Spectral Leakage and Rethinking the Kernel Size in CNNs

Final Report

Wenhan Liu
Shuo Zhang
Zhiqi Bei

SYDE 671 - Advanced Image Processing
University of Waterloo
December 10, 2021
Prof. David Abou Chacra

Introduction and Scope of reproducibility

The original paper discusses the impact of spectral leakage on CNNs and shows that a small kernel size will make CNN more susceptible to spectrum leakage through the experiments of using two different kernels, one is a normal model with the kernel of size 7 and the other is a kernel of size 11 but with hamming windows. Then by comparing their performance when applying them in a single layer, the model with a window has higher accuracy.

The original paper recommends using a larger size kernel and window function to reduce the impact of spectral leakage, thereby improving the accuracy of classification. The original paper also used experiments to support their claim. They build three models: 1. A model with hamming window and using the 3x3 kernel only on the first layer of the model. 2. A model with hamming window and using a 7x7 kernel only on the first layer. 3. A model using the hamming window on all the layers and with the kernel size of 7x7. And test those models on CIFAR-10, CIFAR-100, Fashion-MNIST, MNIST, and ImageNet datasets. The result is that the third model has an improvement in accuracy in all the datasets.

According to the original paper, the experiments we reproduced in this work is:

1. Using the network with 7x7 kernel size and the hamming window in all its layers will have a higher classification accuracy when applying it to the CIFAR-10 and CIFAR-100 [5] datasets.
2. Using the network with 7x7 kernel size and the hamming window in all its layers has a higher performance on the Fashion-MNIST dataset.
3. Using the network with 7x7 kernel size and the hamming window in all its layers and shows no significant performance improvement on the MNIST dataset. However, the model shows an improvement in accuracy when subsampling the input image in the MNIST dataset from 28x28 to 14x14.
4. Using the network with 7x7 kernel size and the hamming window in all its layers result an accuracy improvement on the ImageNet dataset.

Background and Related Work

Spectral Leakage -- appears when the number of periods of signal passed to discrete Fourier transform is not an integer. When applying a window function to the periodic signal, there is a chance that the cut window is not periodic and the sharp edges (sinc like function) will generate a bunch of frequencies that distorts the filtered image around the signal.

There are some approaches that try to solve this problem. In this paper [11], a sampling operator composed of windowing and sampling operations is used to encode the sparse frequency representation of a filter, and use it to reconstruct the window function that depends on the size of every object, which suppresses the leakage problem. In the selected paper, instead of a dynamic window function, what it does is to use a fixed size kernel with a hamming window that helps tackle the spectral leakage [6, 7, 8] issue.

Methodology

Since the central claim is about increasing classification accuracy with the use of the Hamming window and larger kernel size in order to alleviate leakage in CNN architectures, the method is very much the same. In order to reduce unwanted frequency components, the use of a standard Hamming window [2] was proposed, and in CNNs, the Hamming window would be used in the convolution operations.

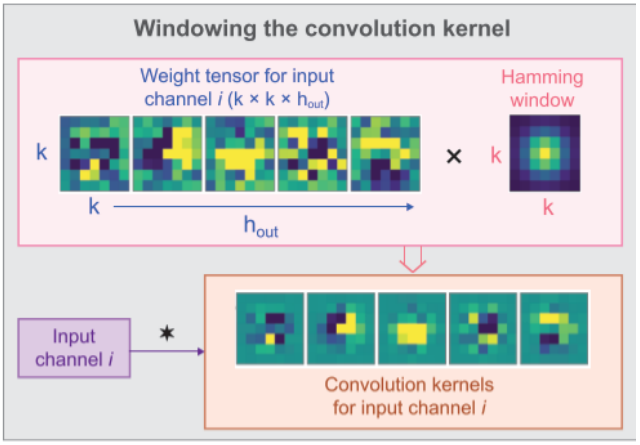


Figure 1. This shows a single input channel i that got convolved with hout distinct *k* x *k* kernels. And the k x k kernels are generated by the multiplication of each *k* x *k* slice of weight tensor with the *k* x *k* Hamming window.

And the implementation of the Hamming window is rather simple, it could be done by multiplying each of the two-dimensional *k* x *k* kernel in a convolutional layer with the *k* x *k* Hamming window function [1] (Fig. 1).

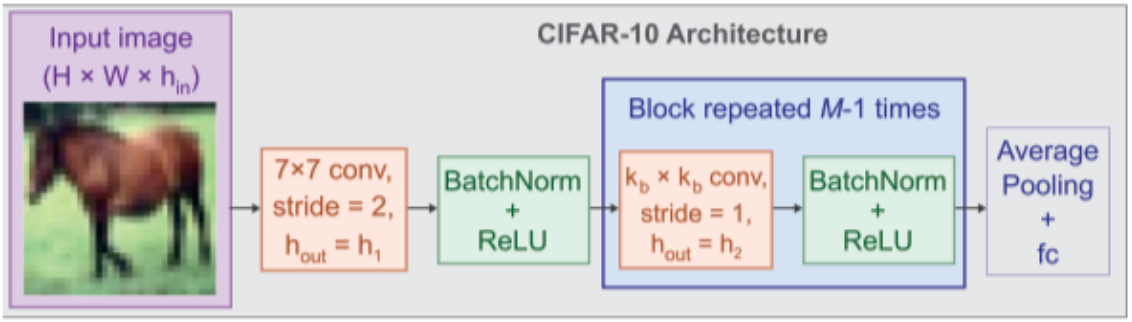


Figure 2. This shows the CNN architecture used. A for loop was used for the blue block of code to construct the layers, and this is useful in varying the number of convolutional layers for the CNN. With the first layer having kernel size of 7 x 7, the kernel size for all other convolutional layers is kb, which is sort of a hyperparameter that can be set by the user and could influence the classification accuracy. Also, by the variation of the depth of the CNN model, the final classification accuracy of both the normal CNN and the windowed CNN can be influenced.

For this project, MNIST, Fashion-MNIST, CIFAR-10, CIFAR-100, and ImageNet were used to test the correctness of the claim. And the CNN architecture used is shown in Figure 2. With a simple use of a for loop, the blue block of layers could be repeatedly constructed as a measure of controlling the depth of both the normal and windowed CNN, with the Hamming window process happening inside the orange block of codes. And just as a test, since the paper stated that using the Hann and Blackman window would result in similar results as using the Hamming window, we used those two windows on the MNIST [3] dataset just to test it out. The construction of those two windows was very much the same as the construction of the Hamming window, it could be constructed just by calling np.hanning and np.blackman with the replacement of np.hamming.

Results

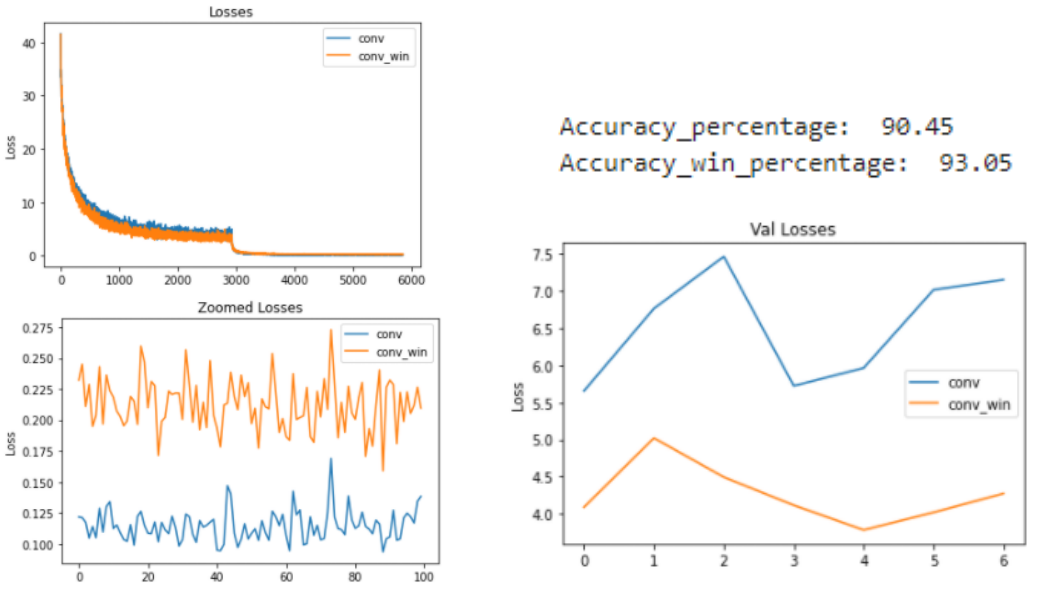


Figure 3. This shows the training loss, validation loss, along with the classification accuracy for both the normal model (blue line) and the windowed model (orange line) when performing on CIFAR-10. Both the two models were trained for 150 epochs with kernel size of 7, and this result matches the claim in the paper that the windowed model would outperform the normal model with a higher accuracy.

For CIFAR-10, our test got a result that matches the claim of the paper. With the training of both a normal CNN and a windowed CNN, although the training loss for windowed CNN is a bit higher, however, both the validation loss and final classification accuracy was higher for windowed CNN (Fig. 3).

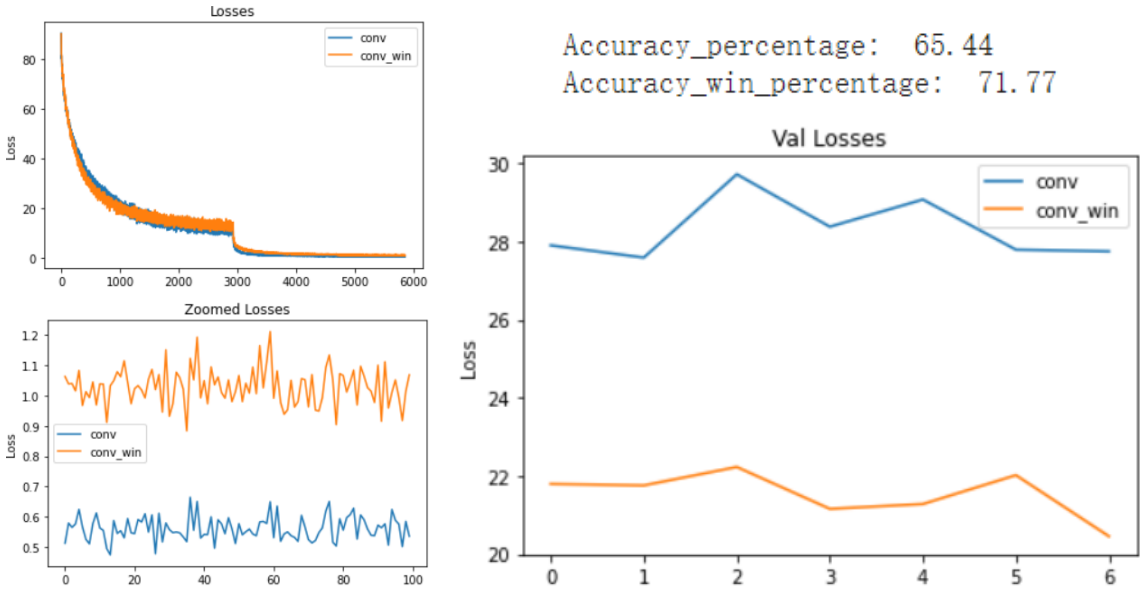


Figure 4. This shows the training loss, validation loss, along with the classification accuracy for both the normal model (blue line) and the windowed model (orange line) when performing on CIFAR-100. Both the two models were trained for 150 epochs with kernel size of 7, and this result matches the claim in the paper that the windowed model would outperform the normal model with a higher accuracy.

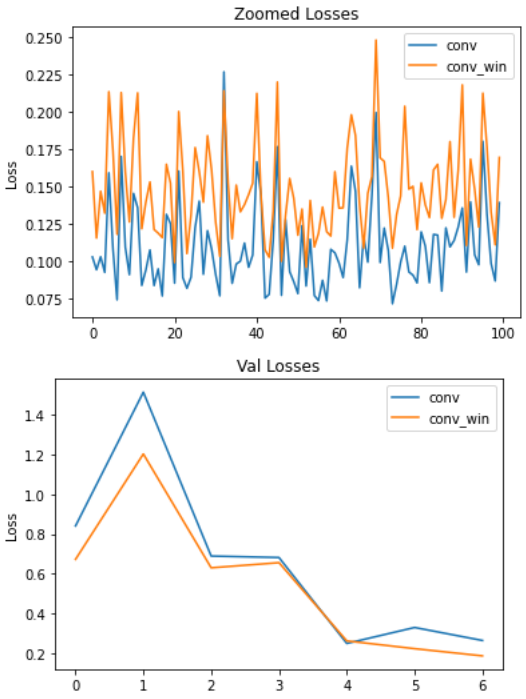


Figure 5. This shows part of the training loss for both the normal and windowed model on the original MNIST dataset. And the training loss and validation loss for these two different models were very close, which means that there does not exist any performance increase for the windowed model this time.

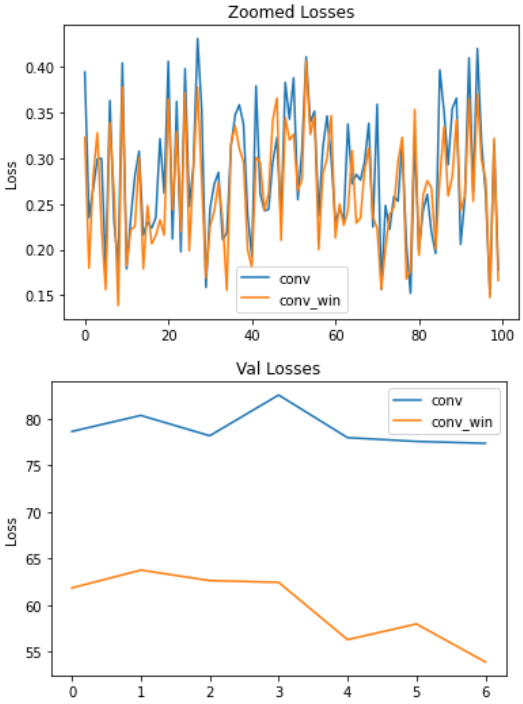


Figure 6. This shows part of the training loss for both the normal and windowed model after subsampling each input image from 28 x 28 to 14 x 14. It's obvious from the graph that the validation loss for windowed model is much lower than that of the normal model.

For CIFAR-100, the test result matches the claim of the paper as well. And this time, the validation loss for the windowed model is much lower than that of the normal model, and compared with the test performed on CIFAR-10, the accuracy for the windowed model is also much higher than that of the normal model (Fig. 4).

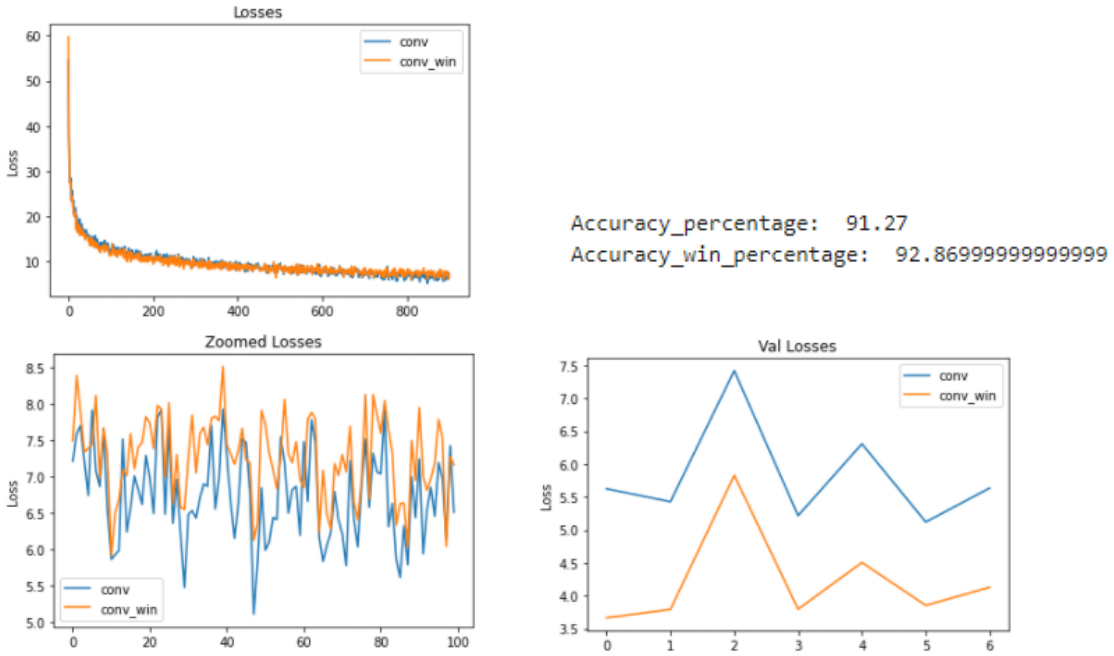


Figure 7. This shows the training loss, validation loss, along with the classification accuracy for both the normal model (blue line) and the windowed model (orange line) when performing on Fashion-MNIST. Both the two models were trained for 20 epochs with kernel size of 7, and this result matches the claim in the paper that the windowed model would outperform the normal model with a higher accuracy.

For MNIST, the result does match the claim of the paper. When performing on the MNIST dataset, as it contains less natural images, the influence of windowing would be less prominent as not all frequency components are well presented in the training set. Hence, there aren't much performance improvements for the windowed model compared with the normal model (Fig. 5). But after subsampled each image in MNIST from the size of 28 x 28 to 14 x 14, according to the paper, the relative magnitude of high frequency components were increased, and this time, both the training loss and validation loss tend to be lower for windowed model than normal model, and the validation loss for windowed model is much lower than that of the normal model (Fig. 6). However, there's a problem with this subsampling of images in the MNIST dataset that we will discuss in the Procedural / Data challenges.

Finally, for the Fashion-MNIST [4] dataset, the result also matches the claim in the paper. With both models having similar training losses, the validation loss for the windowed model is lower than the validation loss for the normal model. And the accuracy for the windowed model is also higher (Fig. 7).

The paper also talked about performing the two models on the ImageNet dataset, however for the reproducing process, we encountered a problem and couldn't reproduce this, and details are discussed in the Procedural / Data challenges.

Procedural / Data challenges

- MNIST

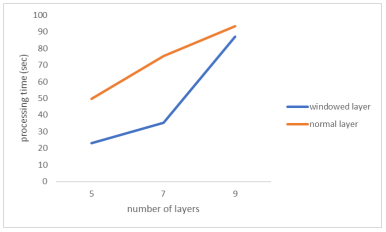
  As not all frequency components are well presented in the training set in the original MNIST dataset, subsampling was used in order to increase the relative magnitude of high frequency components. Although the Hamming window indeed provided performance improvements as the validation loss was much lower for the windowed model, the validation losses for both the models were rather very high, and the accuracy was not good. Hence, our guess was that the authors were just trying to use this method to prove their guess, which is that the lack of high frequency components for images in MNIST is why there's no performance increase for windowed model, because, by increasing the relative magnitude of high frequency components of the images indeed cause the windowed model to perform much better.

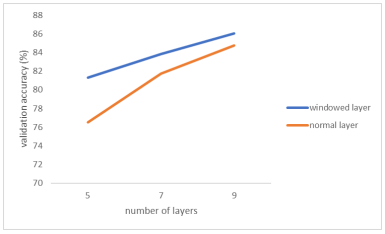
- ImageNet

  Although a test with ImageNet was scheduled, during the code modification section, we did not find a way to use the online imageNet resources and there is about 10 Gigabyte data in the training-validation-test package, which makes it extremely hard for us to exam and reconstruct the validation accuracy diagram in the paper. Our conclusion will be made only based on CIFAR-10, CIFAR-100, Fashion-MNIST as well as the MNIST dataset testing results. Which means that accuracy improvement of the windowed kernel on large size images (ImageNet images are in size 469x387) is not included.

Computational Cost

In the paper, the author suggests that the computation cost of large kernels with a hamming window is larger than small kernels. When verifying this concept, 3x3_stride 1 _pad 1 normal kernels and 7x7_stride 2_pad 3 kernels are selected respectively and the result shows that hamming windowed models take shorter computing time and also be more accurate in classifying images. Based on this result, we believe that for simple models, the computational cost can be reduced by correctly



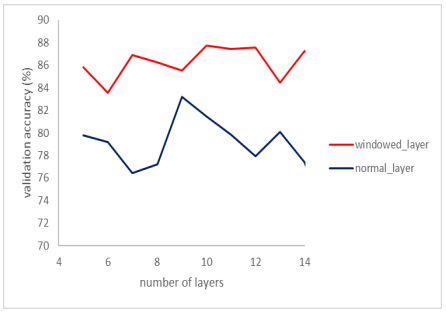


designing the kernel's shape, padding and stride, and a 7x7 windowed kernel can have a smaller cost than a 3x3 normal convolutional layer.

Figure 8. The graph above uses the CIFAR-10 dataset with 25 epochs.

Findings of the Study

- To find how the depth of convolution network influences the windowed kernel method, two tests have been scheduled , within each a normal CNN and a windowed CNN are used, one is normal (7x7 and 3x3 respectively) as a control group and the other set applies a hamming window on every convolution layer (hamming + 7x7 in both tests). With 50 epochs, and within 14 layers, the performance of the windowed layer CNN is always better than the normal CNN. This result is in line with the graph present in the original paper (Fig. 9.1) . But after adding more layers, the



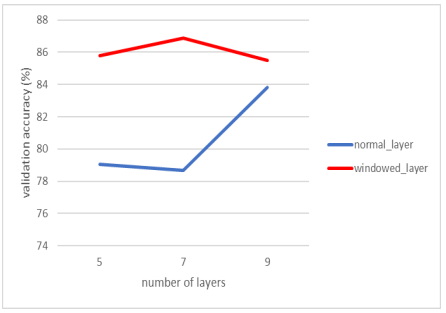
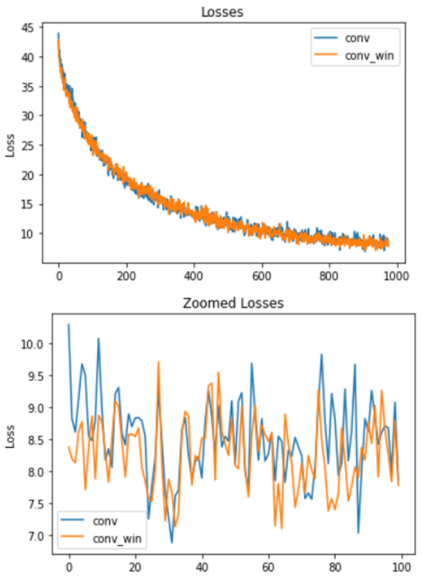


Figure 9.1. The above two graphs show the relationship between the validation accuracy of two trained CNN and number of layers in the model. The normal layers on the left graph use 7-by-7 kernels (same size), and the normal layers on the graph use 3-by-3 kernels while the size of the windowed kernel is unchanged (7x7).



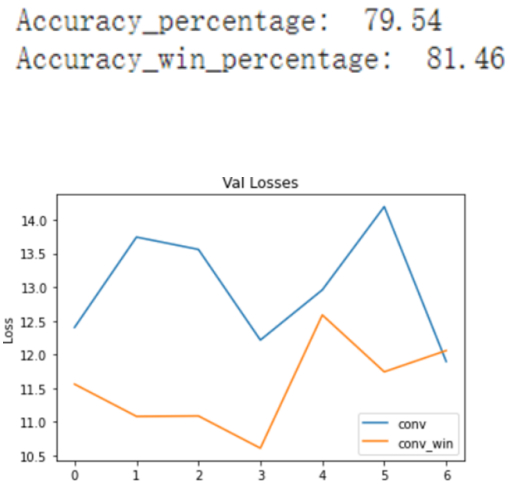


Figure 9.2. This shows the training loss, validation loss, along with the classification accuracy for both the normal model (blue line) and the windowed model (orange line) model when performing on CIFAR-10. Both two models are a composite of 20 7 by 7 kernels and were trained for 25 epochs.

advantage of the hamming window on improving accuracy is becoming increasingly smaller (Fig. 9.2). According to the test result, an assumption has been made that, for small size image dataset (CIFAR-10 and CIFAR-100 are both 32x32 pixels, MNIST is 28x28 pixels) the “large kernel + hamming” window method works efficiently and surely improve the accuracy of simple models but little effect when models are getting more complicated. Taking the computation cost test result into consideration, our research has reach a conclusion that for small size images (images in datasets similar to CIFAR-10, CIFAR-100, MNIST…), “large kernel+hamming” method should be used during model construction when the number of convolutional layers is less than 15-20 to improve the classification accuracy and an unnecessary and insignificant steps when designing CNN model with more layers.

- Hann and Blackman window instead of Hamming window

  Since the authors also mentioned using the Hann and Blackman window instead of the Hamming window, we have also tested that out. Due to the time constraint, we only tested the normal and windowed models with two windows on the MNIST dataset.

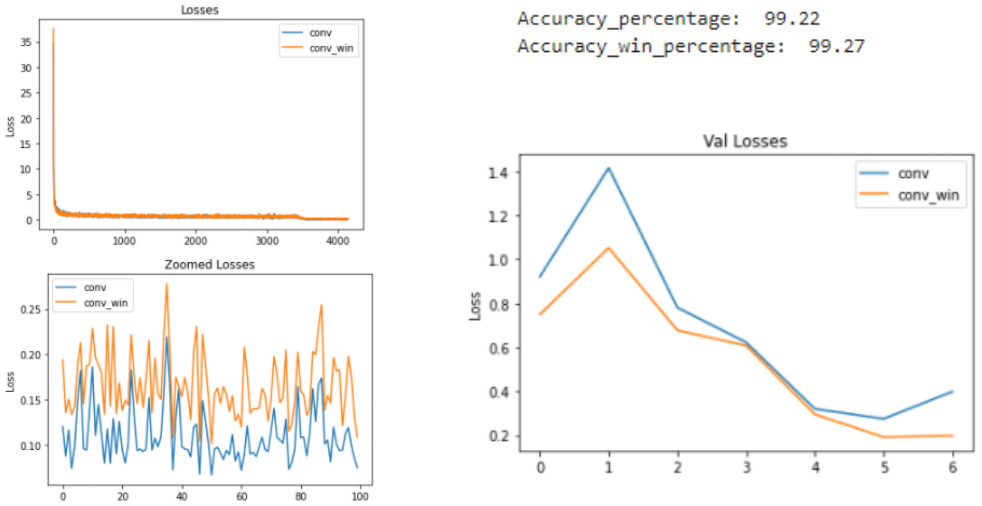


Figure 10. This shows the training loss, validation loss, along with the classification accuracy for both the normal model (blue line) and the windowed model (orange line) when performing on MNIST with the use of the Hanning window. Both the two models were trained for 90 epochs with kernel size of 7, and this result matches the claim in the paper that the result of using the Hanning window is no different with using the Hamming window.

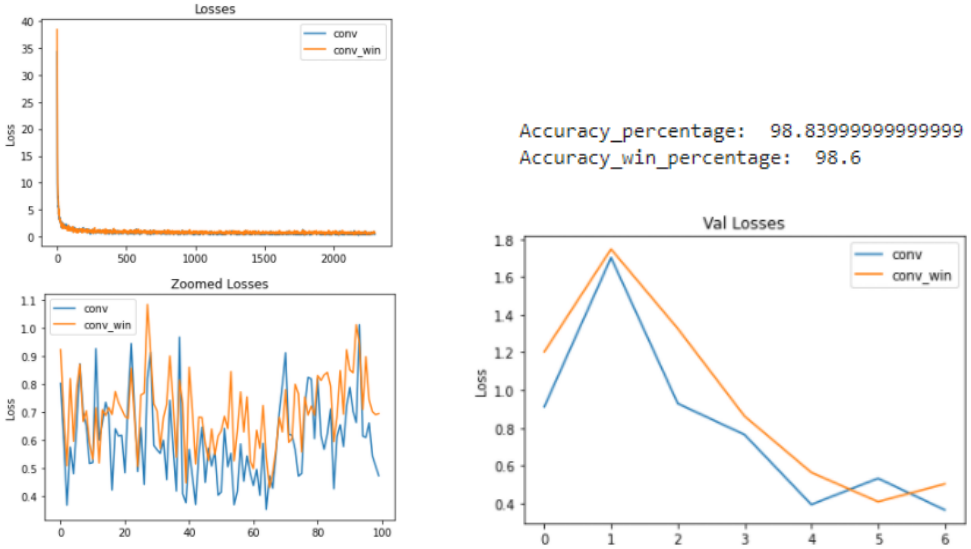


Figure 11. This shows the training loss, validation loss, along with the classification accuracy for both the normal

model (blue line) and the windowed model (orange line) when performing on MNIST with the use of the Blackman window. Both the two models were trained for 50 epochs with kernel size of 7, and this result matches the claim in the paper that the result of using the Hanning window is no different with using the Hamming window.

For the two models using the Hanning and Blackman window on the original MNIST dataset (without subsampling the size from 28 x 28 to 14 x 14), the result is shown in Figure 10 and Figure 11. Which has almost the same result with using the Hamming window, and this matches the claims the authors made in the paper.

Discussion

For the process of reproducing, the easy part was that the authors already provided the code to us, so we didn't have to implement everything from scratch. Also for the part of using the Hann and Blackman window instead of Hamming window, the code was easy to implement as numpy already had those window functions, and we just needed to call them.

The hard part was that although the code was provided, we were having trouble making it run at the beginning, and that cost us some time. We had to rerun the code whenever we made change to it so the new model is made, and the code is time-consuming to run, it could take at least several hours to complete, and due to the characteristic of colab, it would sometimes disconnect, and we had to re-run the whole thing all over again, this is really time-consuming and annoying. Also, when using the datasets, the original batch_size the authors used was 32, and that got changed by us to 64. If using the original batch_size, the code would take even longer to run.

Conclusions

- The convolutional layers in CNNs that employ small kernel sizes may be susceptible to performance degrading artefacts as for models using the Hamming window does perform better than the normal model.
- For the CNN models, the use of a standard Hamming window on larger kernels indeed lowered the validation loss and enhanced the classification accuracy on the benchmark datasets.
- The use of the Hanning and Blackman window does make the model have the same performance as using the Hamming window.
- After adding more layers, the advantage of the hamming window on improving accuracy is becoming increasingly smaller, which does match the warnings in the code.

Future Work and Lessons Learnt

With this paper, the authors also claimed that the robustness of the windowed model would be improved against DeepFool [9] (white-box) and spatial transformation [10] (black-box) attacks. However, since it's not the central claim, we didn't put our focus on this part, and thus couldn't manage to reproduce and prove this claim. Hence, some future work may be trying to reproduce this and see if the windowed model indeed has higher robustness against DeepFool and spatial transformation attacks.

By trying to reproduce the test and claims, we learned many things related to image processing. Although the use of small kernel sizes could reduce computational complexity and most of the times increase the classification accuracy, it could make the CNNs susceptible to spectral leakage, and that could induce performance-degrading artifacts, and the use of Hamming, Hanning, or Blackman window could be a solution to use. Also, with the use of these window functions, larger kernel sizes could be used and there will be less truncations.

References

[1] N. Tomen, Jan C. van Gemert. Spectral Leakage and Rethinking the Kernel Size in CNNs. Computer Vision Lab, Delft University of Technology. arXiv: 2101.10143v2, 2021.

[2] Richard W. Hamming. Digital Filters. Dover Civil and Mechanical Engineering. Dover Publications, 1998.

[3] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[4] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747, 2017.

[5] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Department of Computer Science, 04 2009.

[6] Richard W. Hamming. Digital Filters. Dover Civil and Mechanical Engineering. Dover Publications, 1998.

[7] A.V. Oppenheim, R.W. Schafer, J.R. Buck, and L. Lee. Discrete-time Signal Processing. Prentice Hall international editions. Prentice Hall, 1999.

[8] KM Muraleedhara Prabhu. Window functions and their applications in signal processing. Taylor & Francis, 2014.

[9] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, CVPR, pages 2574–2582, 2016.

[10] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In International Conference on Machine Learning, ICML, pages 1802–1811, 2019.

[11] M. Lee, T. Kim, Y. Ban, E. Song, S. Lee. Sampling Operator to Learn the Scalable Correlation Filter for Visual Tracking. IEEE Access, 2019.

Appendix for Work Breakdown

For all three of us, although the author provided the original code, but it was only on Cifar-10. And we had to tweak and change or create the method for the code in order for it to successfully run on colab for different scenarios.

- Wenhan Liu
  - Progress Report
    - Completed the paper summary of the progress report
    -
  - Final Report
    - Completed the Methodology part.
    - Completed the Results part.
    - Completed the second dot of Findings of the Study part (About using the Hanning and Blackman window for the CNN model instead of the Hamming window).
    - Completed the first dot of Procedural / Data challenge, about MNIST.
    - Completed the Discussion part.
    - Completed the Conclusions part.
    - Completed the Future work and Lessons Learnt part.
  - Code
    - Tested the model on MNIST, Fashion-MNIST
    - Subsampled the images in MNIST and tested again
    - Tested the model using Hanning and Blackman window instead of Hamming window on MNIST dataset.

- Shuo Zhang
  - Progress Report
    - Completed the backgrounds of the progress report

  - Final Report
    - Completed the Computational Cost part.
    - Completed the first dot of Findings of the Study part (the influence of the number of layers to the model).
    - Completed the Background and Related Work part.
    - Completed the second dot of Procedural / Data challenge, about ImageNet.
  - Code
    - Tested the model on CIFAR-100
    - Tested how the model would perform with different number of layers
- Zhiqi Bei
  - Final Report
    - Completed the Introduction and Scope of Reproducibility part
  - Code
    - Tested the model on CIFAR-10