Spring 2018	Report#5 - 02/04/2018	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: First, we should do the original Saak transform and get the class probability for each test images.

Then we compare the top 1 class probability of top 2 class probability and calculate the ratio of these two number. If the ratio is larger than the threshold, then we use the class of the maximum probability as the test label and the process is done. If the ratio is less than or equal to the threshold, then we should do the second stage Saak transform to further compare these two confusion classes.

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

This week, I did the new method to further improve the Saak transform.

1. Two "stages" Saak transform

For the original Saak transform, the test accuracy is already 98.5%, which is very high. So in the new method, we utilized the original Saak transform as our first stage.

After the original Saak transform, we can get the probability for each class in each image. Then in the second stage, we test each test images again by comparing the maximum class probability and the second maximum class probability. If the ratio of the maximum class probability and the second maximum class probability is larger than the threshold we set, then we do not need to do further and just use the class with maximum probability as the label for that test image. However, if the ratio of the maximum class probability and the second maximum class probability is less than or equal to the threshold we set, we have to do further comparison to select which one is better. We had better choose the threshold to let the number of retesting images be around 300.

For example, if we need to further compare the label "7" and "9", we need to utilize all labeled "7" train images as the kernel and calculate the Saak coefficients of all labeled "7" and "9" train images, and also the retesting test images. And then we use all labeled "9" train images as the kernel and do the process again. Thus we have two pipelines: one is using "7" as the kernel and another is using "9" as the kernel. After that, we calculate the energy of each coefficient (the square of each coefficient) and get the mean of the energy for each coefficient for all labeled "7" train images and do the same process for all labeled "9" train images. Thus we compute the absolute difference between "7" and "9" for each corresponding coefficient. And we pick up all big differences and apply it to the retesting test image. This process is replacing the F-test to pick up the coefficients which can mostly distinguish the two classes.

2. The results for two "stage" Saak transform

I did many ratio cases in the experiments, but I only show 4 cases in the below:

ratio	1.5	2	5	10
Number of retest images	55	102	240	355
Number of correct labeled images in the retest images	26	55	148	240
Wrong → Correct	13	15	18	17
Correct → Wrong	6	7	27	38
Test accuracy	0.9856	0.9858	0.9840	0.9828

When the ratio is small, the test accuracy is truly improved because we correct more wrong labeled images than making the correct labeled images wrong. But when ratio is small, the number of retest images seems no enough. However, when the ratio is increased, the test accuracy reduces because we the number we make the correct labeled images wrong is larger than the number we correct the wrong labeled images.

Why? To further explore the reason why this method did not work so well, I did another interesting experiment. We know that the new method is to replace F-test for picking up the coefficients which can mostly distinguish different classes. So I picked up the good coefficients and bad coefficients to compare. For example, I picked all ratio larger than 0.2 as “good” coefficients, and I also picked all ratio less than 0.2 as “bad” coefficients as comparison. If this method really works which is that it can pick up truly good coefficients, then the results using “good” coefficients should be much better than the results using “bad” coefficients. In this experiment, I utilized 2 as the ratio.

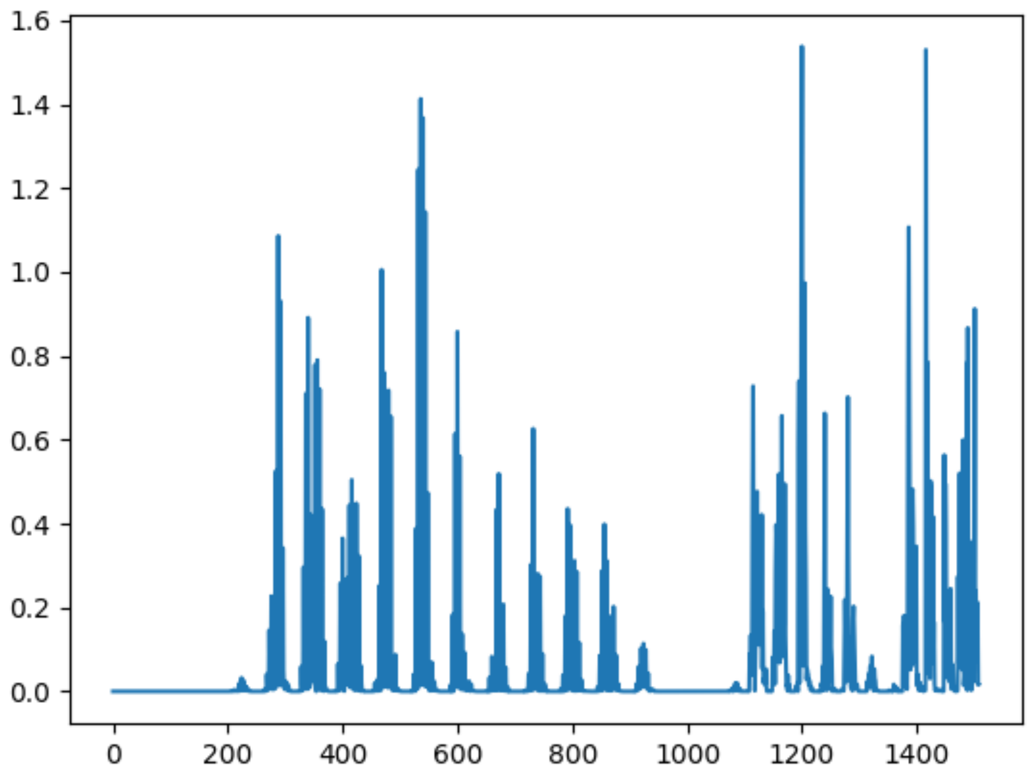
ratio	<0.1	>0.1	<0.2	>0.2
Wrong → Correct	13	13	16	9
Correct → Wrong	27	19	24	21
Test accuracy	0.9835	0.9843	0.9841	0.9837

We can obviously see from above that the test accuracy of “bad” coefficients is better than that of “good” coefficients. It shows that the method cannot pick up the coefficients which mostly distinguish the classes.

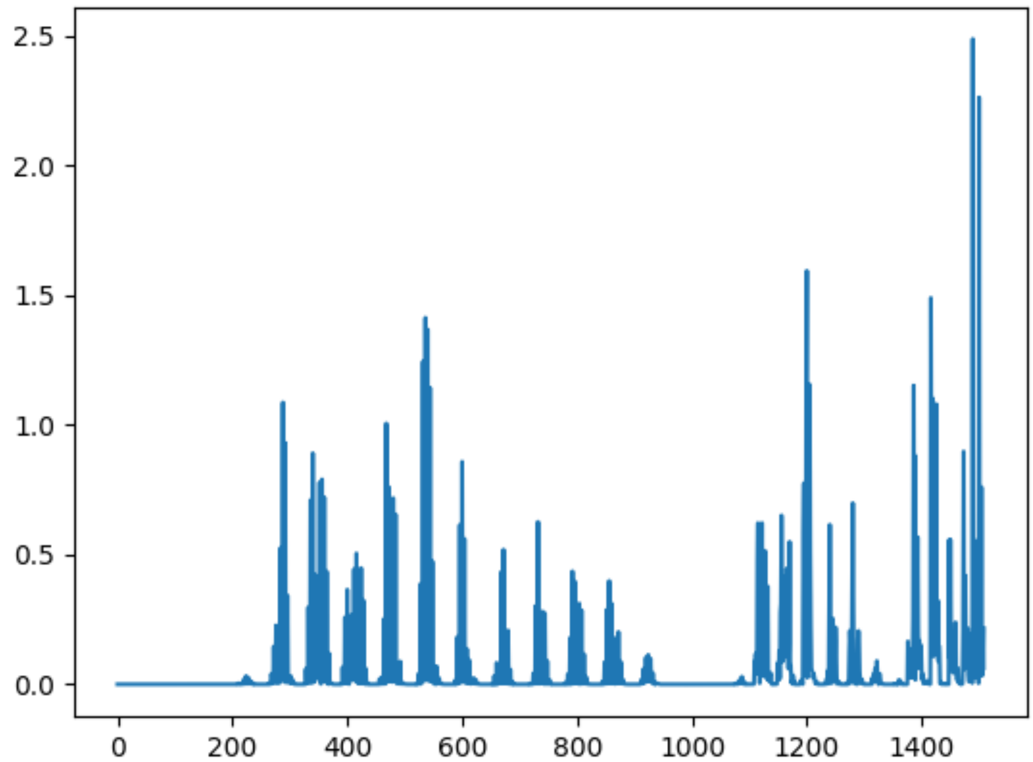
3. The results from the data

I did another experiment to see the data distribution.

For this experiment, I utilized “7” as kernel to get the coefficients of “7” and “9”, and also I used “9” as kernel to get the coefficients of “7” and “9”.



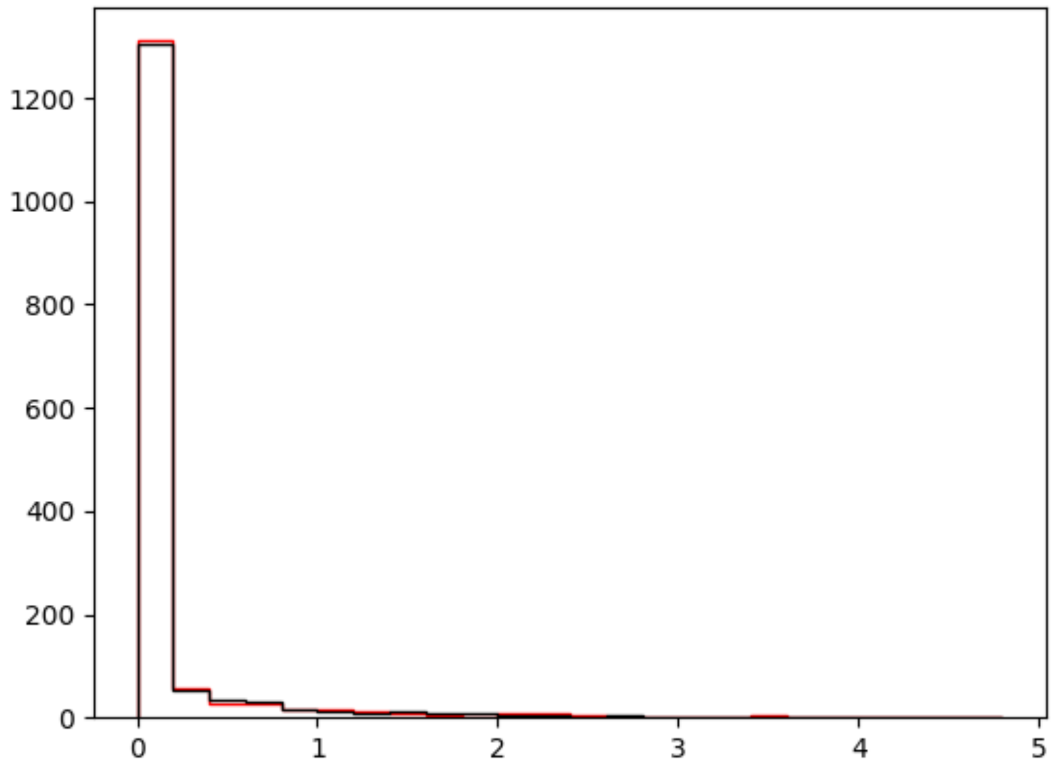
Above image shows that “7” as the kernel and the energy differences for each coefficients between “7” and “9”.



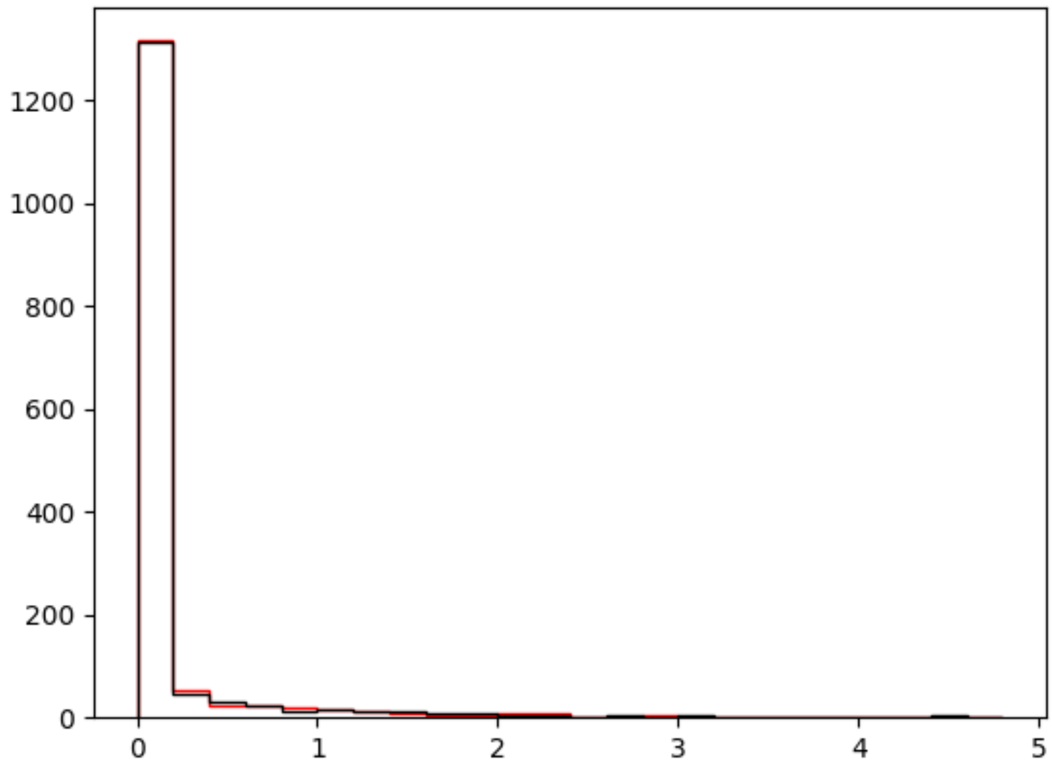
Above image shows that “9” as the kernel and the energy differences for each coefficients between “7” and “9”.

They all have coefficients which have big differences. However, when we pick up the coefficients which have these big differences, it did not work.

When I plotted the histogram of the energy mean for “7” and “9”, most of the energy is overlapped.



Above shows “7” as kernel, the histogram of the energy means for “7” and “9”.



Above shows “7” as kernel, the histogram of the energy means for “7” and “9”.
We can see that the overlapping is serious, so this method seems not that good to pick up the coefficients.

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)

- Find some new methods to improve the Saak transform.

V. Milestone