AN EFFICIENT DEEP CONVOLUTIONAL LAPLACIAN PYRAMID ARCHITECTURE FOR CS RECONSTRUCTION AT LOW SAMPLING RATIOS

Wenxue Cui[1], *Heyao Xu*[1], *Xinwei Gao*[1,2], *Shengping Zhang*[1], *Feng Jiang*[1], *Debin Zhao*[1]

1.Department of Computer Science, Harbin Institute of Technology, Harbin, China
2.Wechat Business Group, Tencent, Shenzhen, China
wenxuecui@stu.hit.edu.cn, xuheyao0129@gmail.com, {s.zhang, fjiang, dbzhao}@hit.edu.cn
vitogao@tencent.com

ABSTRACT

The compressed sensing (CS) has been successfully applied to image compression in the past few years as most image signals are sparse in a certain domain. Several CS reconstruction models have been proposed and obtained superior performance. However, these methods suffer from blocking artifacts or ringing effects at low sampling ratios in most cases. To address this problem, we propose a deep convolutional Laplacian Pyramid Compressed Sensing Network (LapCSNet) for CS, which consists of a sampling sub-network and a reconstruction sub-network. In the sampling sub-network, we utilize a convolutional layer to mimic the sampling operator. In contrast to the fixed sampling matrices used in traditional CS methods, the filters used in our convolutional layer are jointly optimized with the reconstruction sub-network. In the reconstruction sub-network, two branches are designed to reconstruct multi-scale residual images and muti-scale target images progressively using a Laplacian pyramid architecture. The proposed LapCSNet not only integrates multi-scale information to achieve better performance but also reduces computational cost dramatically. Experimental results on benchmark datasets demonstrate that the proposed method is capable of reconstructing more details and sharper edges against the state-of-the-arts methods.

Index Terms— Compressed sensing, deep networks, image compression, laplacian pyramid, residual learning

1. INTRODUCTION

The compressed sensing theory [1, 2] shows that if a signal is sparse in a certain domain Ψ, it can be accurately recovered from a small number of random linear measurements less than that of Nyquist sampling theorem. Mathematically, the measurements are obtained by the following linear transformation

$$y = \Phi x + e \tag{1}$$

where $x \in R^N$ is the signal, $y \in R^M$ is known as the measurement vector, $\Phi \in R^{M \times N}$ is the measurement matrix and e denotes noise. If $M \ll N$, reconstructing x from y is generally ill-posed, which is one of the most challenging issues in compressed sensing.

To design an efficient CS reconstruction algorithm, many methods have been proposed, which can be generally divided into two categories: traditional optimization-based methods and recent DNN-based methods.

In the optimization-based reconstruction methods, given the linear projections y, the original image x can be reconstructed by solving the following convex optimization problem [1, 2]:

$$\tilde{x} = \underset{x}{\operatorname{argmin}} \frac{1}{2} \| \Phi x - y \|_2^2 + \lambda \| \Psi x \|_1 \tag{2}$$

To solve this convex problem, many algorithms have been proposed [4–6]. However, these algorithms suffer from uncertain reconstruction qualities and high computation cost, which inevitably limit their applications in practice.

Recently, some DNN-based algorithms have been proposed for image CS reconstruction. In [7], Mousavi et al. propose to utilize a stacked denoising autoencoder (SDA) to reconstruct original images from their measurements. A series of convolutional layers are adopted in [3, 8, 9] for image reconstruction. Despite their impressive results, massive block artifacts and ringing effects are delivered at low sampling ratios.

To overcome the shortcomings of the aforementioned methods, we propose a Laplacian Pyramid based deep architecture for CS reconstruction. Our network contains two sub-networks: sampling sub-network and reconstruction sub-network. In the sampling sub-network, a convolutional layer with the kernel size of $B \times B$ (B is the size of current block) is utilized to mimic the sampling process. In the reconstruction sub-network, we use two branches to reconstruct multi-scale residual features and multi-scale target images progressively through Laplacian pyramid architectures. Besides, the second branch integrates multi-scale information from the first branch to preserve finer textures. The sampling sub-network and reconstruction sub-network are optimized jointly with the robust Charbonnier loss [10].

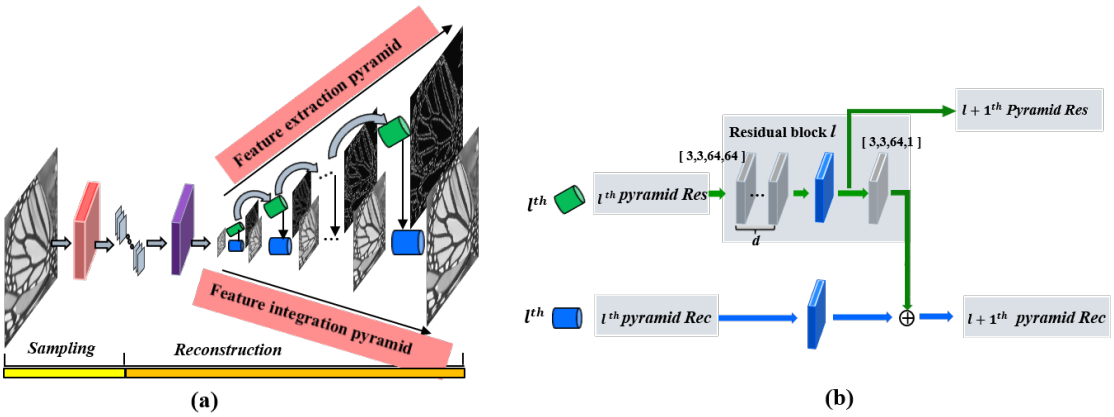

Fig. 1. (a) is the overview of the proposed LapCSNet and (b) shows the detailed structure in each level. The red box indicates convolutional layer for sampling operator. The sequence of squares indicate measurements. The purple box represents the "reshape+concat" layer [3] for initial reconstruction. The gray and blue boxes denote convolutional layers and transposed convolutional layers respectively and the four tuples in the bracket indicate the dimensions of parameters for adjacent convolutional layers.

2. PROPOSED METHOD

In this section, we describe the methodology of the proposed LapCSNet including the sampling sub-network and the reconstruction sub-network as well as the loss function.

2.1. Sampling Sub-network

In traditional block-based compressed sensing (BCS), each row of the sampling matrix Φ can be considered as a filter. Therefore, the sampling process can be mimicked using a convolutional layer [3, 9]. In our model, we use a convolutional layer with $B \times B$ filters and set stride as B for non-overlapping blocks. Specifically, given an image with size $w \times h$, there are a total of $L = \lfloor \frac{w}{B} \rfloor \times \lfloor \frac{h}{B} \rfloor$ non-overlapping blocks with size $B \times B$ ($B = 32$). The dimensions of measurements for each block is $n_B = \lfloor \frac{M}{N} B^2 \rfloor$, Therefore, the dimensions of measurements for the current image is $L \times n_B$. Traditional sampling matrices are fixed for various reconstruction algorithms. The proposed DNN-based sampling matrices are learned jointly with the reconstruction sub-network from large amounts of data.

2.2. Reconstruction Sub-network

For the CS reconstruction, several DNN-based models have been proposed [8, 9]. These methods are implemented for each block, which ignore the relationship between blocks and therefore results in serious blocking artifacts in most cases. To solve this problem, we adopt a "reshape+concat" layer [3] to concatenate all blocks to obtain initial reconstruction, which is then refined to obtain superior reconstruction.

Initial reconstruction Given the compressed measurements, the initial reconstruction block $\tilde{x}_{(j,S)}$ can be obtained by

$$\tilde{x}_{(j,S)} = \tilde{\Phi}_{(B,S)} y_j \tag{3}$$

where y_j is the measurement of the current j^{th} block and S is the scale factor for current block. $\tilde{\Phi}_{(B,S)}$ is a $(\frac{B}{S})^2 \times n_B$ matrix. In the aforementioned methods without scaling ($S = 1$), is difficult to be accurate calculated. By introducing the pyramid structure, a small block is first reconstructed in our method. Let $\mathcal{Q} = \{2^i | i = 1, 2, \dots, \}$ and $\mathcal{S} = \{s | s \in \mathcal{Q} \quad and \quad \frac{1}{2^s} > \frac{M}{N}\}$. The optimal scale factor S is obtained as

$$S^* = Max(\mathcal{S}) \tag{4}$$

where $Max(\cdot)$ is used to return the maximal element of a set. Therefore, $S > 1$ in our model. The convolution output of an image block in the sampling sub-network is a $n_B \times 1$ vector, so the size of the convolution filter in the initial reconstruction layer is $1 \times 1 \times n_B$. We use 1×1 stride convolution to reconstruct each block. Since a smaller version of the target block is reconstructed, $(\frac{B}{S})^2$ convolution filters of size $1 \times 1 \times n_B$ are used. However, the reconstructed outputs of each block is still a vector. To obtain the initial reconstructed image, a "reshape+concat" layer is adopted. This layer first reshapes each $(\frac{B}{S})^2$ reconstructed vector into a $\frac{B}{S} \times \frac{B}{S}$ block, then concatenates all the blocks to get the reconstructed image. From Eq. (4), we can get that when $\frac{M}{N} > 0.25$, we can not obtain the smaller version of the target image. In this case, we only use this model for CS reconstruction at low sampling ratios.

Further reconstruction CSNet [3] has only 5 layers for CS reconstruction, which results in poor performance at low sampling ratios. Deep networks and elaborated architectures are essential for accurate reconstruction while increasing

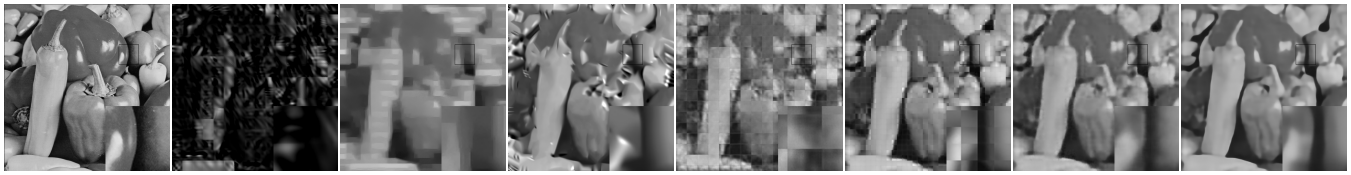

Fig. 2. Visual quality comparison of image CS recovery on image *Pepper* from Set14 in the case of sampling ratio = 0.01

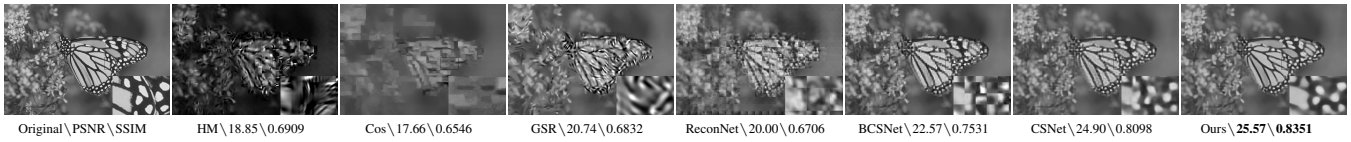

Fig. 3. Visual quality comparison of image CS recovery on image *Monarch* from Set14 in the case of sampling ratio = 0.02

computational complexity due to the unenlightened superposition of convolutional layers. Our further reconstruction network takes the output of initial reconstruction as input and progressively predicts residual images at a total of $log_2 S$ levels where S is the scale factor from Eq. 4. Moreover, our reconstruction model has two branches: residual feature extraction [11] and feature integration.

(1) Multi-scale residual feature extraction: There is a one-to-one correspondence between reconstruction levels and residual blocks (shown in Fig. 1). In level l, the corresponding residual block consists of d convolutional layers and one transposed convolutional layer [12] to upsample the extracted features by a scale of 2. The output of each transposed convolutional layer is connected to two different layers: (i) a convolutional layer for reconstructing a residual image for level l; (ii) a convolutional layer for extracting features in next level $l+1$. Note that the residual feature at lower levels are shared with higher levels and thus can increase the non-linearity of the network to reconstruct the target image. Moreover, the hierarchical architecture is used for the CS feature extraction, which can preserve more details.

(2) Multi-scale feature integration: In level l, the input image is up-sampled by a scale of 2 using a transposed convolutional layer [12]. The upsampled image is then combined with the predicted residual images from the residual feature extraction branch. Then, the output in this level l is fed into the next reconstruction module of level l+1. This integration architecture fuses the the multi-scale residual branch and corresponding reconstruction branch efficiently. In addition, most of convolution operation are executed on the smaller version of the target image, which indicates our model is more efficient than the existing models with the same depth.

2.3. Loss function

Let x be the input image, θ_S is the parameter of the sampling sub-network and θ_R the parameter of reconstruction sub-network. Two mapping function f_S and f_R are desired to produce accurate measurements and reconstructed image $\hat{y} = f(x; \theta_S, \theta_R)$. We denote the residual image in level l by r_l, the up-scaled image by x_l and the corresponding target image by $\hat{y}_l$. The desired output images in level l is modeled by $\hat{y}_l = x_l + r_l$. We use the bicubic downsampling method to resize the ground truth image y to y_l at each level. We propose to use the robust Charbonnier penalty function for each level. The overall loss function is defined as:

$$\ell(\hat{y}, y; \theta_S, \theta_R) = \frac{1}{N} \sum_{i=1}^{N} \sum_{l=1}^{L} \rho(\hat{y_l}^{(i)} - y_l^{(i)}) \tag{5}$$

where $\hat{y}_l^{(i)} = x_l^{(i)} + r_l^{(i)}$ and $\rho(x) = \sqrt{x^2 + \varepsilon^2}$. N is the number of training samples, and L is the number of levels in our pyramid. ε is empirically set to $1e-3$.

3. EXPERIMENTAL RESULTS AND ANALYSIS

3.1. Implementation and training details

In the reconstruction sub-network, each convolutional layer consists of 64 kernels with size of 3×3. We initialize the convolutional filters using the same method [15]. The size of the transposed convolutional filter is 4×4. A ReLU layer with a negative slope of 0.2 is subsequent for all convolutional and transposed convolutional layers. We pad zero values around the boundaries before applying convolution to keep the size of all feature maps the same as the input of each level.

We use the training set (200 images) and testing set (200 images) of the BSDS500 database [16] for training, and the validation set (100 images) of BSDS500 for validation. We set the patch size as 128×128, and batch size as 64. We augment the training data in three ways: (i) Randomly scale between $[0.75, 1.2]$. (ii) Rotate the images by $90°$, $180°$, and $270°$. (iii) Flip the images horizontally or vertically with a probability of 0.5. We train our model with the Matlab toolbox MatConvNet [17] on a Titan X GPU. The momentum parameter is set as 0.9 and weight decay as $1e-4$. The learning rate is initialized to $1e-6$ for all layers and decreased by a factor of 2 for every 50 epochs. We train our model for 200 epochs and each epoch iterates 1000 times.

Table 1. Quantitative evaluation of state-of-the-arts CS reconstruction algorithms: Average PSNR\SSIM\time\network Layers for sampling ratios 0.01, 0.02, 0.1 on dataset Set5. Red text indicates the best and blue the second best performance

Alg.	sampling ratio 0.01	sampling ratio 0.02	sampling ratio 0.1	Avg.
TV [5]	16.31\0.4101\22.48\–	17.94\0.4439\17.56\–	18.33\0.5921\6.93\–	17.53\0.4820\15.66\–
MH [13]	12.43\0.1999\84.32\–	20.62\0.5381\78.59\–	28.57\0.8211\69.27\–	20.54\0.5197\77.39\–
Cos [14]	17.42\0.4122\30487.56\–	18.46\0.4827\27453.49\–	29.55\0.8522\6433.25\–	21.81\0.5824\21454.65\–
GSR [6]	20.81\0.5128\494.34\–	22.78\0.5873\532.45\–	29.99\0.8654\412.85\–	24.53\0.6552\479.88\–
ReconNet [8]	18.07\0.4138\0.34\7	20.05\0.4927\0.41\7	24.58\0.6762\0.37**7**	20.90\0.5276\0.37\7
BCSNet [9]	22.07\0.5465**0.01**\3	23.40\0.6168**0.01**\3	30.01\0.8837**0.01**\3	25.16\0.6823**0.01**\3
CSNet [3]	24.04\0.6374**0.03**\6	25.87\0.7069**0.02**\6	32.30\0.9015\0.04\6	27.40\0.7486**0.03**\6
LapCSNet-2	**24.31****0.6537**\0.07**17**	**26.20****0.7397**\0.05**12**	**32.34****0.9023****0.03****7**	**27.62****0.7652**\0.05**12**
LapCSNet-4	**24.42****0.6686**\0.10**23**	**26.45****0.7520**\0.07**16**	**32.44****0.9047**\0.05**9**	**27.77****0.7751**\0.07**16**

Table 2. Quantitative evaluation of state-of-the-arts CS reconstruction algorithms: Average PSNR\SSIM\time\ network Layers for sampling ratios 0.01, 0.02, 0.1 on dataset Set14. Red text indicates the best and blue the second best performance

Alg.	sampling ratio 0.01	sampling ratio 0.02	sampling ratio 0.1	Avg.
TV [5]	15.17\0.3691\58.43\–	17.20\0.4069\47.36\–	17.96\0.5381\17.21\–	16.78\0.4380\41.00\–
MH [13]	12.26\0.1319\95.48\–	19.20\0.4923\89.37\–	26.38\0.7282\70.19\–	19.28\0.4508\85.01\–
Cos [14]	16.73\0.3533\23563.48\–	18.35\0.4074\23042.53\–	27.20\0.7433\16596.02\–	20.76\0.5013\21067.32\–
GSR [6]	19.41\0.4583\1654.39\–	20.89\0.4900\1486.10\–	27.50\0.7705\948.03\–	22.60\0.5729\1362.86\–
ReconNet [8]	18.09\0.3907\1.04\7	19.46\0.4507\1.12\7	22.91\0.5974\1.23**7**	20.15\0.4796\1.13\7
BCSNet [9]	20.94\0.4910**0.05**\3	22.00\0.5557**0.04**\3	27.33**0.8732****0.05**\3	23.42\0.6400**0.05**\3
CSNet [3]	22.78\0.5574**0.13**\6	24.33\0.6185**0.12**\6	28.91\0.8119\0.14\5	25.34\0.6626**0.13**\6
LapCSNet-2	**23.03****0.5688**\0.25**17**	**24.55****0.6324**\0.19**12**	**28.94**\0.8124**0.13****7**	**25.51****0.6712**\0.19**12**
LapCSNet-4	**23.16****0.5818**\0.39**23**	**24.76****0.6454**\0.25**16**	**29.00****0.8147**\0.17**9**	**25.64****0.6806**\0.24**16**

3.2. Comparison with the state-of-the-arts

We compare our algorithm with seven representative methods, i.e., total variation (TV) method [5], multi-hypothesis (MH) method [13], collaborative sparsity (Cos) method [14], group sparse representation (GSR) method [6], ReconNet [8], BCSNet [9] and CSNet [3]. In these algorithms, the first four belong to traditional optimization-based methods, while the last three are recent network-based methods. The PSNR and SSIM reconstruction performances at three different sampling ratios: 0.01, 0.02 and 0.1 for the datasets Set5 and Set14 are summarized in Table 1 and Table 2, respectively. The "LapCSNet-2" denotes $d = 2$ and "LapCSNet-4" $d = 4$. It can be seen from the tables that about 0.2-0.5dB PSNR improvement is obtained on both test datasets Set5 and Set14 when the sampling ratio is very low. Obviously, the proposed method outperform the existing algorithms in low-ratio by a large margin, which fully demonstrates the effectiveness of our model. Moreover, for the ratio=0.01, our model is about 4 times deeper than CSNet, while the running time is just about 3 times longer than that of the CSNet, which demonstrate the Laplacian pyramid is a high-efficiency design for CS reconstruction. The visual comparisons in the case of ratio=0.01 and ratio=0.02 in Fig. 2 and Fig. 3 show that the proposed LapCSNet is able to reconstruct more details and sharper edges without obvious blocking artifacts.

4. CONCLUSION

In this work, we propose a deep convolutional network (LapCSNet) within a Laplacian pyramid framework for fast and accurate CS reconstruction. Our model consists of two sub-networks namely sampling sub-network and reconstruction sub-network. In the sampling sub-network, the sampling filters are learned jointly with the reconstruction sub-network. In the reconstruction sub-network, we divide our model into two branches to extract the residual features and integrate target images using laplacian pyramid architectures, respectively. In other words, the reconstruction sub-network progressively predicts high-frequency residuals and integrate multi-scale information in a coarse-to-fine manner. The sampling sub-network and reconstruction sub-network are optimized jointly using a robust Charbonnier loss function. Experimental results show that the proposed LapCSNet is capable of reconstructing more details and sharper edges against several state-of-the-arts algorithms.

5. ACKNOWLEDGEMENTS

This work is partially funded by the Major State Basic Research Development Program of China (973 Program 2015CB351804) and the National Natural Science Foundation of China under Grant No. 61572155 and 61672188.

6. REFERENCES

[1] Emmanuel J Candès, Justin Romberg, and Terence Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006. 1

[2] David L Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006. 1

[3] Wuzhen Shi, Feng Jiang, Shengping Zhang, and Debin Zhao, "Deep networks for compressed image sensing," *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 877–882, 2017. 1, 2, 4

[4] Scott Shaobing Chen, David L Donoho, and Michael A Saunders, "Atomic decomposition by basis pursuit," *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001. 1

[5] Chengbo Li, Wotao Yin, Hong Jiang, and Yin Zhang, "An efficient augmented lagrangian method with applications to total variation minimization," *Computational Optimization and Applications*, vol. 56, no. 3, pp. 507–530, 2013. 1, 4

[6] Jian Zhang, Debin Zhao, and Wen Gao, "Group-based sparse representation for image restoration," *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3336–3351, 2014. 1, 4

[7] Ali Mousavi, Ankit B Patel, and Richard G Baraniuk, "A deep learning approach to structured signal recovery," *IEEE Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1336–1343, 2015. 1

[8] Kuldeep Kulkarni, Suhas Lohit, Pavan Turaga, Ronan Kerviche, and Amit Ashok, "Reconnet: Non-iterative reconstruction of images from compressively sensed random measurements," *IEEE Conference on Computer Vision and Pattern Recongnition (CVPR)*, pp. 449–458, 2016. 1, 2, 4

[9] Amir Adler, David Boublil, Michael Elad, and Michael Zibulevsky, "A deep learning approach to block-based compressed sensing of images," *arXiv preprint arXiv:1606.01519*, 2016. 1, 2, 4

[10] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr, "Lucas/kanade meets horn/schunck: Combining local and global optic flow methods," *Springer International Journal of Computer Vision*, vol. 61, no. 3, pp. 211–231, 2005. 1

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. 3

[12] Vincent Dumoulin and Francesco Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint arXiv:1603.07285*, 2016. 3

[13] Chen Chen, Eric W Tramel, and James E Fowler, "Compressed-sensing recovery of images and video using multihypothesis predictions," *IEEE Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pp. 1193–1198, 2011. 4

[14] Jian Zhang, Debin Zhao, Chen Zhao, Ruiqin Xiong, Siwei Ma, and Wen Gao, "Compressed sensing recovery via collaborative sparsity," *IEEE Data Compression Conference (DCC)*, pp. 287–296, 2012. 4

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *IEEE International Conference on Computer Vision*, pp. 1026–1034, 2015. 3

[16] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2011. 3

[17] Andrea Vedaldi and Karel Lenc, "Matconvnet: Convolutional neural networks for matlab," *ACM International Conference on Multimedia*, pp. 689–692, 2015. 3