Spring 2018	Report#4 - 01/28/2018	Tianxiao Zhang

I. Tasks achieved Last Week (***, **, *: order of priority)

■ Project: Content-Adaptive Saak Transform

Purpose: Using content-adaptive method to further improve the accuracy of the MNIST test data.

Method: Using clustering method to cluster the content and doing Saak transform in each cluster and then combine them together as feature for each image.

Note: The time cost for running the codes should be controlled because what we want to do is beat CNN, not just running the codes without considering the time. If the time cost is long, the method is not that good.

Slogan: Beat CNN!

II. Feedback and Interaction

■ ***Prof. Kuo's feedback***

-

■ ***Discussion***

III. Report

Last week, I clustered the images based on their labels and got very good results. However, we cannot utilize the test labels to cluster the test images because we do not know the test labels. So we need to find a new method to solve this problem.

1. Using train data clustered by labels as training

In this report, suppose that the accuracy for train data is 0.9966 and the accuracy for test data is 0.9848 for original Saak transform under certain parameters, and I would not change the parameters in the following experiments.

First I utilized the train data clustered by the labels as training. For example, I used all labeled "0" train images as PCA fitting and transformed the original train images and test images. And the results are shown below:

label	0	1	2	3	4	5	6	7	8	9
Train accuracy	0.9964	0.9965	0.9962	0.9962	0.9965	0.9963	0.9964	0.9966	0.9965	0.9964
Test accuracy	0.9851	0.9847	0.9854	0.9854	0.9850	0.9858	0.9849	0.9851	0.9851	0.9852

Using all "5" train images as PCA fitting, we can get the highest test accuracy 0.9858. However, it seemed that the improvement is not useful enough because of its small value.

Then I did another experiment. In this experiment, I treated all labeled "0" train images as "0" and all labeled non-"0" images as "1". And then I utilized these changed labeled dataset as PCA fitting and transformed the original train images and test images. Then I can get the following results:

PCA	32		64		128		256	
label	0	1	0	1	0	1	0	1
Train accuracy	0.9978	0.9978	0.9964	0.9966	0.9919	0.9920	0.9840	0.9842
Test accuracy	0.9823	0.9820	0.9851	0.9849	0.9830	0.9832	0.9769	0.9773

In the table, PCA is the components we keep after F-test in the final classification stage. And the label indicates that we utilize labeled "0" as PCA fitting or labeled "1" ("2"—"9") as PCA fitting. It seems that

the results are not good enough either.

After that, I still utilized the changed labeled dataset as PCA fitting, but I did not transform the original train images and test images, instead, I transform the changed labeled train images and test images just as the above dataset.

PCA	32		64		128		256	
label	0	1	0	1	0	1	0	1
Train accuracy	0.9978	0.9978	0.9964	0.9966	0.9919	0.9920	0.9840	0.9840
Test accuracy	0.9823	0.9820	0.9851	0.9850	0.9827	0.9833	0.9771	0.9771

The results are shown above. This is still not a very good way to boost the performance.

2. Saak transform is better when the number of classes are reduced?

What the results will be when we reduce the number of classes for both train and test data? To do this experiment, I utilized the train labels and test labels to pick out all the images belonging to that specific label. To simplify the experiment, when classes are 2, I picked out all labeled “0” and “1” images for both train and test images to test; when classes are 3, I picked out all labeled “0”, “1” and “2” images for both train and test images to test and so on. Even though in this experiment I still used the test labels, but I utilized the test labels just for picking up for specific labels and reducing the number of classes instead of clustering. After I reduced the number of classes for both train images and test images, I did the original Saak transform.

classes	2	3	4	5	6	7	8	9	10
Train accuracy	1.0000	0.9997	0.9995	0.9993	0.9991	0.9987	0.9981	0.9978	0.9966
Test accuracy	0.9991	0.9968	0.9966	0.9965	0.9939	0.9923	0.9904	0.9890	0.9848

It seems that with the number of classes increasing, the accuracy for both train images and test images will drop down. To further test the property, I did another experiment.

For classes are 2, I picked up “7” and “9”, “5” and “8”, “1” and “6” to do the experiment the same as the first experiment in the report. First I did the original Saak transform for these 3 dataset and then I utilized the train images clustered by label as PCA fitting. For example, “7” and “9”, first using “7” as PCA fitting and then transformed (“7”, “9”) train images and test images. Then using “9” as PCA fitting and

then transformed (“7”, “9”) train images and test images. The results are shown below:

classes	(7,9)	(5,8)	(1,6)
Train accuracy	0.9982	0.9994	1.0000
Test accuracy	0.9917	0.9968	0.9990
cluster	**7**	**5**	**1**
Train accuracy	0.9982	0.9993	1.0000
Test accuracy	0.9917	0.9957	0.9986
cluster	**9**	**8**	**6**
Train accuracy	0.9983	0.9994	1.0000
Test accuracy	0.9921	0.9962	0.9990

From the results above, for some labels, it will work and for some labels, it will not work. And even though it works, the improvement seems not enough.

So I think maybe Saak transform is suitable for small number of classes, then how to reduce the number classes and do the Saak transform separately in each group maybe is the key point for the following work.

References

[1] C.-C. Jay Kuo, “Understanding convolutional neural networks with a mathematical model,” the Journal of Visual Communications and Image Representation, Vol. 41, pp. 406-413, November 2016.

[2] C.-C. Jay Kuo, “The CNN as guided multi-layer RECOS transform,” the IEEE Signal Processing Magazine, Vol. 34, No. 3, pp. 81-89, May 2017.

[3] C.-C. Jay Kuo and Yueru Chen, “On data-driven Saak transform,” arXiv preprint arXiv: 1710.04176 (2017).

IV. Plan for the next week (***, **, *: order of priority)

- Reducing the number of classes and doing Saak transform in each group separately

V. Milestone