Table 1. Top-1 accuracy of WaveCNets on ImageNet validation set.

Wavelet		WVGG16bn	WResNet18	WResNet34	WResNet50	WResNet101	WDenseNet121
None (baseline)*		73.37	69.76	73.30	76.15	77.37	74.65
Haar		74.10 (+0.73)	71.47 (+1.71)	74.35 (+1.05)	**76.89 (+0.74)**	78.23 (+0.86)	75.27 (+0.62)
Cohen	ch2.2	74.31 (+0.94)	**71.62 (+1.86)**	74.33 (+1.03)	76.41 (+0.26)	78.34 (+0.97)	75.36 (+0.71)
	ch3.3	**74.40 (+1.03)**	71.55 (+1.79)	74.51 (+1.21)	76.71 (+0.56)	**78.51 (+1.14)**	**75.44 (+0.79)**
	ch4.4	74.02 (+0.65)	71.52 (+1.76)	**74.61 (+1.31)**	76.56 (+0.41)	78.47 (+1.10)	75.29 (+0.64)
	ch5.5	73.67 (+0.30)	71.26 (+1.50)	74.34 (+1.04)	76.51 (+0.36)	78.39 (+1.02)	75.01 (+0.36)
Daubechies	db2	74.08 (+0.71)	71.48 (+1.72)	74.30 (+1.00)	76.27 (+0.12)	78.29 (+0.92)	75.08 (+0.43)
	db3		71.08 (+1.32)	74.11 (+0.81)	76.38 (+0.23)		
	db4		70.35 (+0.59)	73.53 (+0.23)	75.65 (−0.50)		
	db5		69.54 (−0.22)	73.41 (+0.11)	74.90 (−1.25)		
	db6		68.74 (−1.02)	72.68 (−0.62)	73.95 (−2.20)		

* corresponding to the results of original CNNs, i.e., VGG16bn, ResNets, DenseNet121.

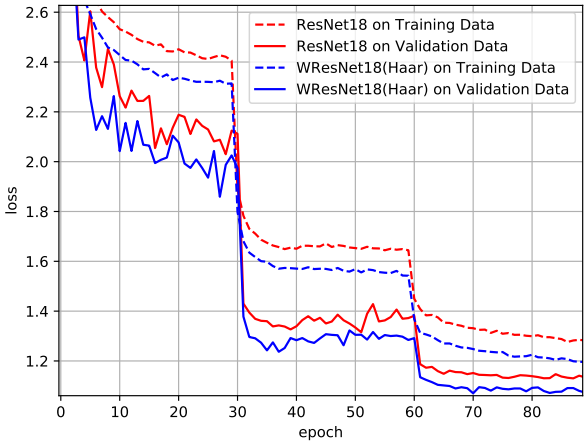

Figure 4. The loss of ResNet18 and WResNet18(Haar).

curacy, although the best wavelet varies with CNN. For example, Haar and Cohen wavelets improve the accuracy of ResNet18 by 1.50% to 1.86%. However, the performance of asymmetric Daubechies wavelet gets worse as the approximation order increases. Daubechies wavelets with shorter filters ("db2" and "db3") could improve the CNN accuracy, while that with longer filters ("db5" and "db6") may reduce the accuracy. For example, the top-1 accuracy of WResNet18 decreases from 71.48% to 68.74%. We conclude that the symmetric wavelets perform better than asymmetric ones in image classification. That is the reason why we do not train WVGG16bn, WResNet101, WDenseNet121 with "db3", "db4", "db5", "db6".

We retrain ResNet18 using the standard ImageNet classification training repository in PyTorch. In Fig. 4, we compare the losses of ResNet18 and WResNet18(Haar) during the training procedure. Fig. 4 adopts red dashed and green dashed lines to denote the train losses of ResNet18 and WResNet18(Haar), respectively. Throughout the whole training procedure, the training loss of WResNet18(Haar) is about 0.08 lower than that of ResNet18, when the two networks employ the same amount of learnable parameters. This suggests that wavelet accelerates the training of ResNet18 architecture. On the validation set, WResNet18 loss (green solid line) is also always lower than ResNet18 loss (red solid line), which lead to the increase of final classification accuracy by 1.71%.

Fig. 5 presents four example feature maps of well trained CNNs and WaveCNets. In each subfigure, the top row shows the input image with size of 224×224 from ImageNet validation set and the two feature maps produced by original CNN, while the bottom row shows the related information (image, CNN and WaveCNet names) and feature maps produced by the WaveCNet. The two feature maps are captured from the 16th output channel of the final layer in the network blocks with tensor size of 56×56 (middle) and 28×28 (right), respectively. The feature maps have been enlarged for better illustration.

From Fig. 5, one can find that the backgrounds of the feature map produced by WaveCNets are cleaner than that produced by CNNs, and the object structures in the former are more complete than that in the latter. For example, in the top row of Fig. 5(d), the clock boundary in the ResNet50 feature map with size of 56×56 are fuzzy, and the basic structures of clocks have been totally broken by strong noise in the feature map with size of 28×28. In the second row, the backgrounds of feature maps produced by WResNet50(ch3.3) are very clean, and it is easy to figure out the clock structures in the feature map with size of 56×56 and the clock areas in the feature map with size of 28×28. The above observations illustrate that the down-sampling operations could cause noise accumulation and break the basic object structures during CNN inference, while DWT in WaveCNets relieves these drawbacks. We believe that this is the reason why WaveCNets converge faster in training and ultimately achieve better classification accuracy.

In [42], the author is surprised at the increased classification accuracy of CNNs after filtering is integrated into the down-sampling. In [12], the authors show that "ImageNet-trained CNNs are strongly biased towards recognising textures rather than shapes". Our experimental results suggest that this may be sourced from the commonly used down-sampling operations, which tend to break the object structures and accumulate noise in the feature maps.