Harmonic Networks for Image Classification

Matej Ulicny
ulinm@tcd.ie
Vladimir A. Krylov
vladimir.krylov@tcd.ie
Rozenn Dahyot
rozenn.dahyot@tcd.ie

School of Computer Science and Statistics
Trinity College Dublin
Dublin, Ireland

Abstract

Convolutional neural networks (CNNs) learn filters in order to capture local correlation patterns in feature space. In contrast, in this paper we propose harmonic blocks that produce features by learning optimal combinations of responses to preset spectral filters. We rely on the use of the Discrete Cosine Transform filters which have excellent energy compaction properties and are widely used for image compression. The proposed harmonic blocks are intended to replace conventional convolutional layers to produce partially or fully harmonic versions of new or existing CNN architectures. We demonstrate how the harmonic networks can be efficiently compressed by exploiting redundancy in spectral domain and truncating high-frequency information. We extensively validate our approach and show that the introduction of harmonic blocks into state-of-the-art CNN models results in improved classification performance on CIFAR and ImageNet datasets.

1 Introduction

CNNs have been designed to take advantage of implicit characteristics of natural images, specifically correlation in local neighborhood and feature equivariance. The wide application of features obtained by convolving images with explicitly defined local filters highlights the shift from the extraction of global information towards local learning.

Standard CNNs rely on the learned convolutional filters that allow them to be adjusted flexibly to the problem and available data. In some cases, however, it may be advantageous to revert to preset filter banks, e.g., in the case of limited training data, when the use of appropriate filter collections can help to avoid overfitting, or in order to reduce the computational complexity of the system. Previously, several collections of preset filters have been proposed to replace learned convolutions or perform preprocessing in the task of image classification. The scattering network have been proposed in [4] to use multiple layers of wavelet filters to model geometrical visual information. It has been shown that the scattering network built on complex-valued Morlet wavelets could achieve state of the art results in handwritten digit recognition and texture classification. The scattering network with its filters designed to extract translation and rotation invariant representations was shown to achieve comparable classification accuracy to unsupervised deep learning [21]. Other types of filters proposed to introduce preset filters in CNNs include oriented Gabor filters [20], Gaussian derivatives [15] and circular harmonics to enforce rotation equivariance [35].

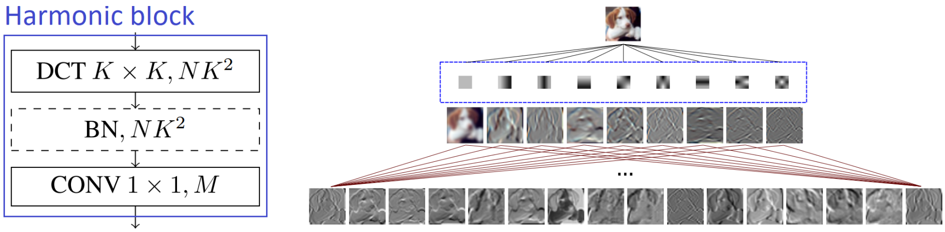

Figure 1: Left: Design of the harmonic block. Boxes show operation type, size of filter (if applicable) and the number of output channels given the block filter size K, number of input channels N and output channels M. Batch normalization (BN) block is optional. Right: Visualization of the harmonic block applied to an input layer.

In this paper we propose to replace the standard convolutional operations in CNNs by harmonic blocks that learn the weighted sums of responses to the Discrete Cosine Transform (DCT) filters, see Fig. 1. The latter have been successfully used in JPEG encoding to transform image blocks into spectral representations to capture the most information with a small number of coefficients. Motivated by frequency separation and energy compaction properties of DCT, the proposed harmonic networks rely on combining responses of window-based DCT with a small receptive field. The key distinction from scattering networks is that these create a new path for each wavelet filter used at every layer, which consequently increases the number of paths exponentially with the increase of the network's depth. Our method learns how to optimally combine spectral coefficients at every layer to produce a fixed size representation defined as a weighted sum of responses to DCT filters. The use of DCT filters allows one to easily address the task of model compression. We extensively validate harmonic networks performance on CIFAR-10/100 and ImageNet-1k classification datasets.

The paper is organized as follows. We first review the related work in Sec. 2, and briefly recall the basics of DCT in Sec. 3. We then introduce the harmonic networks in Sec. 4, assess experimentally their classification performance in Sec. 5, and conclude the paper in Sec. 6.

2 Related work

DCT & CNNs Several works consider combining spectral information with CNNs. Networks trained on DCT coefficients are frequently used in forensics, specially for detection of multiply compressed images. A common practice in several works [2, 3, 31] is to classify histograms of preselected DCT coefficients by 1D convolutional network. In another work [18] a multi-branch 2D CNN is trained on feature maps spanned by the first 20 AC coefficients (corresponding to non-zero frequencies in DCT) extracted from JPEG images.

A number of studies have investigated the use of spectral image representations for object recognition. DCT on small resolution images coupled with coefficient truncation was used to speed up training of fully connected sparse autoencoders [37]. DCT features from the entire image were used to train Radial Basis Function Network for face recognition [7]. A significant convergence speedup and case-specific accuracy improvement have been achieved by applying DCT transform to early stage learned feature maps in shallow CNNs [9] whereas the later stage convolutional filters were operating on a sparse spectral feature representation. In [12, 29] it was demonstrated how precomputed or JPEG-extracted DCT coefficients can be efficiently used to train classification CNNs.

Wavelets & CNNs The scattering network based on rotation and scale invariant wavelet transform was shown to effectively reduce the input representation while preserving discriminative information for training CNN on image classification [22, 27] and object detection task [23] achieving performance comparable to deeper models. *Williams et al.* [33] have advocated image preprocessing with wavelet transform, but used different CNN for each frequency subband. Wavelet filters were also used as a preprocessing method prior to NN-based classifier [26], and to enhance edge information in images prior to classification [6].

Other works have used wavelets in CNN computational graphs. Second order coefficients from Fast Wavelet Transform were used in [34] to design wavelet pooling operator. Similar approach was taken by *Ripperl et al.* who designed spectral pooling [25] based on Fast Fourier Transform of the features and high-frequency coefficient truncation. They also proposed to parametrise filters in Fourier domain to decrease their redundancy and speed up the convergence. In both works, the pooled features were recovered with Inverse Fast Wavelet or Discrete Fourier Transform respectively, thus the CNN still operates in spatial domain. To address texture classification, *Fujieda et al.* [8] proposed a Wavelet Convolutional Network that is trained on responses to Haar wavelets and concatenates higher order coefficient maps along with features of the same dimensionality learned from lower-order coefficients. Similar approach is taken by *Lu et al.* [19] that learns from both spatial and spectral information that is decomposed from first layer features. The higher-order coefficients are also concatenated along with the lower dimensional feature maps. However, contrary to our method, Wavelet CNNs decompose only the input features and not features learned at intermediate stages. Moreover, the maximum number of decompositions performed was limited to the number of spatial resolutions of CNN features. Robustness to object rotations was addressed by modulating learned filters by oriented Gabor filters [20]. Furthermore, *Worrall et al.* incorporated complex circular harmonics into CNNs to learn rotation equivariant representations [35]. Similarly to our harmonic block, the structured receptive field block [15] learns new filters by combining fixed filters, i.e. a considerably larger set of Gaussian derivatives. DCFNet [24] expresses filters by truncated expansion of Fourier-Bessel basis, maintaining accuracy of the original model while reducing the number of parameters.

Compressing DNNs Numerous works have focused on compressing the size of neural networks and decreasing the inference and training time. Speedup and memory saving for inference can be achieved by approximating the trained full-rank CNN filters by separable rank-1 filters [16]. Assuming smoothness of learned filters, Frequency-Sensitive Hashed Network (FreshNet) [5] expresses filters by their DCT representation and groups their parameters to share the same value within each group. *Wang et al.* [32] relaxes this constrain to express each weight by its residual from the cluster center. Weights in this form were quantized and transformed via Huffman coding for storage purposes. Convolution was performed in the frequency domain to reduce the computational complexity. *Han et al.* [13] compressed networks by pruning, clustering and quantizing weights which are consequently fine-tuned. It has been shown [17] that a model complexity can be adjusted during the training time: increased via introduction of new filters by rotating and applying noise to existing ones, and reduced by clustering to selectively decrease their redundancy.

3 Discrete Cosine Transform

DCT is an orthogonal transformation method that decomposes an image to its spatial frequency spectrum. In continuous form, a 2D signal is projected to a sum of sinusoids with different frequencies. The contribution of each sinusoid towards the whole signal is determined by its coefficient calculated during the transformation. DCT is also a separable transform and due to its energy compaction properties on natural images [10] it is commonly used for image and video compression in widely used JPEG and MPEG formats. Karhunen-Loève transform is considered to be optimal in signal decorrelation, however it transforms signal via unique basis functions that are not separable and need to be estimated for every image.

The literature provides several distinct definitions of DCT. We will rely on the most common formulation, DCT-II, which is computed on a 2-dimensional grid X of size $A \times B$ representing the image patch with 1 pixel discretisation step as

$$Y_{u,v} = \sum_{x=0}^{A-1} \sum_{y=0}^{B-1} \sqrt{\frac{\alpha_u}{A}} \sqrt{\frac{\alpha_v}{B}} X_{x,y} \times \cos\left[\frac{\pi}{A}\left(x+\frac{1}{2}\right)u\right] \cos\left[\frac{\pi}{B}\left(y+\frac{1}{2}\right)v\right]. \tag{1}$$

This reports the DCT coefficient $Y_{u,v}$ representing the transformation of the input with sinusoids at frequency u and v in horizontal and vertical orientations, respectively. Basis functions are typically normalized with factors $\alpha_0 = 1$ and $\alpha_i = 2, i > 0$ to ensure orthogonality.

4 Harmonic Networks

A convolutional layer extracts correlation of input patterns with locally applied learned filters. The idea of convolutions applied to images stems from the observation that pixels in local neighborhoods of natural images tend to be strongly correlated. In many image analysis applications, transformation methods are used to decorrelate signals forming an image [10]. In contrast with spatial convolution with learned kernels, this study proposes feature learning by weighted combinations of responses to predefined filters. The latter extract harmonics from lower-level features in a region. The use of well selected predefined filters allows one to reduce the impact of overfitting and decrease computational complexity.

A *harmonic block* is proposed to replace a conventional convolutional operation and relies on processing the data in two stages, see Fig. 1: Firstly, the input features undergo harmonic decomposition by a transformation method. Conceptually, various transformation methods can be used e.g. wavelets, derivatives of Gaussians, etc. In this study we focus on window-based DCT. In the second stage, the transformed signals are combined by learned weights. The fundamental difference from standard convolutional network is that the optimization algorithm is not searching for filters that extract spatial correlation, rather learns the relative importance of preset feature extractors (DCT filters) at multiple layers.

Harmonic blocks are integrated as a structural element in the existing or new CNN architectures. Specifically, we design harmonic networks that consist of one or more harmonic blocks and, optionally, standard learned convolutions and fully-connected layers. Spectral decomposition of input features into block-DCT representation is implemented as a convolution with DCT basis functions. A 2D kernel with size $K \times K$ is constructed for each basis function, comprising a filter bank of depth K^2, which is separately applied to each of the input features. Convolution with the filter bank isolates coefficients of DCT basis functions to their exclusive feature maps, creating a new feature map per each channel and each frequency considered. The number of operations required to calculate this representation can be minimized by decomposing 2D DCT filter into two rank-1 filters and applying them as separable convolution to rows and columns sequentially. Despite the operation being computationally cheaper compared to dense convolutions, the spectral decomposition upsamples

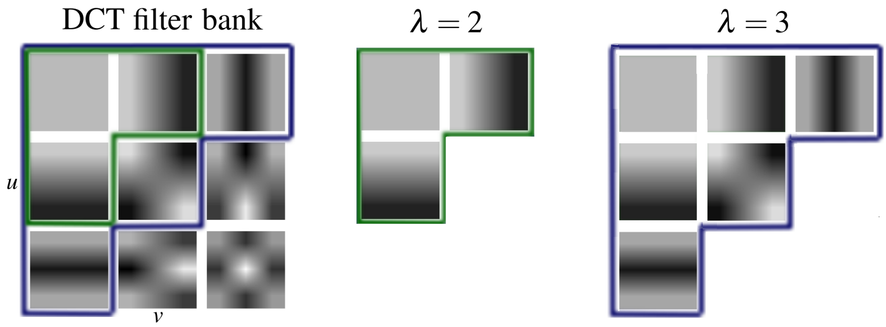

Figure 2: 3x3 DCT filter bank employed in the harmonic networks and its compression.

the number of intermediate features by K^2 factor, thus notably increasing the corresponding memory requirements.

Each feature map h^l at depth l is computed as a weighted linear combination of DCT coefficients across all input channels N:

$$h^l = \sum_{n=0}^{N-1} \sum_{u=0}^{K-1} \sum_{v=0}^{K-1} w^l_{n,u,v} \psi_{u,v} ** h_n^{l-1} \tag{2}$$

where $\psi_{u,v}$ is a u,v frequency selective DCT filter of size $K \times K$, $**$ the 2-dimensional convolution operator and $w^l_{n,u,v}$ is learned weight for u,v frequency of the n-th feature. The linear combination of spectral coefficients is implemented via a convolution with 1x1 filter that scales and sums the features, see Fig. 1. Since the DCT is a linear transformation, backward pass through the transform layer is performed similarly to a backward pass through a convolution layer. Harmonic blocks are designed to learn the *same* number of parameters as their convolutional counterparts. Such blocks can be considered a special case of depth-separable convolution with predefined spatial filters.

DCT is distinguished by its energy compaction capabilities which typically results in higher filter responses in lower frequencies. The undesirable behaviour of relative loss of high frequency information can be efficiently handled by normalizing spectrum of the input channels. This can be achieved via batch normalization that adjusts per frequency mean and variance prior to the weighted combination. The spectrum normalization transforms Eq. 2 into:

$$h^l = \sum_{n=0}^{N-1} \sum_{u=0}^{K-1} \sum_{v=0}^{K-1} w^l_{n,u,v} \frac{\psi_{u,v} ** h_n^{l-1} - \mu^l_{n,u,v}}{\sigma^l_{n,u,v}}, \tag{3}$$

with parameters $\mu^l_{n,u,v}$ and $\sigma^l_{n,u,v}$ estimated per input batch.

The JPEG compression encoding relies on stronger quantisation of higher frequency DCT coefficients. This is motivated by the human visual system which often prioritises low frequency information over high frequencies. We propose to employ similar idea in the harmonic network architecture. Specifically, we limit the visual spectrum of harmonic blocks to only several most informative low frequencies, which results in a reduction of number of parameters and operations required at each block. The coefficients are (partially) ordered by their relative importance for the visual system in triangular patterns starting at the most important zero frequency at the top-left corner, see Fig. 2. We limit the spectrum of considered frequencies by hyperparameter λ representing the number of levels of coefficients

Method	dropout	compression	parameters	CIFAR-10	CIFAR-100
WRN-28-10 [36]	✓		36.5M	3.91	18.75
Harm1-WRN-28-10 (no BN)			36.5M	4.10	19.17
Harm1-WRN-28-10			36.5M	3.90	18.80
Harm1-WRN-28-10	✓		36.5M	**3.64**	**18.57**
Harm-WRN-28-10	✓		36.5M	3.86	**18.57**
Harm-WRN-28-10	✓	$\lambda = 3$	24.4M	3.84	18.58
Harm-WRN-28-10	✓	$\lambda = 2$	12.3M	4.25	19.97
Harm-WRN-28-10		progressive λ	15.7M	3.93	19.04
Gabor CNN 3-28 [20]			17.6M	3.88	20.13
WRN-28-8 [36]	✓		23.4M	4.01	19.38
WRN-28-6 [36]	✓		13.1M	4.09	20.17

Table 1: Settings and median error rates (%, out of 5 runs) achieved by WRNs and their harmonic modifications on CIFAR datasets. Number of parameters reported for CIFAR-10.

included perpendicularly to the main diagonal direction starting from zero frequency: DC only for $\lambda = 1$, 3 coefficients (green) for $\lambda = 2$, and 6 coefficients (purple) for $\lambda = 3$. Fig. 2 illustrates filters used at various levels assuming a 3x3 receptive field.

5 Experiments

In this section we validate the performance of the harmonic networks on CIFAR-10/100 and ImageNet-1K datasets. We will consider two typologies of Residual Networks [14, 36] as the baselines for substituting the standard convolution operations with harmonic blocks[1].

5.1 CIFAR-10/100 datasets

The first set of experiments is performed on popular benchmark datasets of small natural images CIFAR-10 and CIFAR-100. Images have three color channels and resolution of 32x32 pixels. The dataset is split into 50k images for training and 10k for testing. Images have balanced labeling, 10 classes in CIFAR-10 and 100 in CIFAR-100.

Baseline. For experiments on CIFAR datasets we adopt WRNs [36] with 28 layers and width multiplier 10 (WRN-28-10) as the main baseline. These improve over the standard ResNets by using much wider residual blocks instead of extended depth. Model design and training procedure are kept unchanged as in the original paper. Harmonic WRNs are constructed by replacing convolutional layers by harmonic blocks with the same receptive field, preserving batch normalization and ReLU activations in their original positions after every block.

Results. We first investigate whether the WRN results can be improved if trained on spectral information, i.e. when replacing only the first convolutional layer preceding the residual blocks in WRN by a harmonic block with the same receptive field (Harm1-WRN). The network learns more useful features if the RGB spectrum is explicitly normalized by integrating the BN block as demonstrated in Fig. 1, surpassing the classification error of the baseline network on both CIFAR-10 and CIFAR-100 datasets, see Table 1. We then construct a fully

[1] The PyTorch implementation of the harmonic network is publicly available at https://github.com/matej-ulicny/harmonic-networks.

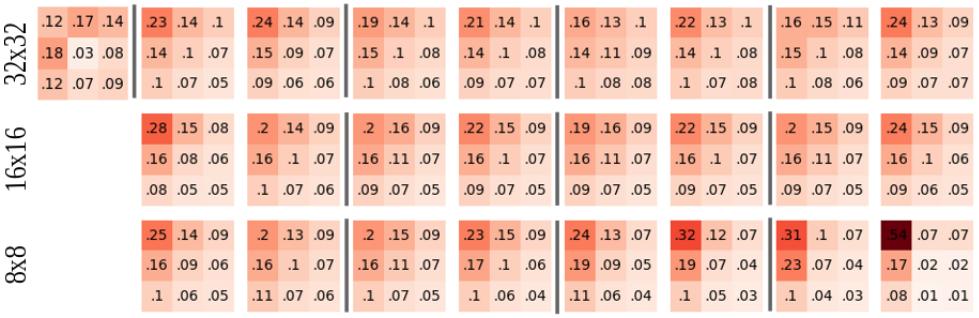

Figure 3: Distribution of weights (averaged in each layer) assigned to DCT filters in the first harmonic block (left-most) and the remaining blocks in the Harm-WRN-28-10 model trained on CIFAR-10. Vertical lines separate the residual blocks.

harmonic WRN (denoted as Harm-WRN) by replacing all convolutional layers with harmonic blocks, retaining the residual shortcut projections unchanged. *Zagoruyko et al.* [36] demonstrated how dropout blocks placed inside residual blocks between convolutional layers can provide extra regularization when trained on spatial data [36]. We have observed a similar effect when training on spectral representations, therefore we adopt dropout between harmonic blocks. The harmonic network outperforms the baseline WRN, see Table 1. Based on this empirical evidence we always employ BN inside the first harmonic block.

Analysis of fully harmonic WRN weights learned with 3x3 spectrum revealed that the deeper network layers tend to favour low-frequency information over high frequencies when learning representations. Relative importance of weights corresponding to different frequencies shown in Fig. 3 motivates truncation of high-frequency coefficients for compression

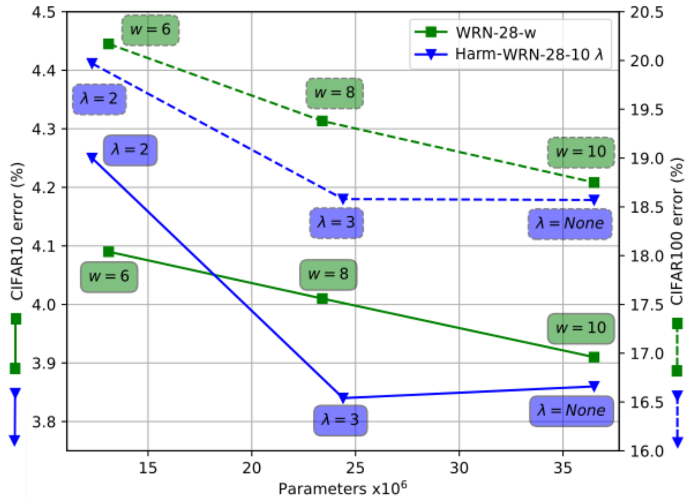

Figure 4: Graphs show a decrease of classification error as a function of model size on CIFAR-10 (solid lines) and CIFAR-100 (dashed). Parameters of harmonic networks are controlled by the compression parameter λ, the WRN baselines by the width multiplier w.

purposes. While preserving the input image spectrum intact, we train the harmonic networks on limited spectrum of hidden features for λ=2 and λ=3 using 3 and 6 DCT filters for each feature, respectively. To assess the loss of accuracy associated with parameter reduction we train baselines with reduced widths having comparable numbers of parameters: WRN-28-8 and WRN-28-6, see Fig. 4. Fully harmonic WRN-28-10 with λ=3 has comparable error to the network using the full spectrum and outperforms the larger baseline WRN-28-10, showing almost no loss in discriminatory information. On the other hand Harm-WRN-28-10 with λ=2 is better on CIFAR-100 and slightly worse on CIFAR-10 compared to the similarly sized WRN-28-6. The performance degradation indicates that some of the truncated coefficients carry important discriminatory information. Detailed comparison is reported in Table 1.

We further compare the performance of the harmonic version of WRN-28-10 with the Gabor CNN 3-28 [20] that relies on modulating the learned filters with Gabor orientation filters. To operate on a similar model we remove dropouts and reduce complexity by applying progressive λ: no compression for 32x32 feature sizes, λ=3 for 16x16, and λ=2 for the rest. With a smaller number of parameters the Harm-WRN-28-10 performs similarly on CIFAR-10 and outperforms Gabor CNN on CIFAR-100.

5.2 ImageNet dataset

In this section we present results obtained on ImageNet-1K classification task. To deploy harmonic networks on large-scale datasets a few adjustments are applied to the harmonic blocks: Firstly, in the absence of BN inside harmonic block we merge the linear operations of feature extraction with DCT basis and weighted combination of responses. This prevents the need to allocate memory for intermediate features and is applied to all layers except for the first (where BN is employed). Secondly, we normalize DCT filters by their L1 norm.

ResNet [14] with 50 layers is adopted as the baseline. Following [11] we apply stride on 3x3 convolution instead of the first 1x1 convolution in the block. To reduce memory consumption maxpooling is not used, instead the first convolution layer employs stride 4 to produce equally-sized features; we refer to this modification as ResNet-50 (no maxpool). Similarly to the above reported CIFAR experiments we investigate the performance of three harmonic modifications of the baseline: (i) replacing solely the initial 7x7 convolution layer with harmonic block (with BN) with 7x7 DCT filters, (ii) replacing all convolution layers with receptive field larger than 1x1 with equally-sized harmonic blocks, (iii) compressed version of the fully-harmonic network. Each model is trained with stochastic gradient descent with learning rate 0.1, reduced 10 times every 30 epochs reporting the final accuracy

Type	Model	Parameters	Top-1 %	Top-5 %
fully trained	ResNet-50 (no maxpool)	25.6M	24.36	7.34
	Harm1-ResNet-50	25.6M	**23.34**	**6.75**
	Harm-ResNet-50	25.6M	23.58	6.91
	Harm-ResNet-50, progr. λ	19.7M	23.71	6.94
converted + finetuned	ResNet-50 ⇒ Harm-ResNet-50	25.6M	24.15	7.15
	ResNet-50 ⇒ Harm-ResNet-50, progr. λ	19.7M	24.60	7.43
benchmark	ResNet-50 (maxpool) [1]	25.6M	23.85	7.13
	ScatResNet-50 [23]	27.8M	25.5	8.0
	JPEG-ResNet-50 [12]	25.6M	23.94	6.98

Table 2: Classification errors on ImageNet validation set using central crops.

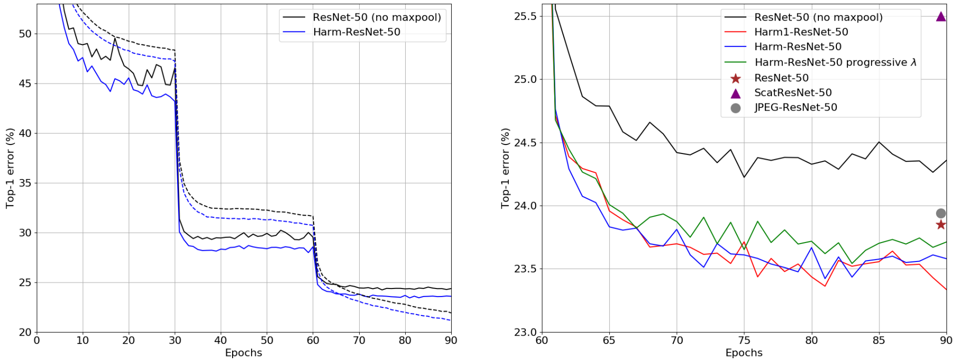

Figure 5: Training of harmonic networks on ImageNet classification task. Left: comparison with the baseline showing validation error (solid line) and training error (dashed). Right: last 30 epochs of training for all the models including scores reported for the benchmark models.

at epoch 90. We employ batch size of 256, weight decay 0.0001 and random scale, aspect ratio & horizontal flip augmentation as recommended in [28], producing 224×224 crops.

Table 2 reports error rates on ImageNet validation set using central 224×224 crops from images resized such that the shorter side is 256. All three harmonic networks have similar performance and improve over the baseline by $0.6-1\%$ in top1 and $0.4-0.6\%$ in top5 accuracy. We observe similar progress of the three modifications during training, see Fig. 5. ResNet-50 architecture has 17 layers with spatial filters which correspond to 11M parameters. We reduce this number by using progressive λ compression: λ=3 on 14x14 features and λ=2 on the smallest feature maps. This reduces the number of weights roughly by half, in total by about 23% of the network size. The compressed network loses about 0.25% in accuracy but still clearly outperforms the baseline. It should be noted that harmonic networks with less bottleneck blocks can be more efficiently compressed. Even with compression the proposed Harm-ResNet-50 confidently outperforms the standard ResNet-50 (maxpool), as well as the more recent ScatResNet-50 [23] and JPEG-ResNet-50 [12], see Table 2.

Finally, we evaluate the conversion of the weights of a pretrained non-harmonic network to those of its harmonic version. To this end each learned filter in the pretrained baseline (ResNet-50 without maxpooling after 90 epochs of training) is expressed as a best matching combination of DCT filters. We skip BN inside the first harmonic layer since the related parameters are not available. The direct conversion resulted in the exact same numerical performance due to the basis properties of DCT. We observe a similar pattern of relative importance of DCT filters to the one reported in Fig. 3. We then finetune the converted model for another 5 epochs with the learning rate of 0.001, which results in the top1 (top5) performance improvement of 0.21% (0.19%) over the pretrained baseline, see Table 2. We also investigate the conversion to a harmonic network with progressive λ compression. After casting the pretrained filters into the available number of DCT filters (from full basis at the early layers to 3 out of 9 filters at the latest layers), the top1 performance degrades by 6.3% due to loss of information. However, if we allow finetuning for as few as 5 epochs the top1 (top5) accuracy falls 0.24% (0.09%) short of the baseline by reducing the number of parameters by 23%. This analysis shows how the harmonic networks can be used to improve the accuracy and / or compress the existing pretrained CNN models.

6 Conclusion

We have presented a novel approach to incorporate spectral information from DCT into CNN models. To this end a harmonic architectural block has been proposed that extracts per-pixel spectral information from image features and learns weights to combine this information to form new representations. We empirically evaluate the use of harmonic blocks with the well-established state-of-the-art CNN architectures to validate the related improvement in classification accuracy as well as parametric complexity. We also ascertain that harmonic networks can be efficiently set-up by converting the pretrained CNN baselines. The use of DCT allows one to order the harmonic block parameters by their significance from most relevant low frequency to less important high frequencies. This enables efficient model compression by parameter truncation with only minor degradation in the model performance. This has been shown to be particularly useful for tasks with limited training samples [30].

Acknowledgement

This research was supported by the ADAPT Centre for Digital Content Technology funded under the SFI Research Centres Programme (Grant 13/RC/2106) and co-funded under the European Regional Development Fund. The second author was also supported by the European Union's Horizon 2020 research and innovation programme under the Marie Curie grant agreement No.713567. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPUs used for this research.

References

[1] Torchvision models. https://pytorch.org/docs/stable/torchvision/models.html.

[2] I. Amerini, T. Uricchio, L. Ballan, and R. Caldelli. Localization of JPEG double compression through multi-domain convolutional neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1865–1871, July 2017.

[3] M. Barni, L. Bondi, N. Bonettini, P. Bestagini, A. Costanzo, M. Maggini, B. Tondi, and S. Tubaro. Aligned and non-aligned double JPEG detection using convolutional neural networks. *J. Vis. Comun. Image Represent.*, 49(C):153–163, November 2017.

[4] Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE Trans. Pat. Anal. Mach. Intel.*, 35(8):1872–1886, 2013.

[5] Wenlin Chen, James Wilson, Stephen Tyree, Kilian Q. Weinberger, and Yixin Chen. Compressing convolutional neural networks in the frequency domain. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1475–1484, New York, NY, USA, 2016. ACM.

[6] D. D. N. De Silva, S Fernando, I. T. S. Piyatilake, and A. V. S Karunarathne. Wavelet based edge feature enhancement for convolutional neural networks. In *Eleventh International Conference on Machine Vision (ICMV 2018)*, volume 11041, page 110412R. International Society for Optics and Photonics, 2019.

[7] Meng Joo Er, W. Chen, and Shiqian Wu. High-speed face recognition based on discrete cosine transform and rbf neural networks. *Trans. Neur. Netw.*, 16(3):679–691, May 2005.

[8] Shin Fujieda, Kohei Takayama, and Toshiya Hachisuka. Wavelet convolutional neural networks for texture classification. *arXiv preprint arXiv:1707.07394*, 2017.

[9] A. Ghosh and R. Chellappa. Deep feature extraction in the DCT domain. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 3536–3541, Dec 2016.

[10] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Prentice Hall, Upper Saddle River, N.J., 3rd edition, 2008.

[11] Sam Gross and Michael Wilber. Training and investigating residual nets. *Facebook AI Research*, 2016. URL `https://github.com/facebook/fb.resnet.torch`.

[12] Lionel Gueguen, Alex Sergeev, Ben Kadlec, Rosanne Liu, and Jason Yosinski. Faster neural networks straight from JPEG. In *Advances in Neural Information Processing Systems*, pages 3933–3944, 2018.

[13] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *International Conference on Learning Representations (ICLR)*, 2016.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[15] Jorn-Henrik Jacobsen, Jan van Gemert, Zhongyu Lou, and Arnold WM Smeulders. Structured receptive fields in CNNs. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2610–2619, 2016.

[16] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Speeding up convolutional neural networks with low rank expansions. In *British Machine Vision Conference (BMVC)*. BMVA Press, 2014.

[17] Minyoung Kim and Luca Rigazio. Deep clustered convolutional kernels. In *Feature Extraction: Modern Questions and Challenges*, pages 160–172, 2015.

[18] Bin Li, Haoxin Zhang, Hu Luo, and Shunquan Tan. Detecting double jpeg compression and its related anti-forensic operations with cnn. *Multimedia Tools and Applications*, 78(7):8577–8601, 2019.

[19] H. Lu, H. Wang, Q. Zhang, D. Won, and S. W. Yoon. A dual-tree complex wavelet transform based convolutional neural network for human thyroid medical image segmentation. In *2018 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 191–198, June 2018.

[20] Shangzhen Luan, Chen Chen, Baochang Zhang, Jungong Han, and Jianzhuang Liu. Gabor convolutional networks. *IEEE Trans. Image Process.*, 27(9):4357–4366, 2018.

[21] Edouard Oyallon and Stephane Mallat. Deep roto-translation scattering for object classification. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[22] Edouard Oyallon, Eugene Belilovsky, and Sergey Zagoruyko. Scaling the scattering transform: Deep hybrid networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[23] Edouard Oyallon, Eugene Belilovsky, Sergey Zagoruyko, and Michal Valko. Compressing the input for CNNs with the first-order scattering transform. In *European Conference on Computer Vision (ECCV)*, 2018.

[24] Qiang Qiu, Xiuyuan Cheng, Robert Calderbank, and Guillermo Sapiro. DCFNet: Deep neural network with decomposed convolutional filters. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 4198–4207. PMLR, Jul 2018. URL http://proceedings.mlr.press/v80/qiu18a.html.

[25] Oren Rippel, Jasper Snoek, and Ryan P Adams. Spectral representations for convolutional neural networks. In *Advances in neural information processing systems*, pages 2449–2457, 2015.

[26] S. Said, O. Jemai, S. Hassairi, R. Ejbali, M. Zaied, and C. Ben Amar. Deep wavelet network for image classification. In *2016 IEEE International Conference on Systems, Man, and Cybernetics*, pages 922–927, Oct 2016.

[27] Amarjot Singh and Nick Kingsbury. Efficient convolutional network learning using parametric log based dual-tree wavelet scatternet. In *Computer Vision Workshop (ICCVW), 2017 IEEE International Conference on*, pages 1140–1147. IEEE, 2017.

[28] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[29] Matej Ulicny and Rozenn Dahyot. On using CNN with DCT based image data. In *Irish Machine Vision and Image Processing Conference*, 2017.

[30] Matej Ulicny, Vladimir A Krylov, and Rozenn Dahyot. Harmonic networks with limited training samples. In *European Signal Processing Conference (EUSIPCO) 2019*, Sep. 2019. URL https://arxiv.org/abs/1905.00135.

[31] Qing Wang and Rong Zhang. Double JPEG compression forensics based on a convolutional neural network. *EURASIP J. on Information Security*, 2016(1):23, Oct 2016.

[32] Yunhe Wang, Chang Xu, Shan You, Dacheng Tao, and Chao Xu. CNNpack: Packing convolutional neural networks in the frequency domain. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 253–261. Curran Associates, Inc., 2016.

[33] T. Williams and R. Li. Advanced image classification using wavelets and convolutional neural networks. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 233–239, Dec 2016.

[34] Travis Williams and Robert Li. Wavelet pooling for convolutional neural networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[35] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 5028–5037, 2017.

[36] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proc. of British Machine Vision Conference (BMVC)*, pages 87.1–87.12. BMVA Press, September 2016.

[37] Xiaoyi Zou, Xiangmin Xu, Chunmei Qing, and Xiaofen Xing. High speed deep networks based on discrete cosine transformation. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 5921–5925. IEEE, 2014.