EE569 Digital Image Processing

HOMEWORK # 6

Aagam Shah

USC ID-8791018480

USC Email – aagamman@usc.edu

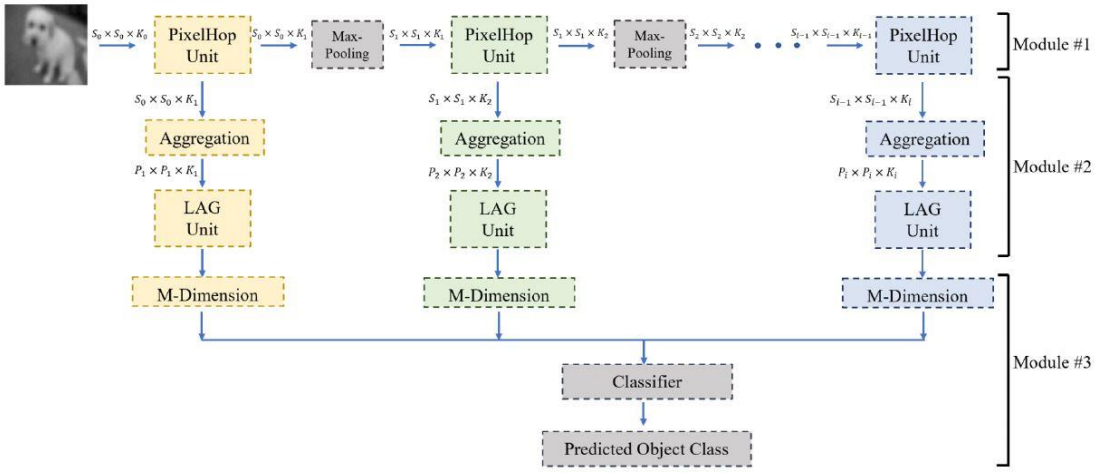
Submission Date – May 3, 2020

Problem 3: EE569 Competition --- CIFAR-10 Classification using SSL

1. ABSTRACT & MOTIVATION

As previously mentioned, that the Successive Subspace Learning technique is basically based on the 4 key parameters which were earlier stated as:
(1) Extension of the near to the far neighborhood progressively and successively
(2) Unsupervised dimension reduction via SSL technique
(3) Supervised dimension reduction via LAG unit
(4) To perform the concatenation of the features and then forming the decision.

Each of the above-mentioned points and their significance in the SSL technique in order to achieve optimum results is discussed and explained in brief in the Problem 1 and 2 of the homework 6. Now using the understanding of these concepts, I am set to improve upon the testing accuracy by putting into action and extracting the usefulness of the properties of each units of the Pixel Hop++ to achieve the better performance as compared to the previous one. The architecture of the SSL system is almost same as that used for the previous question just the only change, I tried is the cascading of more hop units to generalize model and improve upon the accuracy along with keeping in mind the model size or the model parameters. The general block diagram of the Pixel Hop using the SSL technique can be explained as follows:



Briefly explaining about the main role of the 3 modules of the Pixel Hop using the SSL technique can be summarized as follows:

Module 1: This module is basically the cascading of multiple Pixel Hop units whose main role is to basically determine the features of the near to the farthest neighbors of the selected pixels via multiple pixel hop units. In simple words it's the connection of the features of neighbors from the previous unit of the target pixel and its neighboring pixels in order to generate the union of them, which leads to increase in dimensions. So, in order to control the growth of the dimensions we perform unsupervised dimensionality reduction which is obtained by the subspace approximation technique known as Saab Transform. This module is basically the combination of the neighborhood construction and the subspace approximation.

Module 2: This module is basically the aggregation unit along with the supervised dimensionality reduction using the LAG unit. In this module the spatial dimension reduction is performed before giving the output to the pooling layer of the pixel hop unit which can be done either by max, min or mean pooling. The spatial dimension obtained after the aggregation unit is then given to the LAG unit for the supervised learning procedure of feature reduction.

Module 3: Concatenation of features from all the Pixel Hop Units and the performing Classification.

Here, in this module the N features obtained from the K pixel hop units are basically concatenated together to get the N x K features which are then fed to the classifier module in order to train the multiclass classifier and then use the trained classifier to predict the labels of the test data and perform the object recognition process and then finally get the testing accuracy, a parameter to judge how much accurately the model has been trained.

2. LOGIC & DISCUSSION

The various attempts of experiments performed in order to improve upon the test accuracy of the model by varying the hyper parameters and the significance of each can be listed as follows:

1. **Pooling Layers**: As we know that pooling operation is basically used to perform dimension reduction in order to reduce the variance and the computation complexity. I basically tried implementing all types of pooling layers i.e., the Maximum pooling, minimum pooling as well as the mean pooling layer. Out of the three pooling layers, the maximum pooling layer yields the most optimum result with highest accuracy of all three of them. In general, we cannot comment that a particular type of pooling method will perform better over the other one. The pooling layer choice depends on the type of data we are dealing with.

   Maximum pooling basically selects the brighter pixels or the high frequency components of an image.

   Mean pooling basically smoothens out the image which in turn results in smoothening of the sharp edges of brighter patches of an image.

Minimum pooling basically selects the dim pixels or may be low frequency components of an image.

2. Next thing I experimented was with the **number of pixel hop units**. Initially, I tried 3-pixel hop units using max pooling layer which gave me the highest accuracy of all my experiments with **spatial neighborhood of size 5 x 5 and stride as 1**. Next variation I chose is to cascade more number of pixel hop units so I added one more pixel hop unit i.e., 4 pixel hop unit with spatial neighborhood as 3 x 3 and stride as 1, but this in turn reduced the accuracy due to the decrease in the size of the spatial neighborhood. I choose 3 x 3 instead of 5 x 5 spatial neighborhood size on increasing the number of pixel hop units basically taking into consideration the model size or the total number of model parameters as well as the computation complexity and time.
3. Then I experimented with the **alpha value of the LAG unit**. Initially I tried with the same alpha value as given in the parameter table i.e., 10. As from the Pixel hop paper alpha is basically the hyper parameter which *determines the relationship between the Euclidean distance and the likelihood of a sample belonging to a particular cluster.* [1] *So, basically if we increase the alpha value the decay rate of the probability increases with the distance, which implies that shorter the Euclidean distance the larger is the likelihood.* [1] So, according to the above explanation for the alpha value the lesser the alpha value the larger is the likelihood of the sample belonging to that cluster and the probability decays at slower rate with the Euclidean distance. So, I chose the alpha to be 5 and in some experiments as 7.
4. Then next hyper parameter varied was **the number of the centroids per class of the LAG unit**. The relation of the number of centroids per class of the LAG unit with accuracy is that it is directly proportional. As the total number of centroids is the product of the clusters and classes, where *probability vector of the clusters indicates the likelihood of an input which may belong to the subspace spanned by the centroids of that particular class.* [1] I experimented with number of centroids as 5 and 7, also taking into consideration the model size I wisely choose the number of centroids per cluster.
5. The next thing I experimented with is **the energy threshold values** i.e., TH1 and TH2 where TH1 is nothing but the energy threshold for the intermediate nodes and the TH2 is the energy threshold for the discarded nodes. So, I tried 4 variations for the energy thresholds i.e., High TH1 and Low TH2, Low TH1 and High TH2, High TH1 and High TH2 & Low TH1 and Low TH2. All the variations basically resulted in the deterioration of the accuracy from the given threshold values. The given threshold values are found to be in the optimum range. On increasing the threshold values the model size increases. Also, the test accuracy decreases as the threshold value decreases thereby reducing the model size significantly. So the values for the energy thresholds TH1 and TH2 should be chosen very wisely for the optimum results and the better performance of the model.

6. The selection of the **Ns** features also plays an important role in determining the test accuracy. I observed that the test accuracy decreases with small amount with the decrease in the number of selected features. This happens basically due to the less generalization of the model. Selecting Ns = 1000 yields more accuracy as we chose the top 1000 features with more discriminant power as compared to choosing the top 50% features having less discriminant power. Also, if I choose Ns = 500 the accuracy dropped but marginally, which is acceptable in a tradeoff between the accuracy and the model size with having more crucial features to train the model. So, I choose Ns = 1000 top features having the more discriminant power to train the model keeping into consideration of number of model parameters.
7. Last but not the least thing which I experimented on was **the classifier** I choose. I experimented with two types of classifiers namely, the Random Forest Classifier with XGBoost and the Support Vector Machine. The first thing I would like to mention is that the inference time of Random Forest is less than the SVM technique thus making the choice of RF classifier more convincing as compared to that of the SVM classifier and the second thing is that it is worth choosing RF classifier as compared to that of the SVM technique although SVM being more sophisticated process takes more execution time with marginal difference in accuracy. Although, the RF classifier yields high accuracy as compared to the SVM classifier technique. So, here in this approach I choose Random Forest Classifier for classification of the multiclass CIFAR 10 dataset.
8. In order to decrease the running time I also tried an alternative technique of K-means clustering which is nothing but the Bin method as discussed in the discussion session. Here we basically compute the histogram of N using the bin method.

$$mc = (1,2,1,3,3,3,2,1,3,2)$$

So basically, to achieve high discriminant power we lower the cross entropy which can be obtained by considering only the top pixel values which is used to localize the salient region and the other rest of the pixels are made zero. The operation of Spatial averaging is carried out at each spectral location and only the first K spectral dimensions are retained which has the high discriminant power or low cross entropy values. This is basically performed for all the pixel hop units to perform image classification.

3. EXPERIMENTAL RESULTS

Classification accuracy with Full and Weak supervision:

- **Full Supervision: (Ns = 1000, alpha = 5, Number of centroids = 7)**

  Depth = 3
  Testing accuracy = **64.66%**

  Depth = 4
  Testing accuracy = **65.82%**

- **Weak Supervision: (depth = 4, alpha = 5)**

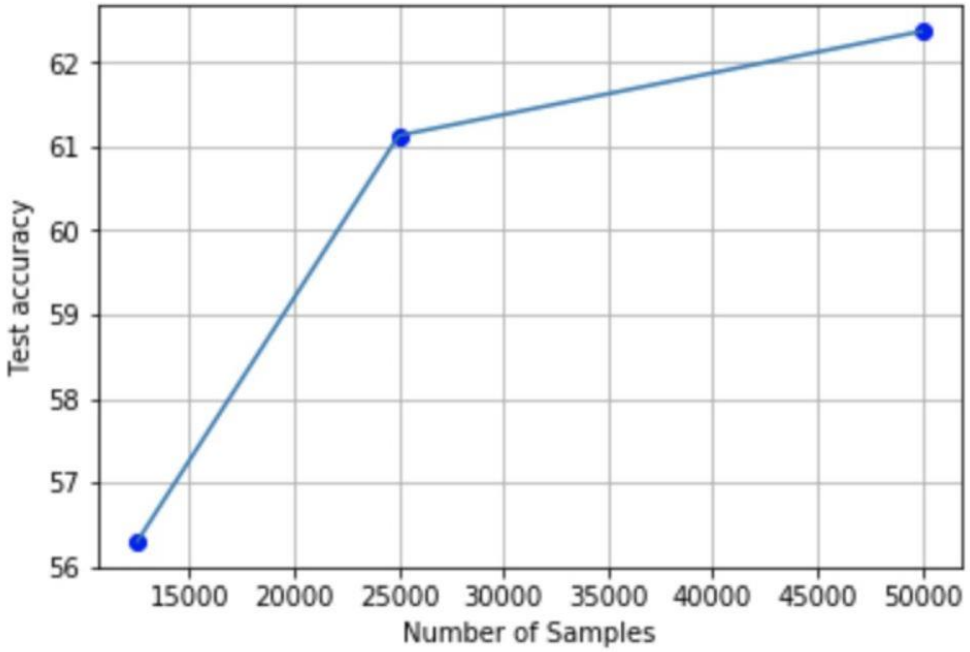
  For (1/2) of training images,
  Testing accuracy = **62.37%**

  For (1/8) of training images,
  Testing accuracy = **59.32%**

  For (1/4) of training images,
  Testing accuracy = **61.41%**

  For (1/16) of training images,
  Testing accuracy = **56.74%**

  The test accuracy curve with the training models for different training images is as follows:



On comparing it with the curve obtained in the HW5 Problem 2 using the back-propagation technique we find that although the accuracy obtained by the BP-CNN is much more but at the cost of high model size i.e., having more number of model parameters, whereas the accuracy obtained by the Pixel Hop++ technique is although less as compared to the BP-CNN but with significantly less number of model parameters, low model size which seems to be a reasonable tradeoff. Also, as the FF-CNN is simple transformation of spatial and spectral space by using only statistics and linear algebra, so it is white box where all parameters are transparent and clear, while BP-CNN uses convex optimization which is more complex in nature and behaves like a black box where it is more unclear and difficult to fix the network if something goes wrong as we are not known which parameter we need to tune in order to fix the network error.

Running time:

The best training time = approximately **1 hr 15 mins.** (using KMeans)
Approximately **25 – 27 mins.** (using bin method)

The best inference time = approximately **7-8 mins.**

Model size: (related to TH1 and TH2 as well as Ns)

We calculate model size for the 3 hops which is given in tabular form as follows:

Module	Hop 1	Hop 2	Hop 3	Total
Module 1	3075	5500	13050	21625
Module 2 (Ns = 1000)	50000 (1000*50)	50000 (1000*50)	26100 (1000*50)	126100
Module 2 (Ns = 50%)	200900 (4018*50)	137500 (2750*50)	13050 (261*50)	351450
Module 2 (Ns = 25%)	100450 (2009*50)	68750 (1375*50)	6525 (130.5*50)	175725

The total number of parameters or the Model size for (Ns = 1000) is **147,725.**

On comparing the model size of the BP-CNN obtained in the Problem 2 of HW 5 which was around 300k parameters but with accuracy as high as around 90% while here we have drastic reduction in the model size almost half of what we got for BP-CNN with approximately 25% less accuracy. To summarize I would like to say that using Pixel Hop++ technique based on SSL we obtain 25% less accuracy with 50% reduction in model size as compared to that of the BP-CNN technique which is a reasonable tradeoff.

EE569 Digital Image Processing

References:

1. Pixel Hop: A Successive Subspace Learning (SSL) Method for Object Classification, technical research paper by, Yueru Chen and C. -C. Jay Kuo.
2. PIXELHOP++: A SMALL SUCCESSIVE-SUBSPACE-LEARNING-BASED (SSL-BASED) MODEL FOR IMAGE CLASSIFICATION, technical research paper by Yueru Chen_, Mozhdeh Rouhsedaghat_, Suya You+, Raghuveer Rao+ and C.-C. Jay Kuo_
3. Discussion slides and Lecture Videos