Harmonic Networks with Limited Training Samples

Matej Ulicny, Vladimir A. Krylov, Rozenn Dahyot
ADAPT Centre, School of Computer Science & Statistics, Trinity College Dublin, Dublin, Ireland
{ulinm, Vladimir.Krylov, Rozenn.Dahyot}@tcd.ie

***Abstract*—Convolutional neural networks (CNNs) are very popular nowadays for image processing. CNNs allow one to learn optimal filters in a (mostly) supervised machine learning context. However this typically requires abundant labelled training data to estimate the filter parameters. Alternative strategies have been deployed for reducing the number of parameters and / or filters to be learned and thus decrease overfitting. In the context of reverting to preset filters, we propose here a computationally efficient harmonic block that uses Discrete Cosine Transform (DCT) filters in CNNs. In this work we examine the performance of harmonic networks in limited training data scenario. We validate experimentally that its performance compares well against scattering networks that use wavelets as preset filters.**

***Index Terms*—Lapped Discrete Cosine Transform, harmonic network, convolutional filter, limited data**

I. Introduction

We have recently proposed a new form of neural network layer called harmonic block [1] that relies on using windowed cosine transform at several frequencies in lieu of learned filters. This harmonic block only involves learning weights for combining several frequency responses together in the frequency domain. Furthermore, uninformative frequencies can be dropped out to improve the computational complexity of the network without compromising performance, i.e. compression [1]. This paper extends further the proposed harmonic block by: 1) showing how it relates to the modified discrete cosine transform when considering overlap in computing convolution, 2) proposing an improved, computationally more efficient implementation, and 3) showing that the CNNs using the harmonic block outperform scattering network, based on the use of wavelet-based filters [2], [3] when training data is scarce. The PyTorch implementation of the harmonic block is provided at *https://github.com/matej-ulicny/harmonic-networks*.

The rest of the paper is organised as follows. We first review the related literature (Sec. II) and present the harmonic block (Sec. III). We then report the experimental validation (Sec. IV) and conclusions of the study (Sec. V).

II. Related work

A. DCT & CNNs

Wang and Zhang [4] propose a double JPEG compression detection algorithm based on a convolutional neural network (CNN) to detect tampered area for image forensics. The 1-dimensional CNN is designed to classify histograms of discrete cosine transform (DCT) coefficients, which differ between single-compressed areas (untampered areas) and double-compressed areas (tampered areas) [4]–[6]. Alternatively, raw DCT (discrete cosine transform) coefficients from JPEG images has also been proposed as input of a 2-dimensional CNN [7]. Spectral image representations combined with neural networks have also been used for object recognition. For instance, truncation of DCT coefficients has been shown to speed up training of fully connected sparse autoencoders [8] and improve face recognition with linear discriminant analysis and radial basis function network [9]. DCT transform has been used in conjunction with CNNs for image classification as an input pre-processing step [10], [11]. Ghosh and Chellappa [12] transformed feature maps inside the CNN pipeline and noted convergence improvements.

B. Wavelets & CNNs

Common approach in literature is to use wavelet transform to extract invariant features prior to classification. One such example is the Scattering convolution network composed of complex Morlet wavelet filters [2] and a PCA or SVM classifier. Wavelet responses were also used with NN-based classifier [13], or with a set of CNNs each operating on exclusive frequency sub-band [14]. Silva et al. used wavelet filters to enhance edges prior to CNN processing [15]. Rotation and scale invariant wavelet based scattering networks with subsequent CNN were formulated in [3], [16]. These hybrid networks were shown to reach comparable classification accuracy to deeper CNNs.

Several studies incorporated wavelets in CNN computational graphs. New feature pooling strategies were designed based on fast Fourier transform [17] or fast wavelet transform [18]. Haar wavelet responses of the input image have been concatenated to features at different stages of CNN to address texture classification [19]. Lu et al. [20] designed a similar approach for medical image segmentation, however based on dual-tree complex wavelets. Robustness to scale and orientation of CNN is increased by modulating learned filters by a set of Gabor filters [21]. Rotation equivariance of learned features was accomplished by incorporated complex circular

This work is supported by the ADAPT Centre for Digital Content Technology funded under the Science Foundation Ireland Research Centres Programme grant 13/RC/2106 and co-funded under the European Regional Development Fund. The second author is also supported by the European Unions Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No.713567.

978-9-0827-9703-9/19/$31.00 ©2019 IEEE

harmonics into CNNs [22]. Jacobsen et al. proposed to learn convolution filters as a composition of Gaussian derivative filter basis [23].

C. Compressing CNNs

Compression of neural networks has received a lot of attention from researchers. Jaderberg et al. [24] approximated full-rank CNN filters by separable rank-1 filters. DCT transform has been used for model compression, to cluster weights into buckets based on their DCT representation [25], or to represent weights as residuals from their cluster centers in DCT domain [26].

III. Harmonic block

A. Overlapping cosine transform

DCT computed on overlapping windows is also known as Lapped Transform or Modified DCT (MDCT), equivalent to our harmonic block using strides. The overlapped DCT has a long history in signal compression and reduces artefacts at window edges [27]. Dedicated strategies for efficient computations have been proposed [27], including algorithms and hardware optimisations. Our current implementation uses standard deep learning libraries (PyTorch) and is not currently taking full advantage of these more advanced DCT implementations.

DCT transform is equivalent to the discrete Fourier transform of real valued functions with even symmetry within twice larger window. DCT lacks imaginary component given by the sine transform of real valued odd functions. However, harmonic block allows convolution with DCT basis with arbitrary stride creating redundancy in the representation. Ignoring the boundary limitations, sine filter basis can be devised by shifting the cosine filters. Given the equivariant properties of convolution, instead of shifting the filters the same result is achieved by applying original filters to the shifted input. Considering DCT-II formulation:

$$F_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right] \tag{1}$$

a corresponding sine transform is

$$G_k = \sum_{n=0}^{N-1} x_n \sin\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right] \tag{2}$$

which is equivalent to

$$G_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{2}+2\pi z-\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right]. \tag{3}$$

The shift given by $\pi/2 + 2\pi z$ for any $z \in \mathbb{Z}$ can be directly converted to shift in pixels applied to data x. After simplification, sine transform can be expressed as

$$G_k = \sum_{n=0}^{N} x_n \cos\left[\frac{\pi}{N}\left(n-\frac{N(1+4z)}{2k}+\frac{1}{2}\right)k\right] \tag{4}$$

which is equivalent to the cosine transform of the image shifted by $\delta = N\left(1+4z\right)/2k$ defined in (5).

$$F_k[\delta] = \sum_{n=0}^{N} x_{n+\frac{N(1+4z)}{2k}} \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right]. \tag{5}$$

This value represents the stride to shift the cosine filters to capture correlation with sine function.

B. Definition of harmonic block

The harmonic block [1] is designed to replace fully learned convolution of multidimensional input features h^{l-1}. Input channels $h_n^{l-1}, n \in \{0..N-1\}$ are convolved using the DCT basis functions $\phi_{u,v}$ given size of the desired receptive field $K \times K$:

$$\phi_{u,v}\left(x,y\right) = \cos\left[\frac{\pi}{K}\left(x+\frac{1}{2}\right)u\right]\cos\left[\frac{\pi}{K}\left(y+\frac{1}{2}\right)v\right]. \tag{6}$$

Specifically, we employ $L1$-normalised filters $\psi_{u,v} \in \mathcal{R}^{K\times K}$:

$$\psi_{u,v} = \frac{\phi_{u,v}}{\left\|\phi_{u,v}\right\|_1}. \tag{7}$$

Due to properties of natural images, high frequency responses are generally of lower magnitude. Employing batch normalization (BN) on DCT coefficients of the RGB channels has been found useful [1] for propagating energy of the whole spatial-frequency spectrum. Output features $h_m^l, m \in \{0..M-1\}$ are learned as superpositions of the DCT coefficients, described in detail in Algorithm 1, where the learned parameters inside each harmonic block are denoted as $w \in \mathcal{R}^{M\times N\times K\times K}$.

The downside of Algorithm 1 is that in order to be executed in parallel, extra memory has to be allocated to store the responses of DCT filters at every layer. Since most of the blocks do not need to use BN they become linear. Hence DCT transform and linear combination can be merged into a single linear operation. In other words, equivalent features can be obtained by factorizing filters as linear combination of DCT basis functions. Therefore we propose here Algorithm 2 that is a more efficient alternative to Algorithm 1. This reformulation

Algorithm 1: Harmonic block

```
Input: $h^{l-1}$
for $n \in \{0, \cdots, N-1\}$ do
    $z_{n,u,v}^l \leftarrow \sum_{u=0}^{K-1}\sum_{v=0}^{K-1} \psi_{u,v} ** h_n^{l-1}$
    if normalize then
        $\mu_{n,u,v}^l, \sigma_{n,u,v}^l \leftarrow$ estimate mean and standard
          deviation of $z_{n,u,v}^l$ over the batch dimension
        $z_{n,u,v}^l \leftarrow \frac{\left(z_{n,u,v}^l - \mu_{n,u,v}^l\right)}{\sigma_{n,u,v}^l}$
    end
end
for $m \in \{0 \ldots M-1\}$ do
    $h_m^l \leftarrow \sum_{n=0}^{N-1}\sum_{v=0}^{K-1}\sum_{u=0}^{K-1} w_{m,n,u,v}\ z_{n,u,v}$
end
Output: $h^l$
```

is similar to structured receptive field [23] utilizing different basis functions. The theoretical number of multiply-add operations compared to the standard convolutional layer increases by a factor of K^2/M for Algorithm 1, and by K^2/AB for Algorithm 2, where the input image size for the block is $A \times B$. The experimental performance of the two algorithms is compared in Section IV-A.

Algorithm 2: Memory efficient harmonic block

```
Input: $h^{l-1}$
Define updates $g \in \mathcal{R}^{M \times N \times K \times K}$;
for $m \in \{0..M-1\}$ do
    for $n \in \{0..N-1\}$ do
        $g^l_{m,n} \leftarrow \sum_{u=0}^{K-1} \sum_{v=0}^{K-1} w_{m,n,u,v} \ \psi_{u,v}$;
    end
end
$h^l \leftarrow g^l ** h^{l-1}$;
Output: $h^l$
```

Control over the filters allows one to achieve reduced computational complexity by selecting subsets of filters to approximate the signal. A λ-subset is a collection of all filters $\psi_{u,v}$ such that their indices u, v satisfy the condition $u+v < \lambda$. Fig. 1 shows example of some subsets of 3-by-3 DCT filters.

IV. Experimental evaluation

A. Computational requirements

Firstly we compare the two implementations of a harmonic block, see Sec. III-B. Experiment is conducted on well performing wide residual network (WRN) [28] trained on CIFAR10 dataset. The baseline WRN 16-8 (for architecture details and training procedure see [28]) with dropout rate 0.2 is compared with harmonic WRN with all convolution layers replaced by blocks defined in Algorithm 1 with additional BN in the first block. The network runtime and memory requirements for Algorithm 1 far exceed those of the baseline WRN (implemented via deep learning framework and run on GPUs) despite being more flexible and having similar amount of arithmetic operations, see discussion in [1]. Fully harmonic WRN based on Algorithm 2 (except the first layer due to the presence of BN) largely outperforms Algorithm 1 and shows only a modest increase in runtime and memory usage over the baseline WRN [28] while having competitive performance.

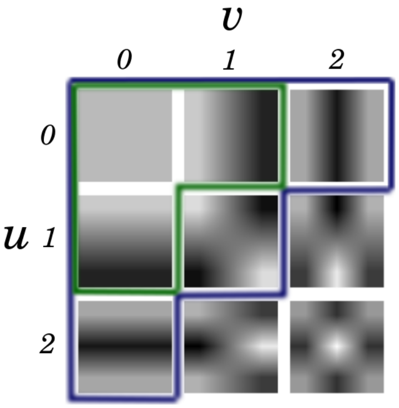


Fig. 1: DCT filter bank employed in harmonic blocks. Blue (green) color filters are used to produce features for λ=3 (λ=2).

TABLE I: Computational requirements of harmonic block implementations on CIFAR10. Accuracy shown is an average over 5 runs with empirical one standard deviation interval.

Model	Ref.	GPU mem.	epoch runtime	acc.
WRN 16-8	[28]	2.8GB	45.0s	95.61±0.14
Harm WRN 16-8 (Alg. 1)	[1]	6.6GB	123.4s	95.56±0.04
Harm WRN 16-8 (Alg. 2)		2.9GB	56.8s	95.62±0.09

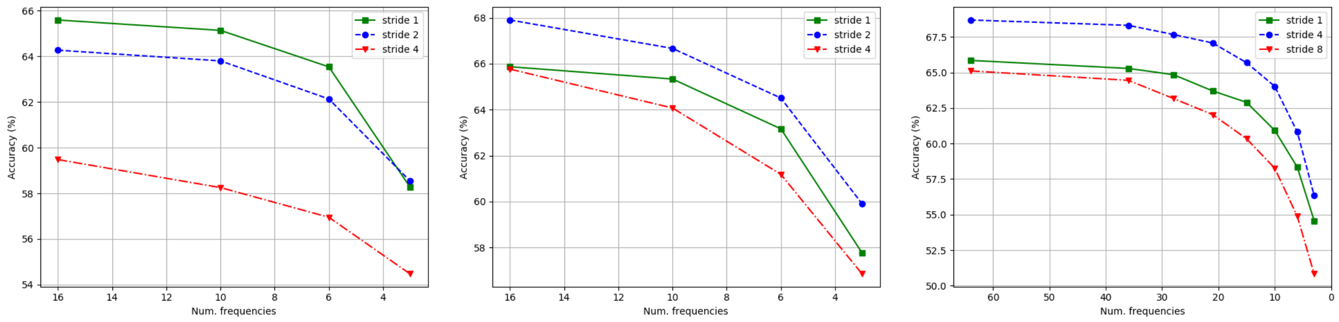
B. Overlapping DCT experiments

In Section III-A we demonstrated that the discrete sine transform can be inferred from the DCT on overlapping blocks. Here we show experimentally the benefits of DCT transform with overlapping windows by using overcomplete representation with strides of 1 pixel or fixing stride to the half of the window size. Effect of striding is evaluated on a shallow harmonic network composed of only one normalized harmonic block with 4x4 receptive field, followed by a Rectified Linear Unit (ReLU) activation and connected to a fully connected layer with softmax classifier. This simple architecture allows one to clearly see the contribution of striding. The network is trained with SGD using learning rate 0.01, Nesterov momentum 0.9, weight decay 0.0005 and batch size 128 for 30 epochs decaying learning rate by factor 10 halfway. Since striding reduces the spatial resolution of the features, to match the model complexity, lower dimensional features are resized to have size of features produced by stride 1. As expected, network without overlapping windows performs notably worse even with full spectrum (see Fig. 2a).

In order to compare models with similar numbers of parameters, instead of replicating features, networks with larger stride employ a higher number of output features: 200 for non-overlapping, 50 for half-window overlap in contrast to 16 when using stride 1. The same experiment is performed using 8x8 filters learning 625, 200 and 16 feature maps respectively. In this setting network with stride 1 and with full window stride perform comparably on full spectrum as can be seen on Fig. 2b and Fig. 2c, but performance degrades more rapidly for non-overlapping filters as the visual spectrum shrinks. The best result was obtained when using half window stride.

C. Limited Data

Deep neural networks require abundant data to achieve high accuracy. It has been shown in [2], [3] that scattering network using geometric priors can learn better discrimination boundaries when presented with a small subset of training samples. We demonstrate capabilities of harmonic networks when learning from limited subsets of data on three datasets.



(a) 4x4 replicated features

(b) 4x4 balanced block

(c) 8x8 balanced block

Fig. 2: Accuracy degradation of models with different strides when truncating number of DCT coefficients. Stride 1 (green), half window stride (blue) and full window stride (red) are compared. Reported values are averaged over 5 runs.

TABLE II: Classification errors in % (median of 21 runs) on subsets of MNIST dataset for harmonic network and benchmarks.

Training size	Scat. net. [2]	Conv. net.	Sep. conv. net.	Harm. net.
300	4.7	3.9	4.67	**3.71**
1000	2.3	1.88	1.91	**1.84**
2000	1.3	1.39	1.35	**1.21**
5000	1.03	0.97	1.06	**0.86**
10000	0.88	0.7	0.76	**0.65**
20000	0.58	0.59	**0.57**	**0.57**
40000	0.53	0.48	0.47	**0.45**
60000	0.43	0.44	0.46	**0.38**

1) MNIST: Bruna and Mallat [2] have chosen a dataset of handwritten digits to test their fully handcrafted scattering network with respect to stability to deformations and classification performance on data subsets. We compare our harmonic network to the "classical" CNN, learned depth-separable convolution network and to the fully handcrafted scattering network (as reported by [2]). Table II shows the harmonic network achieves the lower classification error for all sizes of the training set. The baseline network is composed of 3 convolution layers with 32, 64 and 128 3×3 filters, respectively, and with overlapping average pooling between them. Convolutional layers are followed by a fully connected layer with 512 neurons. Batch normalization and ReLU are applied after each layer. The harmonic network uses the same configuration replacing convolution with harmonic block while using additional BN in the first block. Harmonic networks are also compared to the depth-separable convolution network that has the same structure but has randomly initialized learnable filters instead of DCT filters. Training is done with SGD for 30 epochs with learning rate 0.1 reduced after every 10 epochs by a factor 10. Weight decay ranges from 0.0005 (for training size 60000) to 0.05 (training size 300). Harmonic networks outperform other networks in all configurations, see Tab. II.

2) CIFAR10: We replicate the experiment in [3] and train harmonic network on random subsets of CIFAR10 dataset with size 100, 500 and 1000 samples preserving equal number of labels per class. Harmonic WRN 16-8 with dropout rate 0.2 is trained as in [3]. Harmonic layers relying on combinations of fixed filters give advantage on limited data compared to fully learned CNNs and to scattering CNN hybrids[1] except for the smallest training dataset, see Tab. III.

TABLE III: Average classification accuracy ± standard deviation of 5 runs on subsets of CIFAR10.

Method	**100**	**500**	**1000**	**Full**
WRN 16-8	34.4±1.8	52.2±1.8	62.8±0.7	**95.6**
Scat + WRN [3]	**38.9±1.2**	54.7±0.6	62.0±1.1	93.1
Harm WRN 16-8	37.7±1.9	58.2±1.4	67.0±0.4	**95.6**
Harm WRN 16-8 $\lambda = 3$	37.9±2.4	**58.4±0.9**	**67.2±0.5**	**95.6**
Harm WRN 16-8 $\lambda = 2$	37.2±1.7	57.0±1.0	65.9±0.8	95.3

TABLE IV: Average classification accuracy ± standard deviation of 5 runs on STL10 (batch size 32).

Method	**10-folds**	**all**
WRN 16-8	73.50 ± 0.87	87.29 ± 0.21
Scat + WRN [3]	76.00 ± 0.60	87.60
Harm WRN 16-8	76.95 ± 0.93	**90.45 ± 0.12**
Harm WRN 16-8 $\lambda = 3$	76.65 ± 0.90	90.39 ± 0.08
Harm WRN 16-8 progressive λ	**77.19 ± 1.02**	90.28 ± 0.20

3) STL10: STL10 [29] is a natural image dataset similar to CIFAR10. Images are 96×96 and only 5000 training images are labeled. The large set of provided unlabeled images is not utilized in this experiment. We design harmonic WRN 16-8 model (based on Algorithm 2) for this task with several necessary modifications. The first layer uses stride 2, and the feature resolution at the final stage is 12×12. We apply dropout 0.3 inside residual blocks and train the network on the whole training set with learning rate of 0.1 decayed by factor 0.2 after 300, 400, 600, 800 epochs, and stopping the training after 1000. The baseline network design and training procedure is similar to [30] that uses additional cutout regularization and reports 87.26% ± 0.23 on test set containing 8000 images when trained on batches of 128 images. The harmonic WRN 16-8 achieves 88.1% ± 0.23 trained with the same settings. Decreasing the batch size to 32 improves our result to 90.45% surpassing the deeper scattering WRN [3] by nearly 3%. Furthermore, when only predefined folds of 1000 samples

[1] The exact subsets used to train scattering CNN hybrids are not known, we report the numerical results from [3].

serve as the training data, we obtain the best accuracy by progressively reducing the number of used frequencies along with spatial resolution: full filter bank is applied on features of size 48×48, filters with $\lambda = 3$ on 24×24 and finally $\lambda = 2$ if features are 12×12. The results of STL10 experiments are summarised in Tab. IV.

V. Conclusion

We have proposed a computationally efficient alternative to the original harmonic block based on DCT [1]. The implementation is characterized by a very small increase in the number of multiply-add operations compared to a standard convolutional layer, thus enabling the wider use of harmonic networks as a tool for reducing model overfitting. The experimental results reported in this manuscript confirm that the harmonic block outperforms the well established scattering networks using wavelets [2], [3] when limited data is available for training. We provide the PyTorch implementation of the improved harmonic block. Future work will investigate the effect of window functions that are also often used in Modified DCT as part of the harmonic block, and test its performance in large scale experiments.

References

[1] M. Ulicny, V. A. Krylov, and R. Dahyot, "Harmonic networks: Integrating spectral information into CNNs," *arXiv preprint arXiv:1812.03205*, 2018.

[2] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1872–1886, 2013.

[3] E. Oyallon, S. Zagoruyko, G. Huang, N. Komodakis, S. Lacoste-Julien, M. B. Blaschko, and E. Belilovsky, "Scattering networks for hybrid representation learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2018.

[4] Q. Wang and R. Zhang, "Double JPEG compression forensics based on a convolutional neural network," *EURASIP Journal on Information Security*, vol. 2016, p. 23, Oct 2016.

[5] 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)*, pp. 1865–1871, July 2017.

[6] 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.*, vol. 49, pp. 153–163, Nov. 2017.

[7] B. Li, H. Luo, H. Zhang, S. Tan, and Z. Ji, "A multi-branch convolutional neural network for detecting double JPEG compression," *CoRR*, vol. abs/1710.05477, 2017.

[8] X. Zou, X. Xu, C. Qing, and X. Xing, "High speed deep networks based on discrete cosine transformation," in *Image Processing (ICIP), 2014 IEEE International Conference on*, pp. 5921–5925, IEEE, 2014.

[9] M. J. Er, W. Chen, and S. Wu, "High-speed face recognition based on discrete cosine transform and rbf neural networks," *Trans. Neur. Netw.*, vol. 16, pp. 679–691, May 2005.

[10] M. Ulicny and R. Dahyot, "On using CNN with DCT based image data," in *Irish Machine Vision and Image Processing Conference*, 2017.

[11] L. Gueguen, A. Sergeev, B. Kadlec, R. Liu, and J. Yosinski, "Faster neural networks straight from jpeg," in *Advances in Neural Information Processing Systems*, pp. 3937–3948, 2018.

[12] A. Ghosh and R. Chellappa, "Deep feature extraction in the dct domain," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 3536–3541, Dec 2016.

[13] S. Said, O. Jemai, S. Hassairi, R. Ejbali, M. Zaied, and C. B. Amar, "Deep wavelet network for image classification," in *2016 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 922–927, Oct 2016.

[14] 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)*, pp. 233–239, Dec 2016.

[15] D. D. N. D. Silva, S. Fernando, I. T. S. Piyatilake, and A. V. S. Karunarathne, "Wavelet based edge feature enhancement for convolutional neural networks," 2018.

[16] A. Singh and N. Kingsbury, "Efficient convolutional network learning using parametric log based dual-tree wavelet scatternet," in *Computer Vision Workshop (ICCVW), 2017 IEEE International Conference on*, pp. 1140–1147, IEEE, 2017.

[17] O. Rippel, J. Snoek, and R. P. Adams, "Spectral representations for convolutional neural networks," in *Advances in neural information processing systems*, pp. 2449–2457, 2015.

[18] T. Williams and R. Li, "Wavelet pooling for convolutional neural networks," in *International Conference on Learning Representations*, 2018.

[19] S. Fujieda, K. Takayama, and T. Hachisuka, "Wavelet convolutional neural networks for texture classification," *arXiv preprint arXiv:1707.07394*, 2017.

[20] 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)*, pp. 191–198, June 2018.

[21] S. Luan, C. Chen, B. Zhang, J. Han, and J. Liu, "Gabor convolutional networks," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4357–4366, 2018.

[22] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Harmonic networks: Deep translation and rotation equivariance," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pp. 7168–7177, IEEE, 2017.

[23] J.-H. Jacobsen, J. van Gemert, Z. Lou, and A. W. Smeulders, "Structured receptive fields in cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2610–2619, 2016.

[24] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Speeding up convolutional neural networks with low rank expansions," in *Proceedings of the British Machine Vision Conference*, BMVA Press, 2014.

[25] W. Chen, J. Wilson, S. Tyree, K. Q. Weinberger, and Y. 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, (New York, NY, USA), pp. 1475–1484, ACM, 2016.

[26] Y. Wang, C. Xu, S. You, D. Tao, and C. Xu, "Cnnpack: Packing convolutional neural networks in the frequency domain," in *Advances in Neural Information Processing Systems 29* (D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, eds.), pp. 253–261, Curran Associates, Inc., 2016.

[27] T. D. Tran, J. Liang, and C. Tu, "Lapped transform via time-domain pre- and post-filtering," *IEEE Transactions on Signal Processing*, vol. 51, pp. 1557–1571, June 2003.

[28] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proceedings of the British Machine Vision Conference (BMVC)* (E. R. H. Richard C. Wilson and W. A. P. Smith, eds.), pp. 87.1–87.12, BMVA Press, September 2016.

[29] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223, 2011.

[30] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.