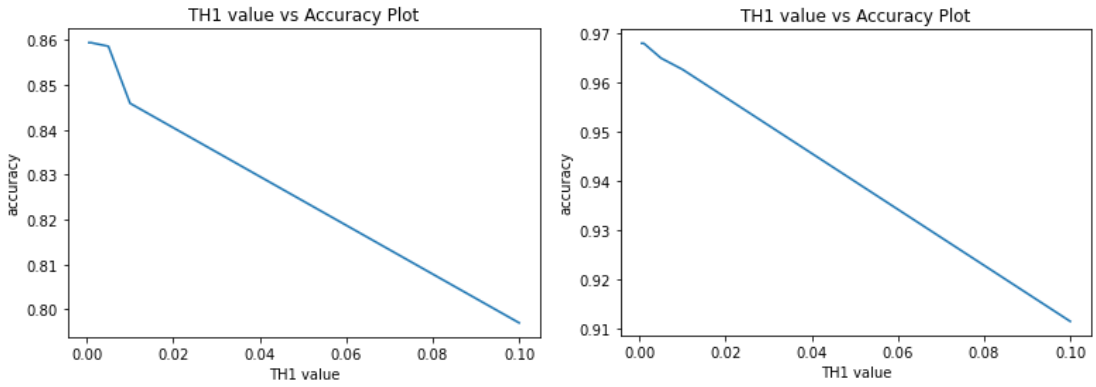
Function to calculate model complexity of Pixelhop++ based on cwSaab: (K1+K2+K3)*25

As is observed from Table 2.1 and Table 2.2, we can conclude that higher TH1 value will keep less parameters in the model because channels whose value is beyond TH1 are sent to next Saab transform, while those below TH1 are remained as leaf node or discarded. And the model complexity of Mnist model is obviously higher than that of Fashion_mnist model.

Also, more model complexity can contribute to higher test set accuracy in some way. But when the model has already been large enough, the increased model complexity will lead to very little increase on the test set accuracy.

TH1 Value Fashion_mnist	K1	K2	K3	Model Complexity	Test Set Accuracy
0.1	23	20	16	1475	0.797
0.01	23	51	53	3150	0.8459
0.005	23	58	70	3775	0.8586
0.001	23	58	78	3975	0.8594
0.0005	23	58	78	3975	0.8594

Table 2.3



(a): Fashion_mnist **(b): Mnist**

Fig 2.1 TH1 Value vs Test Set Accuracy Plot

From Fig 2.1, we can observe that in results of both datasets, with the TH1 increasing the test set accuracy is lower and lower. From previous discussion, we know that higher TH1 will lead to less model parameters or less complex model. Less complex model will save some time in computation but sometimes the model needs some complexity to be trained effectively. Here, with the model complexity going down, the test set accuracy also decreases.

(b) Comparison between PixelHop and PixelHop++ (15%)

I. Experimental Results

(1) (2):

Dataset	Training Set (s)	Training Set Accuracy (60000 data points)	Test Set Accuracy (10000 data points)	K1	K2	K3	Model Complexity Based on Number of Parameters
Pixelhop++							
Mnist	283.4	0.98595	0.9649	24	109	128	6625
Fashion_Mnist	133.6	0.9084	0.8586	23	58	70	3375
Pixelhop							
Mnist	204.5	0.987	0.9635	24	57	70	134550
Fashion_Mnist	183	0.9117	0.862	23	44	42	72075

Table 2.4

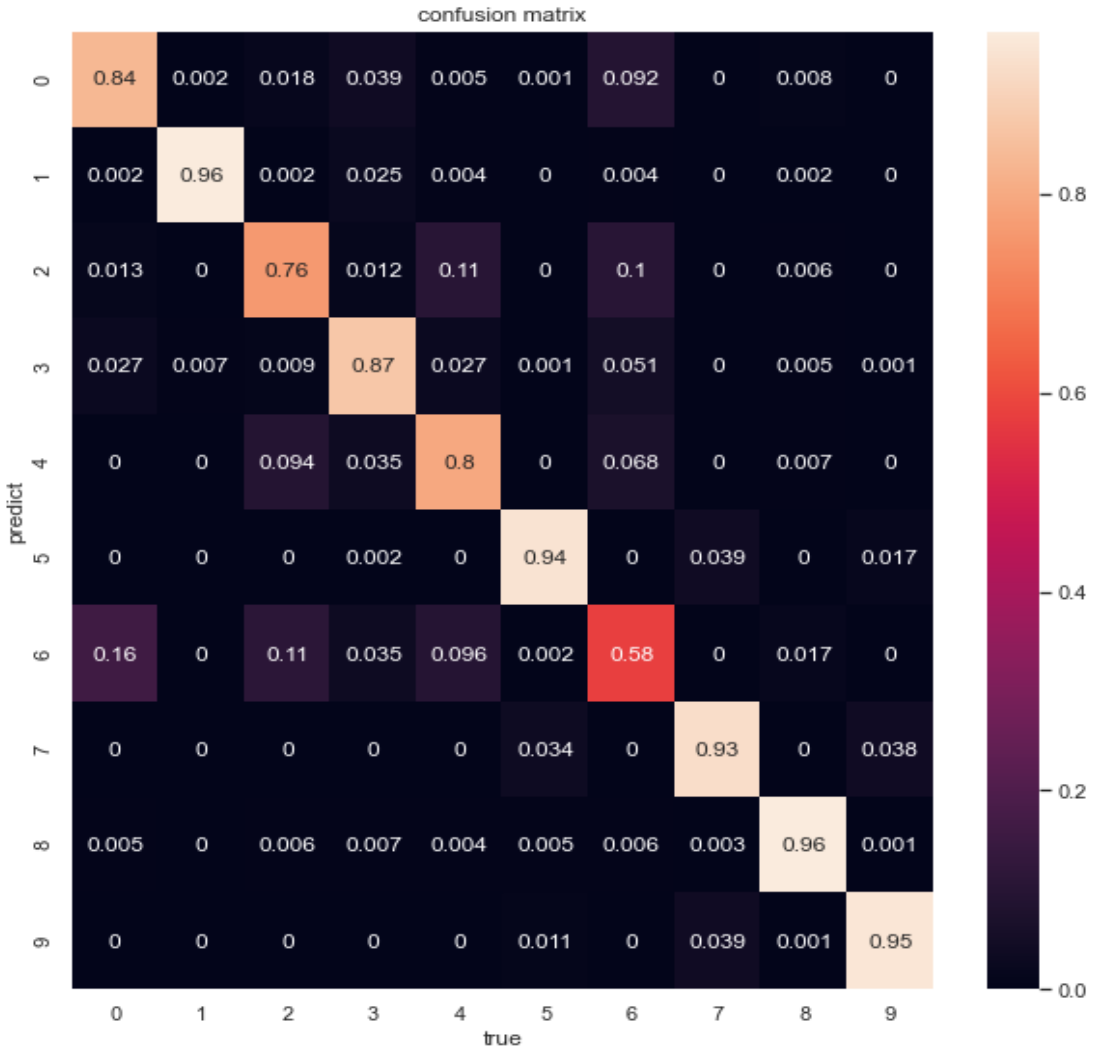
Function to calculate model complexity of Pixelhop based on Saab: (25*K1+25*K1*K2+25*K2*K3)

For the training time, training set accuracy and test set accuracy, the results of pixelhop and pixelhop++ only have very little difference. But when it comes to model complexity, with nearly the same test set accuracy, pixelhop++ can use less model complexity to get the result than that of the pixelhop. Because pixelhop++ based on cwSaab can eliminate more useless features and maintain the useful ones, which leads to less redundancy of the model.

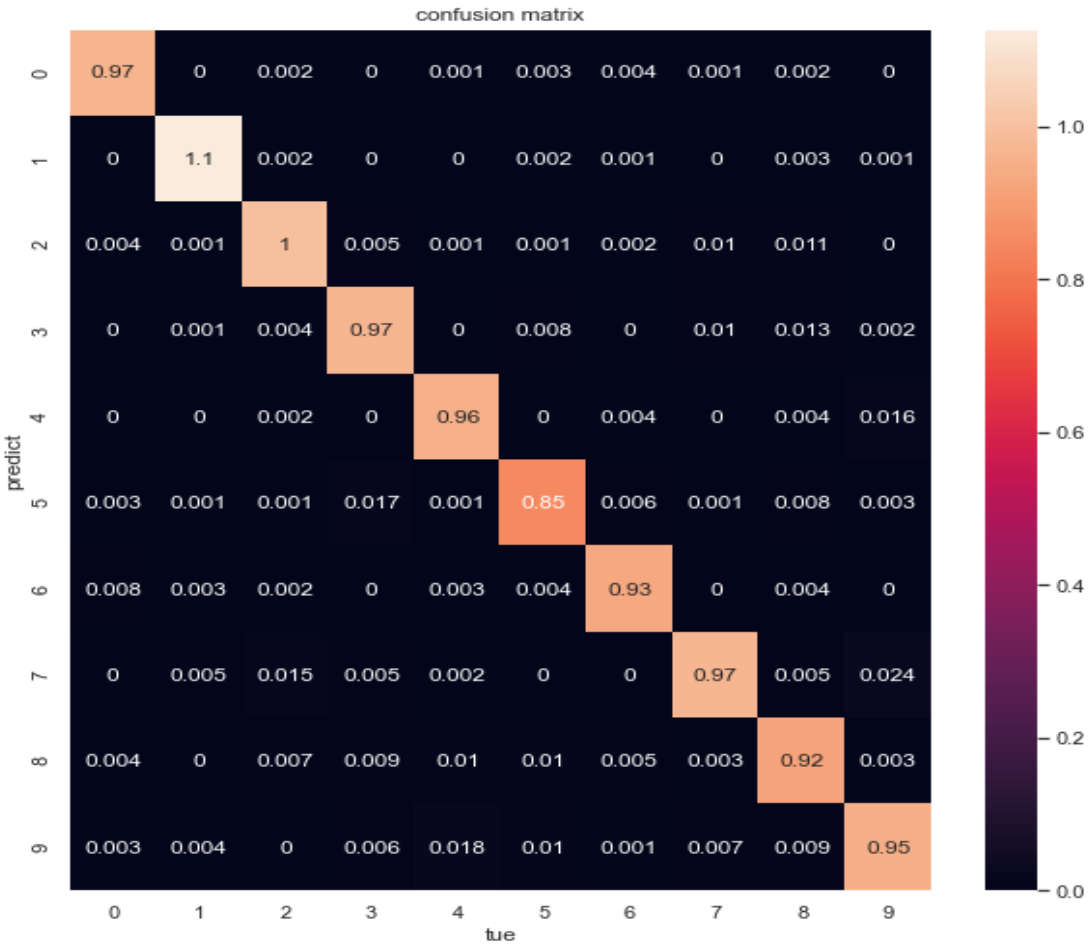
(c) Error analysis (15%)

I. Experimental Results

(1):



(a): Heatmap of Fashion_mnist



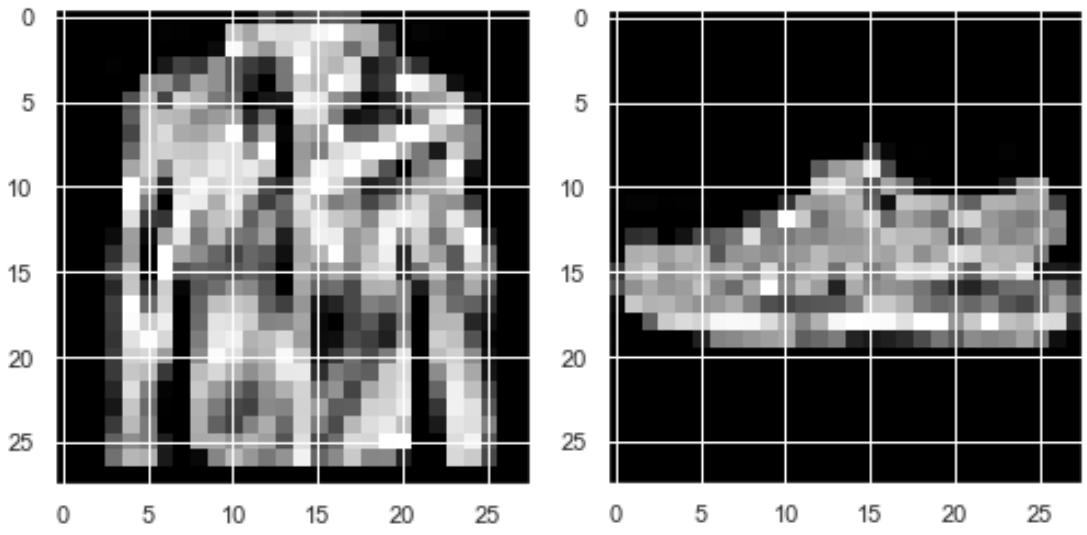
(b): Heatmap of Mnist

Fig 2.2 Heatmap on Testset of Fashion_mnist and Mnist

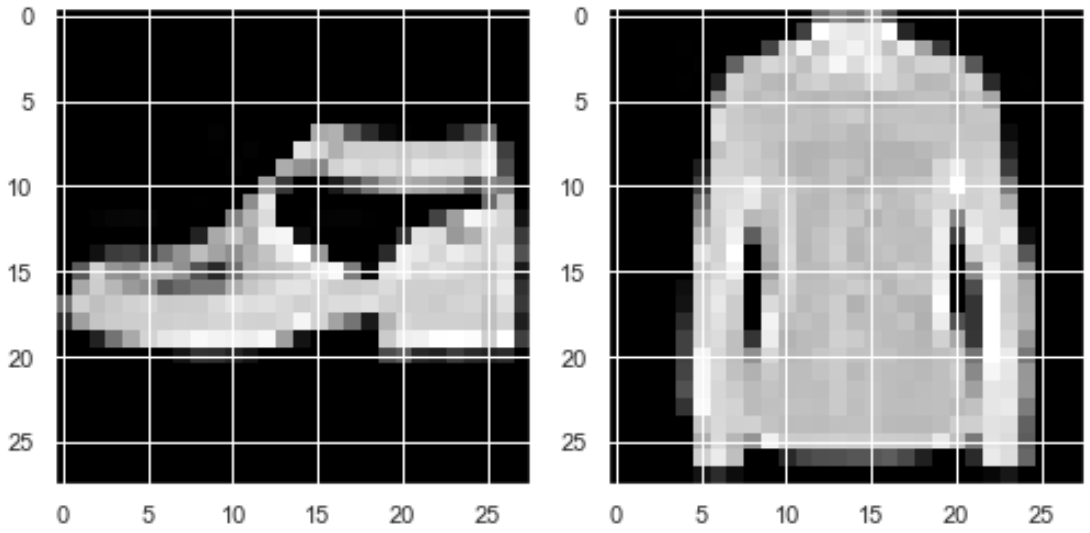
For the Fashion_mnist data set, class 1 and 8 yields the lowest error rate while class 6 yields the highest error rate.

For the Mnist data set, class 1 yields the lowest error rate while class 5 yields the highest error rate.

(2):



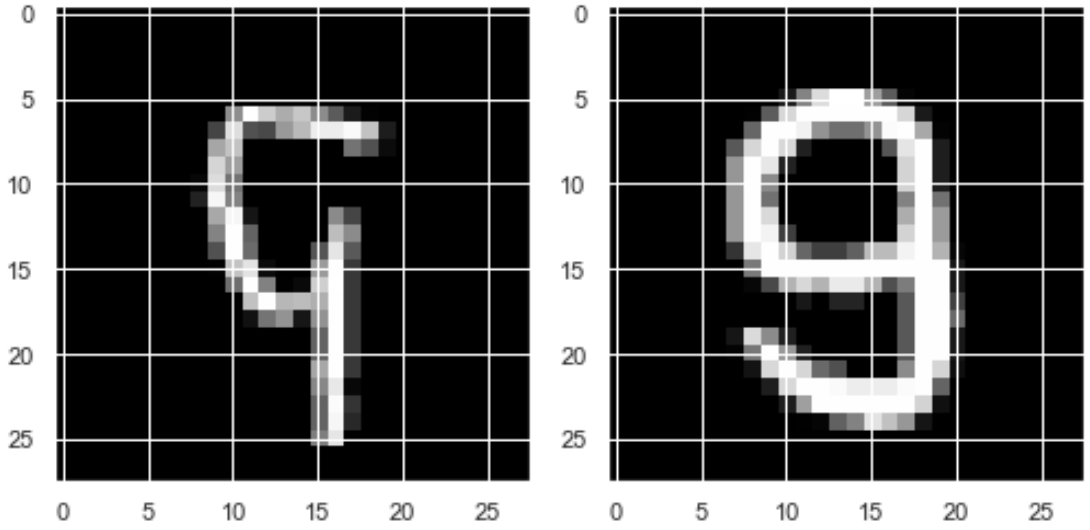
(a) Real Class= 4 Predicted Class= 2 (b): Real Class= 5 Predicted Class= 7



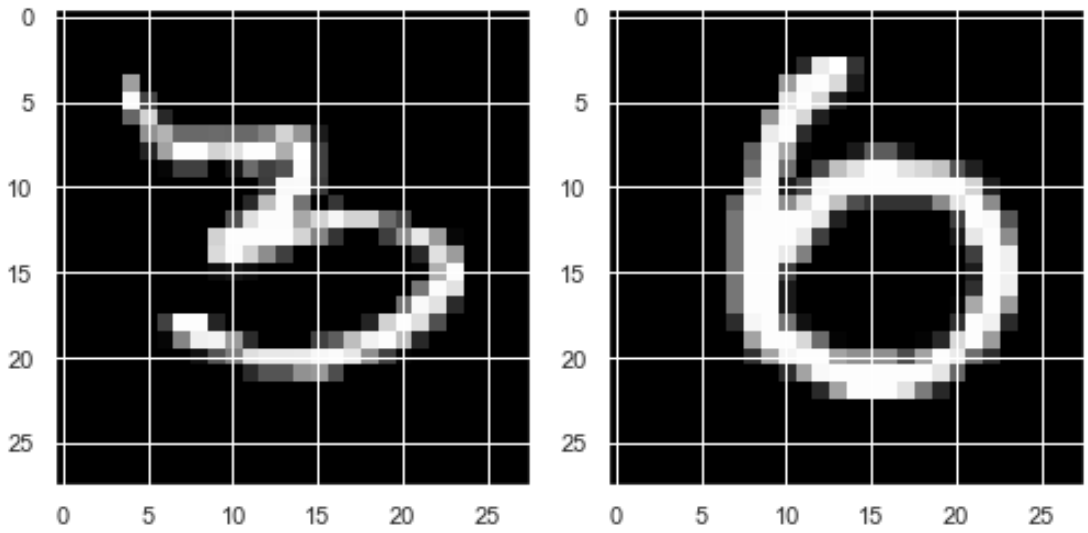
(c): Real Class= 9 Predicted Class= 5 (d): Real Class= 6 Predicted Class= 4

Fig 2.3 Misclassified Images in Fashion_mnist Test Set

As we can observe from Fig 2.3 and Fig 2.4, we can see in the confusion class images that more features or patterns are shared by some of the two class images. This is the reason why some images are misclassified. Take Fig 2.4(b) for example, the bottom left part of 9 in the image is nearly a closed loop but the image really means 9 and it is classified into class 8.



(a): Real Class= 9 Predicted Class= 8 (b): Real Class= 9 Predicted Class= 8



(c): Real Class= 3 Predicted Class= 5 (d): Real Class= 6 Predicted Class= 0

Fig 2.4 Misclassified Images in Mnist Test Set

(3): We can change the feature extraction part to other method such as Fisher Discriminant Analysis (FDA) to make the features extracted via PixelHop++ more appropriate. In the training process, the information extracted by small local patches in the shallow layers can be useless sometimes. However, in relatively deeper layer, each patch can have a very large receptive region, the feature map here tends to represent the features of a certain class much better. Therefore, the it is more proper that the deeper layer process could be given some supervision, which makes the features more discriminative for each class. To get the improvement, some semi-supervised algorithms in paper **[6]** can be implemented to do the improvement, this algorithm merges PCA and Local FDA together.

We can also use data augmentation to increase the accuracy between difficult classes by increasing the number of training datapoints in LAG and classifier.

References:

[1] C.-C. Jay Kuo and Yueru Chen, "On data-driven Saak transform," Journal of VisualCommunication 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

[6] Sugiyama M, Idé T, Nakajima S, Sese J. Semi-supervised local Fisher discriminant analysis for dimensionality reduction. Machine learning. 2010 Jan 1;78(1-2):35