EE569 Digital Image Processing

HOMEWORK #6

Zheng Wen
zwen1423@usc.edu

Problem 1: Understanding Successive Subspace Learning (SSL)

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

(1) Saab Transform:

Saab transform is a variation of principle component analysis (PCA), by adding a bias term for each single layer PCA outcome, eliminating the sign confusion which occurs when several layers of feature extractors are cascaded together and thus the non-linear activation function in CNN like relu activation could be eliminated, making the multi-layer linear combination of feature extraction possible. Saab transform comes from Saak transform, which solves the sign confusion problem by setting up two channels for a single outcome of each filter, but Saak transform doubled the size of feature map, consuming more resources. The bias term in Saab transform is a more reasonable way in comparison with Saak transform. By selecting the number of principle components to be preserved, Saab transform could also serve as dimension reduction method.

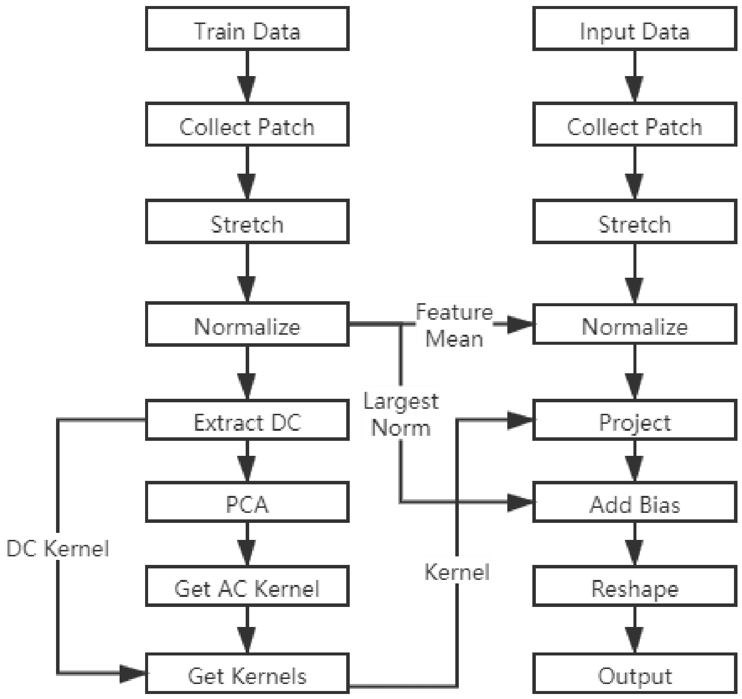


Fig. 1. Training Process (Left) and Transform Process (Right) of Saab Transform

To implement Saab transform, the first thing to do is to get kernels from training data. Each training image or training feature map is scanned with a window with a certain size and certain stride along the spatial domain, and the block in the window is collected as a patch. After collecting all of the patches from training data, they are put together and each patch is stretch into a 1D vector. The mean is removed feature-wise to normalize the data. After doing this kind of normalization, then the sample-wise mean is extracted as AC-DC decomposition. The remaining part is AC part, the principle component of AC part is extracted with PCA, these principle components are called AC anchor vectors. Combining AC anchor vectors with a DC vector extracting the mean of each patch, a group of anchor vector, i.e. kernels, is obtained. When doing Saab transform to any input data, the input data is transformed into patches and extracted to vector in the same way as the training process, then normalized using the feature-wise mean extracted from training data. Each feature vector is projected onto these anchor vectors, the bias term which is the largest norm of the training feature vectors is added to each element of the projections to eliminate sign confusion. Then each feature vector is reshaped to the transformed spatial dimension, and the output of Saab transform is obtained. The diagram of Saab transform is shown in Fig. 1. When training, the input data in the right-hand side is training data, and cascading the system by using the output of training data in this layer as the training data to train the next layer, the system could have a multiple-layer structure.

(2) Similarities and differences between FF-CNN and BP-CNN:

Because there a lot of variation of BP-CNN, LeNet-5 is used here to illustrate.

Similarities: both FF-CNN and BP-CNN could be used to classify. FF-CNN and BP-CNN have similar structure, like feature extraction part and classification part. They all use filters to do feature extraction, there are pooling operation between two convolution part to get a stronger invariance and increase the receptive field, and fully-connected like design at the end of the whole pipeline.

Differences: FF-CNN aims at deciding kernels by a one-pass method based on the statistic of the output of previous layer, instead of back propagation used in BP-CNN. In FF-CNN, the kernels are mainly decided by the principle component of patches collecting from AC part by a certain size and stride of window. DC component here has little information to leverage so it is dropped at most of the time. The convolution operation here is replaced by projection because there is no need to change the kernels time and time again during the training process once the training is done in a certain layer, so there is no need to keep the original shape of the kernels like BP-CNN. After the projection or convolution in a certain layer, there is a bias adding to the outcome feature map to eliminate the sign confusion in FF-CNN, so the activation like relu in BP-CNN could be eliminated. And the bias term makes the cascade of several PCA meaningful, or the cascade of several linear operations is the same as one linear operation. This part of feature extraction is unsupervised. To leverage the information offered by label, these features extracted by the convolution part is firstly clustered by K-means into different clusters, and then they are sent into label-assisted regression (LAG) unit, the features in the same clusters will be given a same label. The number of pseudo label is much more than the real class number. The regression result is then sent into another similar layer, and finally these features are cluster into classes with the same number of training classes. The later process of FF-CNN is supervised, and most of the time in training is spend to calculate the inverse of matrix in LAG unit to do linear regression. While BP-CNN is totally supervised. The information in labels is back propagated to each layer by minimizing the cross entropy in the last layer. The convolution layer takes a long time to train, and the parameters of fully connected layer are decided by back propagation. So, the parameters of convolution layers and fully connected layers changes in each epoch, the system need to look into training samples for hundreds of times to get a better performance. The number of parameters in the BP-CNN is much larger. Besides, BP-CNN need more training samples to get a higher accuracy, while the kernels of FF-CNN converge much faster as the number of training samples increase, so the system is more stable.

(b) Successive Subspace Learning (SSL)

(1) SSL is a brand-new machine learning method which uses several successive subspaces to extract statistical information from training samples and then using label-assisted intermediate sub-classes to reduce the feature dimension, then these features are fed into a classifier to classify. Instead of doing the traditional one-step transformation in machine learning, SSL could make use of several intermediate steps to approach the final result. This method has several advantages in comparison with the traditional ones. For example, in unsupervised part of PixelHop, there are several cascading layers with different input size in spatial domain, in this way, the receptive field could be different, the former part could capture the detail while the later part could capture an overall look, which will provide more information when classifying. Besides, because the patch-wise design, the neighborhood information could be fully exploited. The supervised part of PixelHop is also a kind of successive subspace learning where the feature vector is not directly matched with the final label, but by an intermediate pseudo label or probability to do feature dimensional reduction, which could make the regression process smoother, as well as combining label information into the feature vectors, and thus the dimensional reduction performance is better than directly projecting the high dimensional unsupervised feature vector to the labels. In the end, all features extracted with different neighborhood is combined together, forming a long feature vector used to classify. There are several similarities between deep learning and SSL. Both DL and SSL use convolution operations to do feature extraction, and pooling operation is used to reduce the redundancy as well as makes the system more stable. Several layers of convolution operation and pooling operation are cascades together, making the multi-scale feature extraction possible. Both DL and SSL have fully-connected design used to create some intermediate steps before making final decisions. Even though there are some similarities between these two methods, they are totally two different method in essence. In SSL, the kernels are decided by statistical information of the former layers, extracted by unsupervised Saab transform, which makes SSL could easily add more layers to a trained system without re-train the whole system, the convolution operation in SSL means projection each patch onto anchor vectors, while DL is a parameterized model, the weights in kernels is decided by back propagation (BP) to minimize the cross entropy, the process deciding kernels is supervised, if the structure is changed, even with the same training dataset, the whole network should be retained to get an optimal performance, the kernels are actually match filter extracting a certain feature of the input. The training time of SSL is shorter than DL, because SSL is a one-pass algorithm, every layer only need to see the training data once, while DL update the weights by BP, it have to look into the training data for hundreds of times to get a better performance. Besides, because of BP, DL demand a stronger computational resource such as GPU, a lot of state-of-art CNN designs are impossible to be trained with a single CPU, while the computational complexity is lower than DL for its one-pass essence. In comparison with DL which is a non-convex optimization problem, SSL have a better interpretability because all parts in SSL is convex optimization problem which is supported by mathematics nowadays. Because the feature extraction part of SSL is unsupervised, the model is less depend on the number of training data, and the kernels in the models converges rapidly as the number of training samples increase, the performance degrades slowly when the training samples are reduced, while DL need a lot of training samples to get a higher performance, the number of training samples is times of test data, which is unusual in traditional machine learning.

(2) There are three modules in SSL. The first module is cascading PixelHop unit or PixelHop++ unit. This module uses unsupervised method, i.e. Saab transform, to extract principle components from a patch size collected from the input image or feature map from the output of the former layers as kernels to project the patches onto these kernels to do feature extraction. The purpose of cascading is that in the first few layers of PixelHop unit, each patch reflects to a relatively small part of the original images, as the PixelHop unit goes deeper, each patch could correspond to a larger part in the original image. In this way, the spatial structure could be exploited thoroughly. Between two PixelHop unit, there is a pooling layer to reduce redundancy, and thus the dimension is reduced. The second module contains aggression part in [1], or feature selection part in [2], and label-assisted feature reduction operation. Aggression part leverages picking the maximum, minimum and average value among non-overlapping part of the output feature map of PixelHop unit to get more representation of the feature map, which could make it easier to classify. Feature selection part make use of cross entropy, selecting the most discriminant part in the extracted features. The aggression part or the feature selection part is actually dimensional reduction part, the redundancy part or the less useful part of the original output of feature extraction module is discarded. The label-assisted regression (LAG) unit is used to project the output feature map onto some intermediate clusters before doing classification. In this way, the dimension of the output features could be reduced rapidly, and the information in labels could also be exploited. Besides, in this way, the feature vector is not projected onto labels directly, but with some intermediate steps like projecting onto some target probability vector, making the performance much better with some seeds in each cluster. The third module is mainly designed for classification. To fully leverage the information extracted by each PixelHop unit, the feature processed by each LAG unit is concatenated together to form a long feature vector. Then these feature vectors are fed into a classifier like random forest or support vector machine to do train the classifier or classify.

(3) The neighborhood construction step is to make use of the information of local patches collected by some certain size of windows with a certain stride. Because in classification problem, the information of each single point is not enough to figure the content of an image. The information in a certain neighborhood of each single pixel point should be made use of. This step is the same in PixelHop and PixelHop++ algorithm. As for the subspace approximation part, in PixelHop, Saab transform is used. The principle component of the AC part of these collected patches is extracted as kernels or anchor vectors, the input is convolved with these kernels to get feature map. To avoid sign confusion, a bias with the largest norm of the patches is added to the feature map after convolution, and sent to the next PixelHop unit. The difference between PixelHop unit and PixelHop++ unit lies in Saab transform. In PixelHop unit, the input feature map or the input image is regarded as a whole. The patches are collected throughout the whole feature map, and they are doing PCA together to form the corresponding anchor vectors. The kernels with energy above a given threshold are selected, and the input feature map is convolved with these kernels to get the output feature map. While in PixelHop++ unit, Saab transform is conducted channel-wise. For each input feature map, the kernels are collected by channel instead of combining all channels together. There are two threshold in PixelHop++ unit, one is to decide which channels could be further split, the channels above this threshold are sent into the next stage of Saab transform for further extraction based on the energy of its parent node, those below this threshold are left as leaf node or discarded if their energy falls below the other threshold. For each stage of PixelHop++ unit, the leaf nodes are set as output, and the intermediate nodes are sent into the next stage of Saab transform. The whole structure is tree-like. The advantage of channel-wise Saab transform is the lower complexity, the parameters used in channel-wise Saab transform is smaller than that in Saab transform because of the

tree-like structure and the double threshold setting. Besides, in principle, the subspace formed by each principle component is orthogonal, they are uncorrelated, so the channel-wise Saab transform is more reasonable in comparison with Saab transform.

(4) LAG unit is the module where the label information is introduced and the dimension of feature vector extracted by PixenHop unit is reduced. because the task of the whole system is to classify, the label information should be considered. LAG unit also uses the SSL theory. In essence, LAG unit is a kind of linear regression, traditionally, feature vectors are directly connected to labels, the dimension is reduced rapidly because the difference between extracted feature vector and number of classes is huge, forcing the higher dimension information into a space with much lower dimension will cause information loss severely. In LAG unit, the feature vectors are clustered into several sub-classes according to the distance between feature vectors in the first hand, and the center of these clusters are recorded. Then, the probability of each vector belongs to each class is calculated, forming the target probability vector. Both information in labels and feature vectors are contained these target probability vector, the dimension of feature vector is thus reduced. These target probability vectors are used to do further classification. In this step, the feature vectors are firstly reduced to an intermediate subspace with dimension less than the original space but more than the number of labels. The regression process is smoother with this kind of intermediate subspace, so the performance could become better.

Problem 2: CIFAR-10 Classification using SSL

(a) Building a PixelHop++ Model

The module 1 is trained with 10000 training samples, 1000 for each single class and the module 2 and module 3 is trained with the whole dataset. The threshold 1 and threshold 2 in LAG unit is set to 0.001 and 0.0001 respectively. The other parameters are all set according to the Table 1 in tutorial, after some experiment, the effect of random forest is not as good as support vector machine, so in this part, support vector machine is selected as classifier. The training time is 4374s on Intel i7-8565U CPU. 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 1000 positions with smaller cross entropy, so in general, there is 1000 + 1000 + 1000 = 3000 parameters, but in my implementation, the last PixelHop++ unit only have 497 dimensions, so the number of parameters is 2497 in my setting. After feature selection by choosing the 1000 features with less cross entropy, the input dimension of each LAG unit is (50000, 1000), (50000, 1000) and (50000, 497), the output of each LAG unit are matrices with 150 dimensions in the second axis, which means the parameters in these three LAG unit is 150000, 150000 and 74550, the number of parameters in LAG unit is in total 375000. So, in total, there are 398372 parameters in the model. The classification accuracy on training set is 86.06%, and the accuracy on test set is 65.84%, the according parameters of support vector machine is C = 30, gamma = 1, the kernel of SVM is Gaussian kernel.

When testing the weak supervision problem, the number of training samples is reduced, to get a higher classification accuracy, the number of clusters in LAG unit is also reduced accordingly as the number of training samples reduced, the kernel in support vector machine is Gaussian kernel, the parameters are shown in Table 1, and the plot is shown in Fig. 2

Number of Training Samples	12500	6250	3120	1560
Number of Cluster in LAG	4	3	2	2
SVM parameters	C=0.5 Gamma=5	C=0.2 Gamma=1.3	C=0.005 Gamma=0.7	C=0.005 Gamma=0.001
Training Accuracy (%)	87.15	83.39	92.85	99.62
Test Accuracy (%)	58.67	53.25	45.65	31.29

Table 1 Training Accuracy and Test Accuracy of Weak Supervision

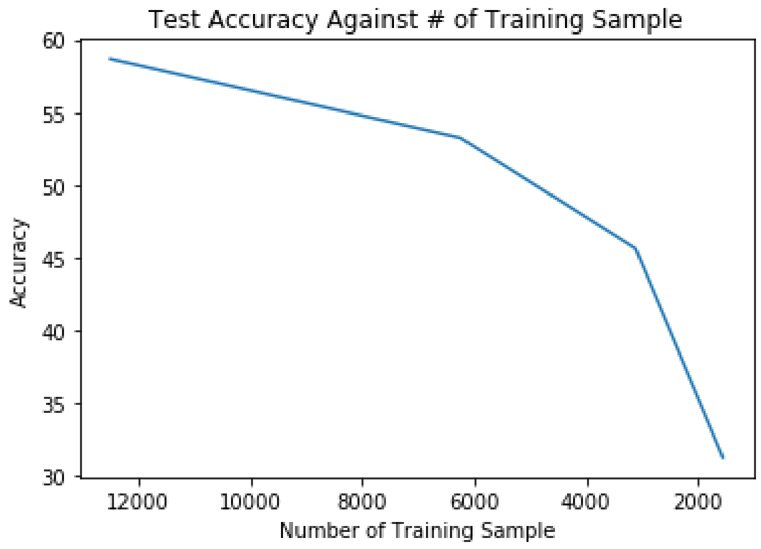
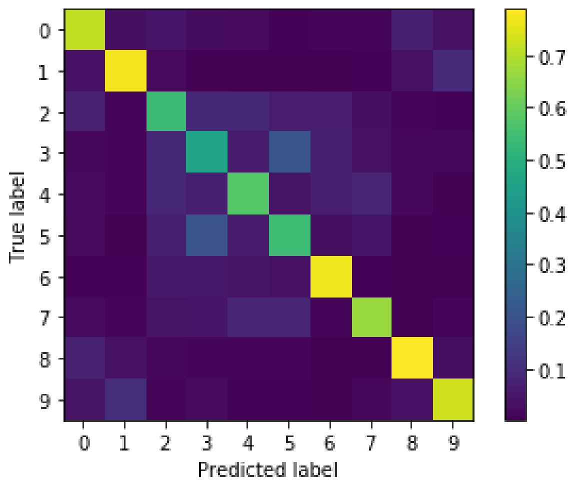
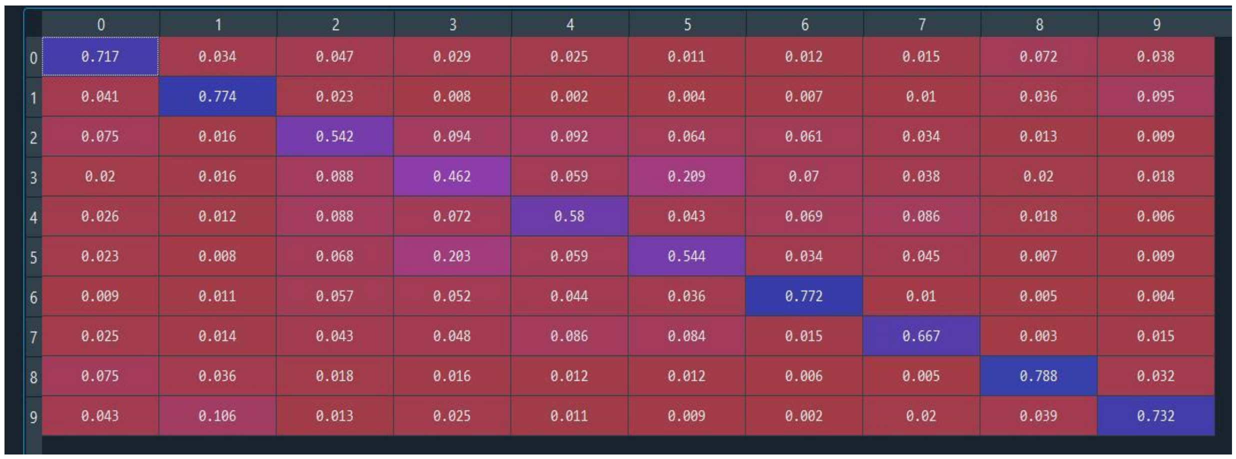


Fig. 2. Trend of Accuracy as Training Samples Decrease

From the plot we could see that the accuracy decreases from about 65% to 31% as the number of training samples fed into module 2 and module 3 decreases. This decrease is caused by weak supervision. Because the number of training sample is decreasing, the LAG unit and classifier could not extract the distribution with relatively smaller number of training samples. The performance of PixelHop++ unit with weak supervision is not as stable as PixelHop unit. One possible reason is that the feature extracted by PixelHop unit is across-channels, the kernels are totally uncorrelated because of the feature maps are sampled across spatial domain, kernels in different PixelHop unit have different number of channels. But in PixelHop++ unit, the kernels may be linear correlated, for example, in my implementation, each kernel in the second unit is 5*5, while there are 215 channels in the outcome, which means that there are 215 kernels in this PixelHop++ unit, while the full dimension of 5*5 filter is 25, so these kernels are linearly dependent. Even though they are operating though different channels, the feature may not be uniform and stable as PixelHop unit. So, the performance of PixelHop++ unit is not as stable as PixelHop unit in [1] as the number of training samples decreases.

(b) Error Analysis



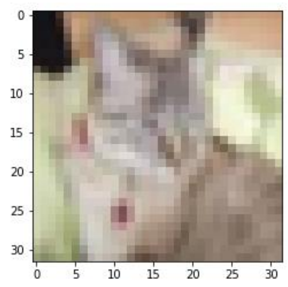


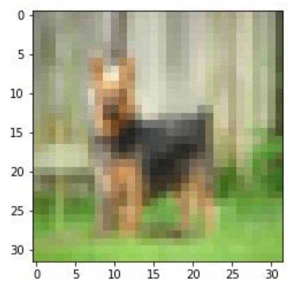
	0	1	2	3	4	5	6	7	8	9
0	0.717	0.034	0.047	0.029	0.025	0.011	0.012	0.015	0.072	0.038
1	0.041	0.774	0.023	0.008	0.002	0.004	0.007	0.01	0.036	0.095
2	0.075	0.016	0.542	0.094	0.092	0.064	0.061	0.034	0.013	0.009
3	0.02	0.016	0.088	0.462	0.059	0.209	0.07	0.038	0.02	0.018
4	0.026	0.012	0.088	0.072	0.58	0.043	0.069	0.086	0.018	0.006
5	0.023	0.008	0.068	0.203	0.059	0.544	0.034	0.045	0.007	0.009
6	0.009	0.011	0.057	0.052	0.044	0.036	0.772	0.01	0.005	0.004
7	0.025	0.014	0.043	0.048	0.086	0.084	0.015	0.667	0.003	0.015
8	0.075	0.036	0.018	0.016	0.012	0.012	0.006	0.005	0.788	0.032
9	0.043	0.106	0.013	0.025	0.011	0.009	0.002	0.02	0.039	0.732

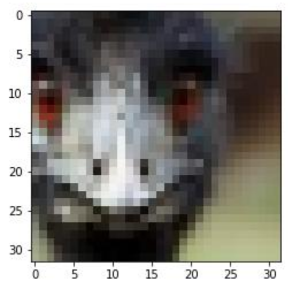
Fig. 3 Confusion Matrix and Raw Data

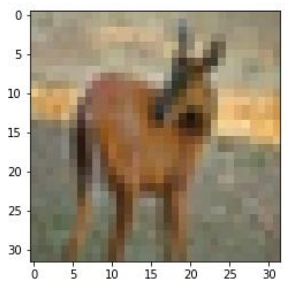
(1) The confusion matrix is shown in Fig. 3, as well as the raw data. From the confusion matrix and the raw data, class 8, ship, yields the lowest error rate, 21.2 samples are misclassified, and the most difficult class is class 3, cat, only 46.2% samples are correctly classified.

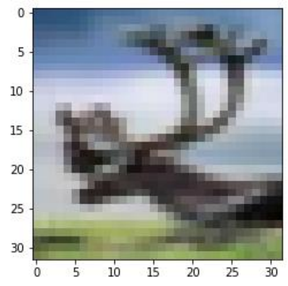
(2) Some misclassified pictures are shown in Fig. 4. The most confusing group is cat and dog. From general look like Fig. 4(a), the difference between cat and dog almost lies only in head, so recognizing these two classes is the most difficult task, their classification accuracy is among the lowest two classes and most of their misclassified data lies in the other class. Besides, from the confusion matrix, accuracy of recognizing different animals or different man-made objects is lower than when recognizing man-made object from animals and vice versa. There are several reasons causing these kinds of confusion. From Fig. 4(b), 4(d) and 4(f), we could see that their appearance is much more like the misclassified class. I could not even recognize Fig. 4(b) as a deer. This kind of difference is caused by the similarity of their shapes. Another kind of confusion is caused by the similarity of background, like Fig. 4(e) and (g), where deer is misclassified as airplane. Because the background of most airplane images is blue, and the horn of deer is kind of similar to the wings of airplane. But from Fig. 4(c), (f), (h), we could see that the feature extraction ability of the model is not so well. These pictures are not confusing, we human could clearly tell their classes.

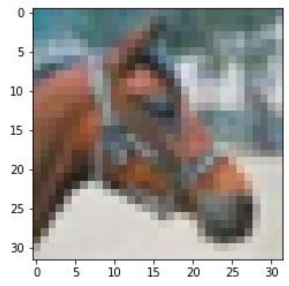

(a) cat -> dog

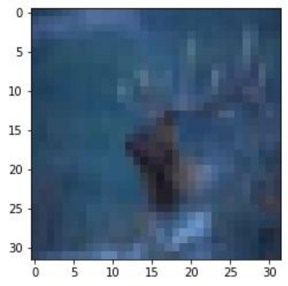

(b) deer -> dog


(c) bird -> automobile


(d) deer -> horse


(e) deer -> airplane


(f) horse -> bird


(g) deer -> airplane

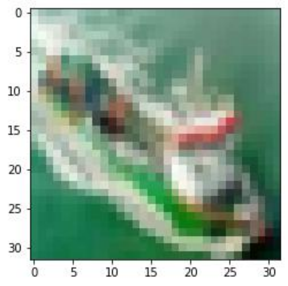

(h) ship -> cat

Fig. 4 Some Misclassified Images

(c) Because the PixelHop unit is an unsupervised feature extraction part, the task of the whole system is to classify, so the feature extraction part could be changed to some other static methods like Fisher Discriminant Analysis (FDA) to make the feature extracted by PixelHop unit more suitable for each class. But because there are a lot of background information in the first few PixelHop unit when using some small local patches to extract the information, so the information extracted by small local patches may be meaningless if assigned label directly, but in the deeper layer, each patch could have a larger receptive field, the feature map in the deeper layer could represent the feature of a certain class better, so if the deeper layer could be processed by a strong supervision algorithm, the feature will be more specific for each class. To avoid this problem, some semi-supervision algorithms are developed like [3], this algorithm combine PCA and a variation of FDA called Local FDA together, concatenating them by a parameter. In my opinion, if the former layers of PixelHop unit could use the assistant of label slightly, and the weight of supervised algorithm could increase as the depth increases, the effect maybe better than just using PCA to extract few features with the largest energy. Another suggestion for increasing the accuracy between difficult classes is to use data augmentation skills to increase the number of training samples in LAG and classifier. Because from the trend of accuracy when the number of training samples decreases as Fig. 2, if the number of training samples is increased, LAG unit could better fit the whole dataset. Besides, as the number of training samples increases, the number of clusters in feature selection unit should become larger to have a more suitable subspace to fit the dataset. But there is no need to feed more training samples into PixelHop++ unit, because the matrices converge rapidly as training samples increase [1].

Reference

[1] Chen Y, Kuo CC. PixelHop: A Successive Subspace Learning (SSL) Method for Object Classification. arXiv preprint arXiv:1909.08190. 2019 Sep 17.

[2] Chen Y, Rouhsedaghat M, You S, Rao R, Kuo CC. PixelHop++: A Small Successive-Subspace-Learning-Based (SSL-based) Model for Image Classification. arXiv preprint arXiv:2002.03141. 2020 Feb 8.

[3] 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.