Spectral Leakage and Rethinking the Kernel Size in CNNs

By Wenhan Liu
Zhiqi Bei
Shuo Zhang

Spectral Leakage

Appears when the number of periods of signal passed to DFT is not an integer.

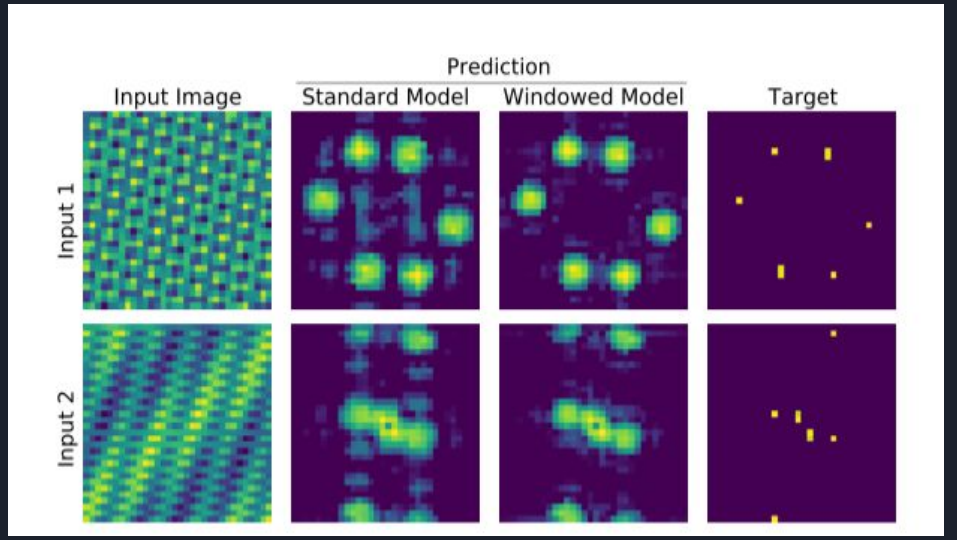
When applying a window function to the periodic signal, there is a chance that the cut window is not periodic and the sharp edges (sinc like function) will generate a bunch of frequencies that distorts the filtered image around the signal.

CNNs

Small Kernel Size

- Reduce computational complexity
- Improve accuracy

- Cause CNNs to be susceptible to spectral leakage
- Induce performance-degrading artifacts
- Typically lead to severe truncation



This picture shows the Learning to predict the FFT magnitude of an input image with a single convolutional layer. Network predictions for two example synthetic input images, randomly generated as the sum of three 2-D sine waves. The target vectors are the FFT magnitudes of the input images, including negative frequencies. And the model using Hamming windows alleviates leakage artifacts.

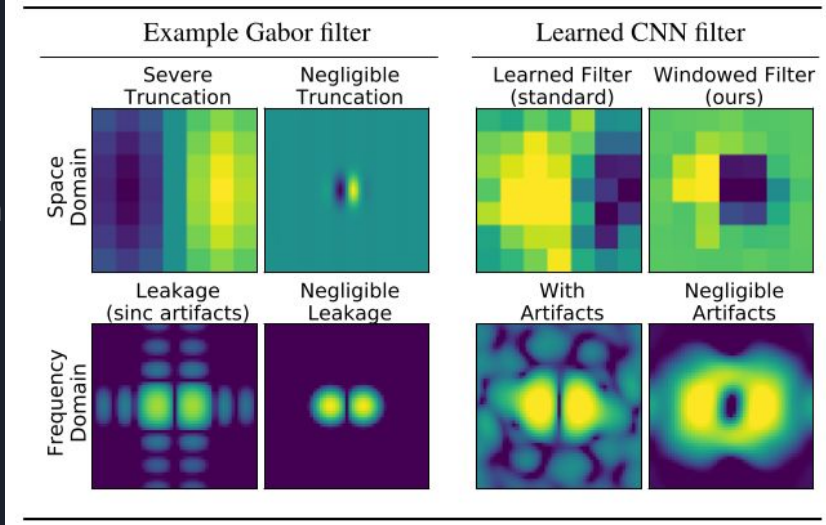
Central Claim

The problem of spectral leakage caused by windowing artifacts in filtering operations in the context of CNNs was considered.

By reducing the spectral Leakage, the classification accuracy of the CNN would be increased and would have a better result compared with normal CNNs.

Central Claim

By reducing the spectral Leakage, the classification accuracy of the CNN would be increased and would have a better result compared with normal CNNs.



Left part is an Example Gabor (bandpass) filter with severe truncation which has kernel size of 7 × 7 leads to spectral leakage in its frequency response due to sinc artifacts. The same filter with negligible truncation which has kernel size of 49×49 is a good quality bandpass filter.
Right part is A standard 7x7 CNN kernel trained on CIFAR-10 struggles to learn good quality bandpass filters, as the use of small kernel sizes typically lead to severe truncation.

Solution

The use of larger kernel sizes along with the Hamming window function to alleviate leakage in CNN architectures.

Hamming Window

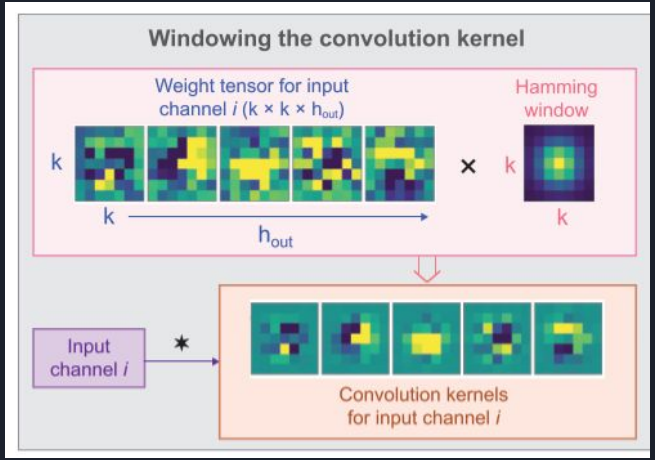
01 The Hamming window is a taper formed by using a raised cosine with non-zero endpoints, optimized to minimize the nearest side lobe.

02 $$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{M-1}\right) \qquad 0 \le n \le M-1$$

03 Most references to the Hamming window come from the signal processing literature, where it is used as one of many windowing functions for smoothing values.

Detailed Process

Figure 1



This shows the tapering of the convolution kernels with the Hamming window. The typical weight tensor in a 2-D convolutional layer has size (k × k × hin × hout). But with this figure, it only shows a single input channel i, which is convolved with h out distinct k × k kernels, which are generated by multiplying each k ×k slice of the weight tensor with the k × k Hamming window.

This architecture is used for CIFAR-10, CIFAR-100, Fashion-MNIST and MNIST experiments. The depth (number of convolutional layers M) of the network was varied by repeating a convolution block, which is the part within the blue box. The first layer downsamples the input via a strided convolution with a 7 × 7 kernel, which is similar to ResNet architectures, while the kernel size is kb for all other convolutional layers.

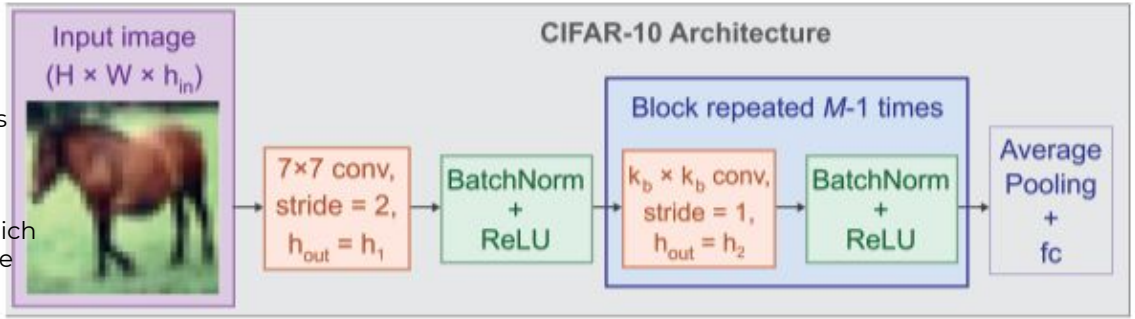
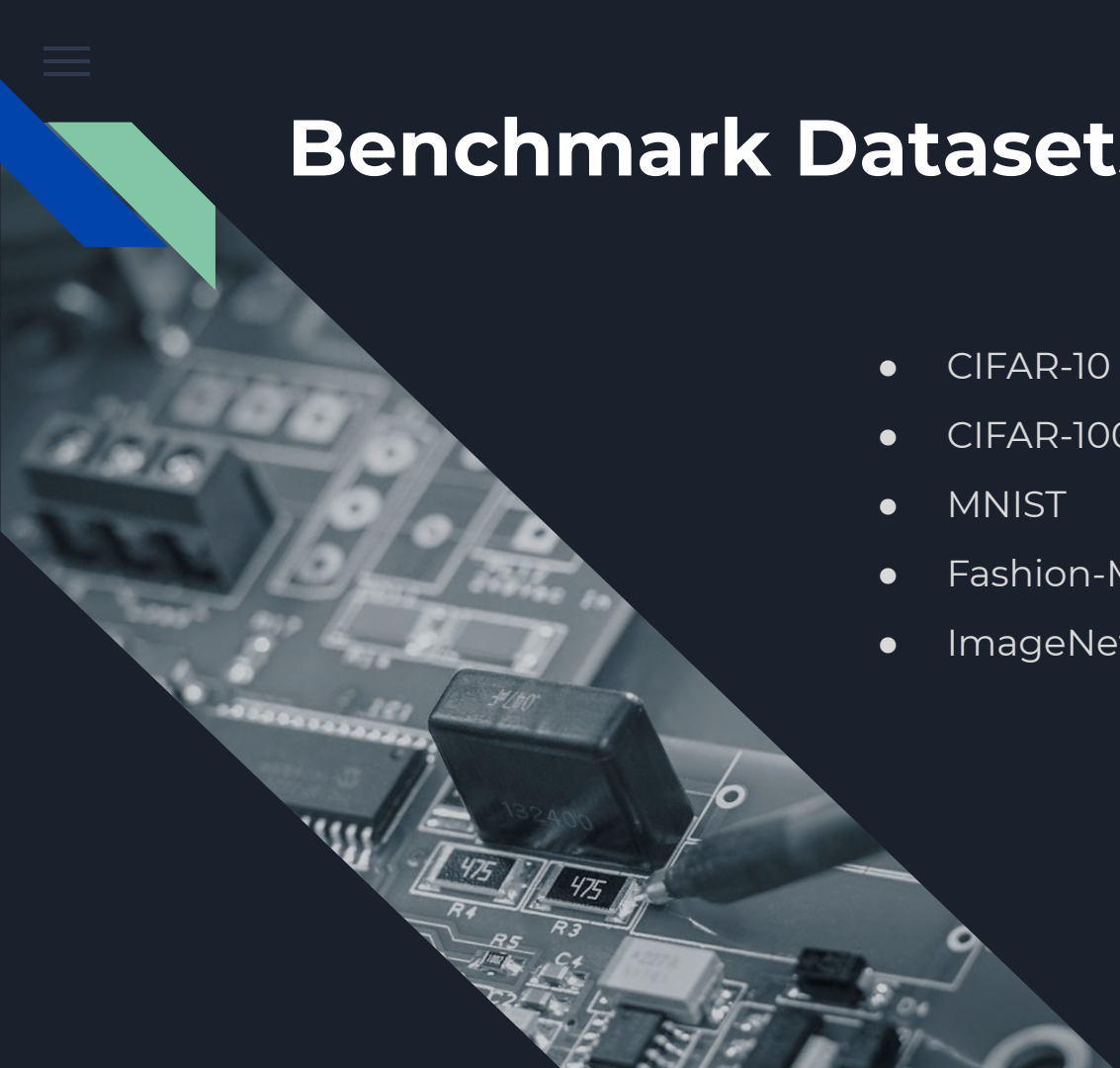


Figure 2

Benchmark Datasets



- CIFAR-10
- CIFAR-100
- MNIST
- Fashion-MNIST
- ImageNet

CIFAR-10 & CIFAR-100

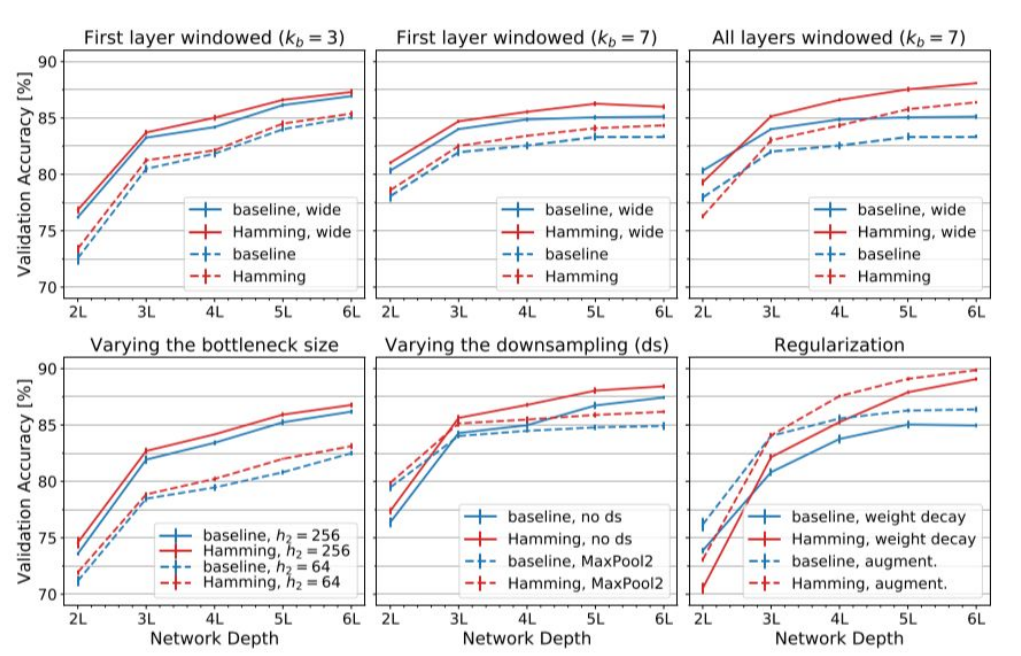
CIFAR-10

- Dataset of color images with size 32x32
- Consists of 60,000 images in 10 classes

CIFAR-100

- Just like CIFAR-10, except it has 100 classes containing 600 images each

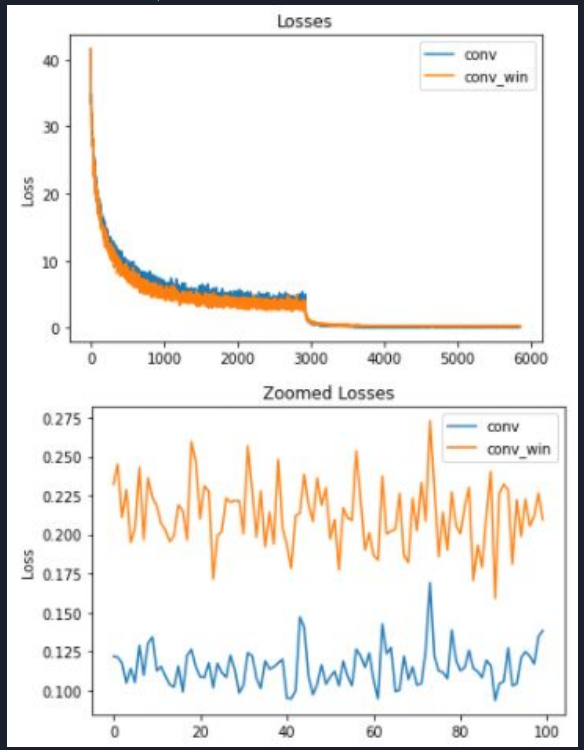
CIFAR-10 & CIFAR-100



A comparison was done between a normal CNN and the windowed CNN, while using kernel size k = 3 for normal CNN, and kernel size = 7x7 and 9x9 for windowed CNN, the windowed CNN (doesn't matter k = 7x7 or k = 9x9) got much better accuracy result compared with the normal CNN with kernel size k = 3.

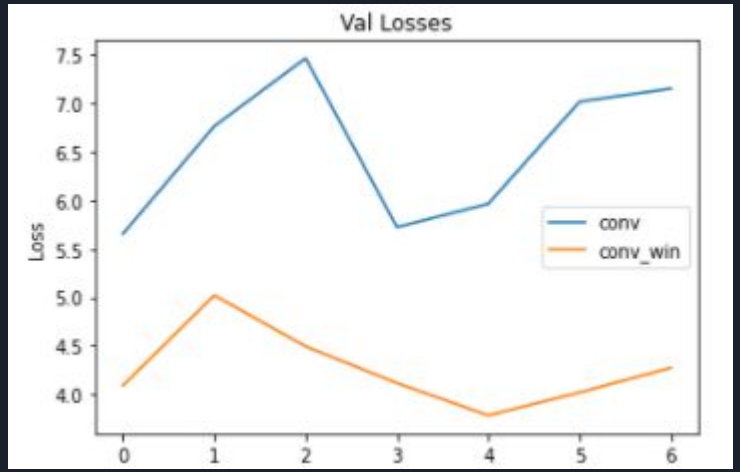
This also shows the CIFAR-10 validation accuracy as a function of varying the network depth (M = 2 . . . 6 convolutional layers), in models using the Hamming window(red lines)and baselines with standard convolutional layers(blue lines). Line plots depict the average of 5 runs with error bars denoting standard deviation. As a result, for all architecture variants: Hamming window only on the first layer; Hamming window on all layers; different channel width; different methods of downsampling; and regularization. The models using the Hamming window consistently outperform the baseline models in networks deeper than 2 layers.

CIFAR-10



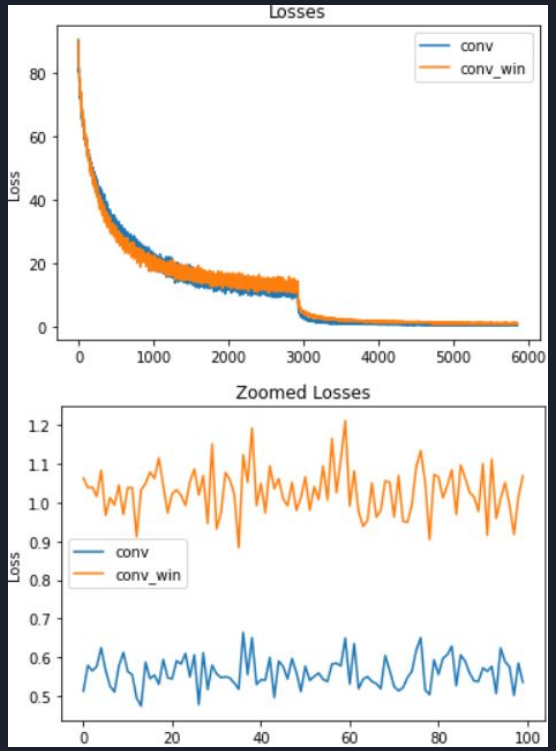
For CIFAR-10, we tested both the normal CNN and windowe CNN. For training loss, the performance of windowed CNN and normal CNN are similar, with windowed CNN having slightly larger loss.

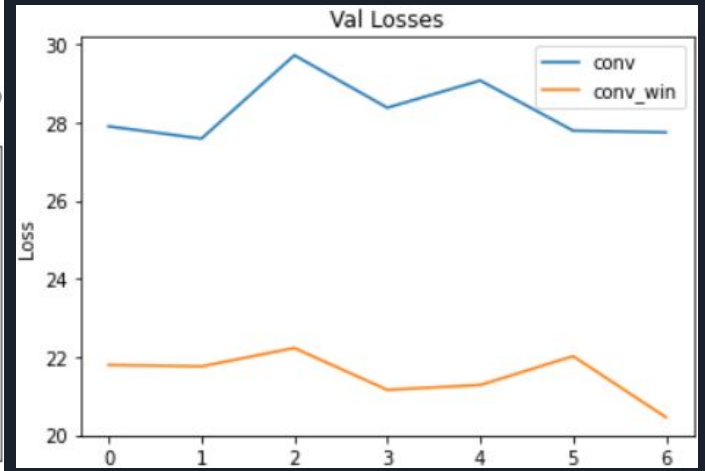
But the validation loss for windowed CNN is much lower than normal CNN. And the accuracy for windowed CNN is also higher.




```
Accuracy_percentage:  90.45
Accuracy_win_percentage:  93.05
```

CIFAR-100



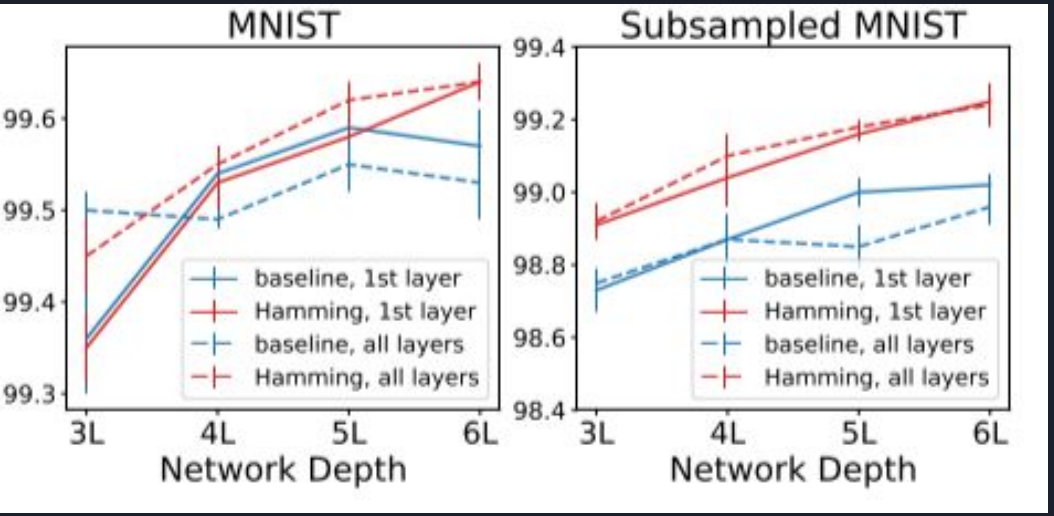


For CIFAR-100, we tested both the normal 7 x 7 CNN and windowed CNN. Same as cifar 10 dataset result, the training loss of windowed CNN is slightly higher than the normal CNN.

The validation loss of windowed CNN is much lower than normally trained CNN network and have a higher accuracy. The test result is consistent with the statement in the article.

```
Accuracy_percentage:  65.44
Accuracy_win_percentage:  71.77
```

MNIST



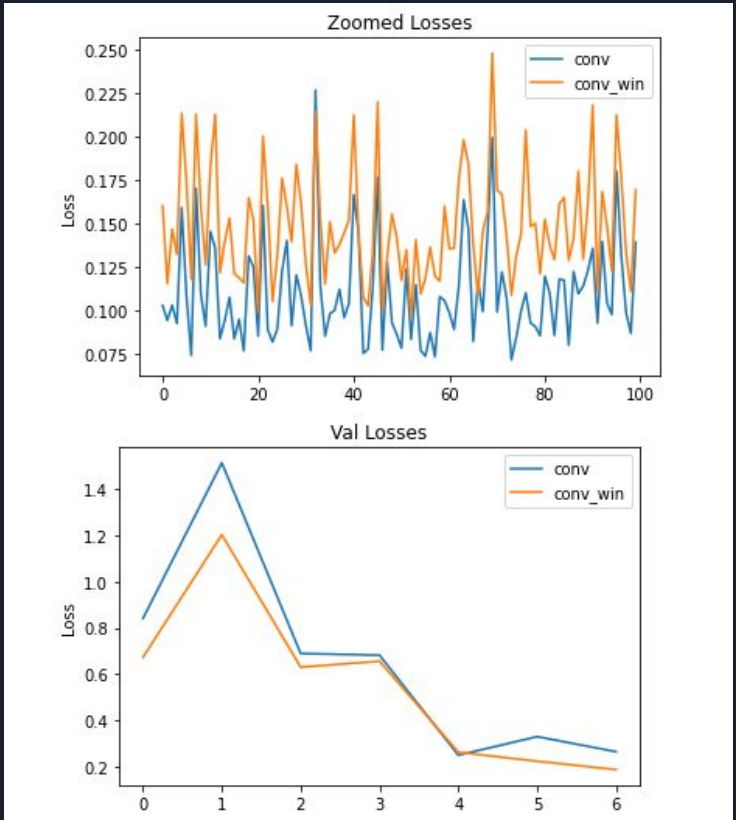
- Dataset of handwritten digits
- Grayscale images with size 28x28

- No performance improvement in the beginning, because the lack of high frequency components for images in MNIST dataset as leakage in lowpass and bandpass filters cannot contaminate high frequency information.

- After the subsampling of each image in MNIST from the size of 28x28 to 14x14, the relative magnitude of the high frequency component was increased. And in this way, the windowed networks again achieved much better classification results than normal CNNs.

- The benefits of windowing are more pronounced when the magnitude of high frequency components is increased by subsampling the input images.

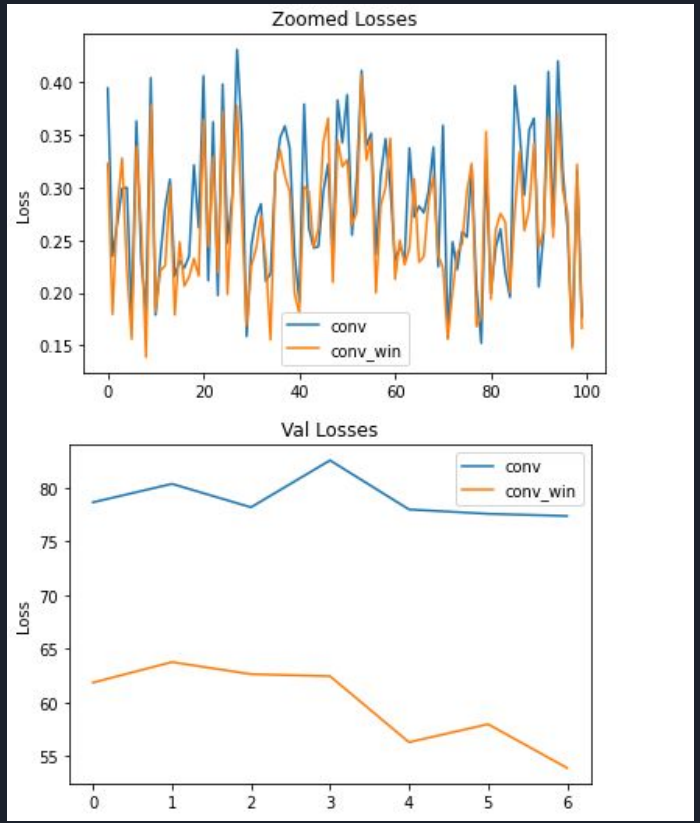
MNIST



This is the original dataset, we tested with normal CNN and a windowed CNN, although most of the times the training loss were little higher for conv_win, but the validation loss is lower for conv_win.

And just like the paper states, there isn't much performance improvement for the windowed network.

MNIST

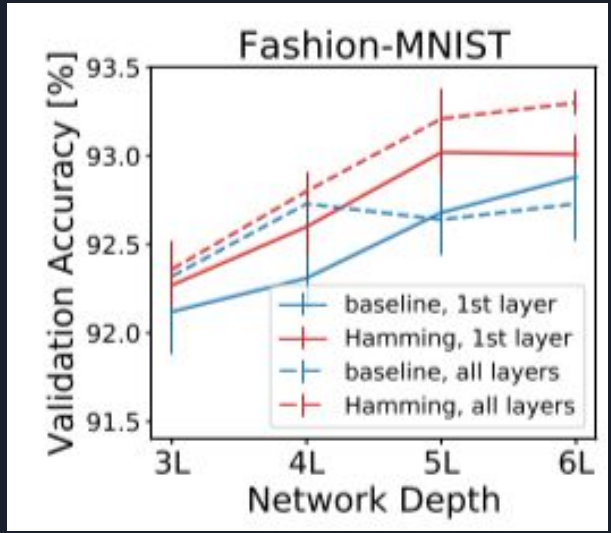


Then, after subsampling each image in MNIST from 28x28 to 14x14, we tested again with normal CNN and a windowed CNN, and this time, the training loss for conv_win is lower most of the time, although it's only a little bit. But the validation loss is much lower for conv_win than normal CNN. Which is exactly the same as the paper states.

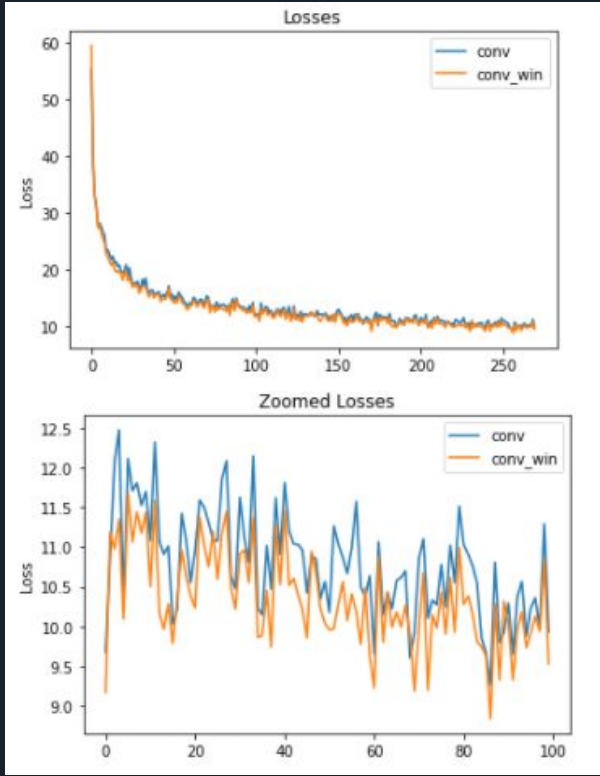
Fashion-MNIST

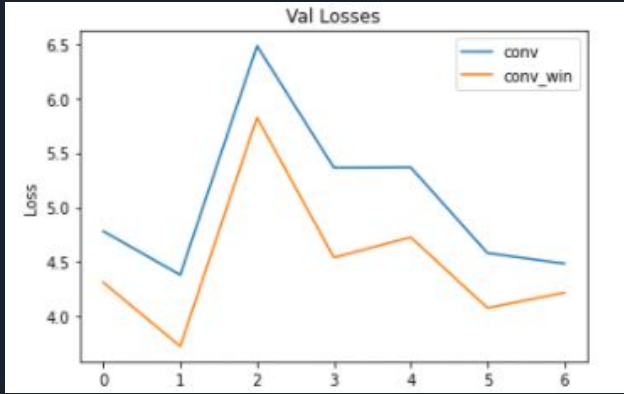
- Dataset of Zalando's article images
- Grayscale images with size 28x28, associated with a label from 10 classes

- The windowed network gained better performance than normal CNNs right from the start with the graph below showing the validation accuracy as a function of varying network depth.



Fashion-MNIST





Comparing with the experiment on the other datasets, for Fashion-MNIST, the training process was rather slow, but the performance improvement of windowed CNN is more obvious right from the start.

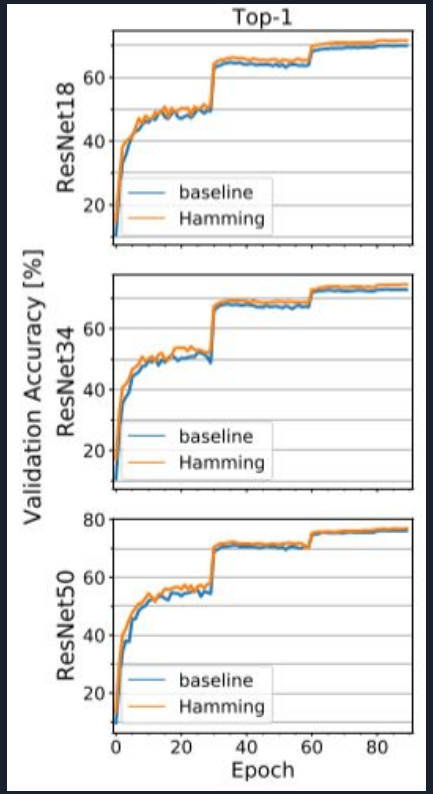
The training loss for conv_win is noticeably lower, and the validation loss is also much lower for windowed CNN. With the accuracy for windowed CNN being higher.

```
Accuracy_percentage:  91.21000000000001
Accuracy_win_percentage:  92.32000000000001
```

ImageNet

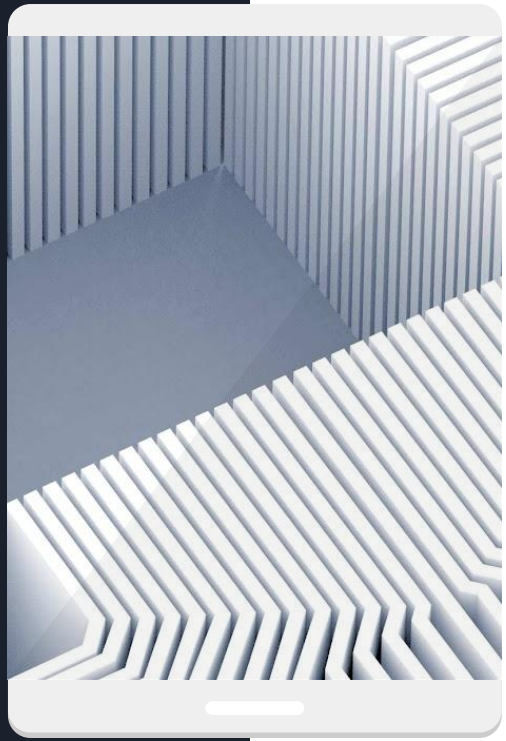
- The ImageNet dataset contains 14,197,122 annotated images according to the WordNet hierarchy.

- Similar result as the performance on other datasets, the windowed network still got better performance.



This graph shows that the ImageNet validation accuracy is higher for ResNet architectures with the Hamming window than baseline ResNet models throughout training.

Experiment

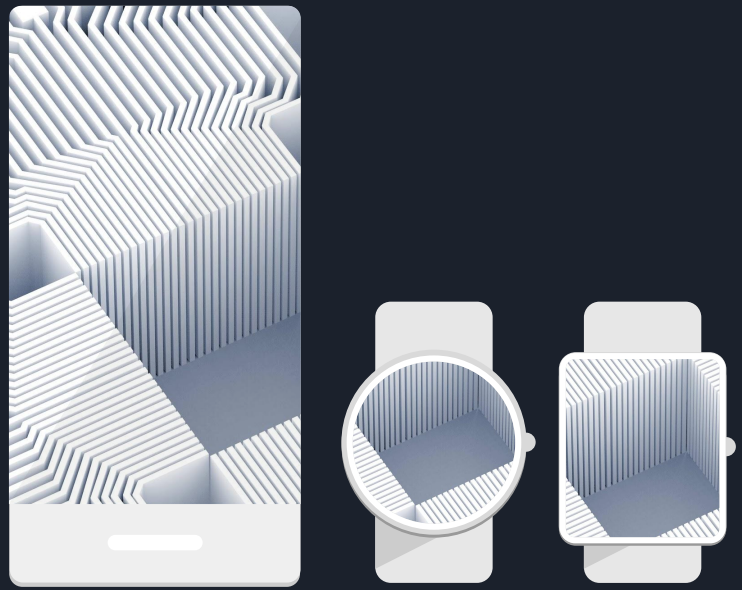


- A network with no downsampling layers was trained, and the strided convolution was replaced with a standard convolution (stride=1) while windowing only the first layer. As another control experiment, another network was trained, it performs downsampling via a max-pooling layer with a 2×2 window instead of a strided convolution. As a result, in both cases using a Hamming window still improves CIFAR-10 validation accuracy, which indicates that the performance increase provided by windowed convolutions is independent of aliasing and the choice of downsampling method.
- Authors also tried Hann and Blackman window instead of the Hamming window, and found that they provide the same performance boost to CNN as the Hamming window does.

Robustness against Adversarial Attack

- Tested on CIFAR-10 against DeepFool, which is a simple and accurate method to fool deep neural networks.
- Although for network with kernel size k = 5, the windowed network didn't perform as good as baseline CNN, but for large kernel sizes, the robustness of windowed network model is significantly better than baseline CNNs.

Spotlight on wearables

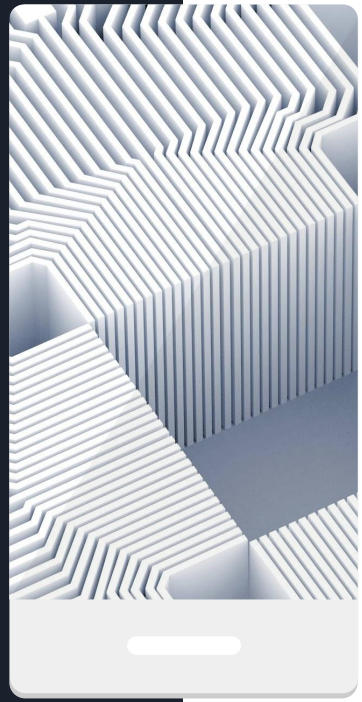


Computational Cost

- Complexity of 2D convolution on a H×W image O(*HW k^2*) scales quadratically with kernel size k (or linearly for separable convolutions O(2*HWk*).

- The use of larger kernels, which are computationally more expensive, but parallelizable compared to deeper networks, is a viable option when the kernels are windowed properly.

Result / Conclusion



- The convolutional layers in CNNs that employ small kernel sizes may be susceptible to performance degrading artifact.
- The use of a standard Hamming window on larger kernels enhanced the classification accuracy on the benchmark datasets.
- The Robustness of the windowed CNN was improved against DeepFool and spatial transformation attacks in windowed CNNs.

Thank you!