This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2018.2879184, IEEE Signal Processing Letters

PhaseNet: A Deep Convolutional Neural Network for 2D Phase Unwrapping

Spoorthi G.E., Subrahmanyam Gorthi, and Rama Krishna Sai Gorthi

***Abstract*—Phase unwrapping is a crucial signal processing problem in several applications that aims to restore original phase from the wrapped phase. In this letter, we propose a novel framework for unwrapping the phase using deep fully convolutional neural network termed as PhaseNet. We reformulate the problem definition of directly obtaining continuous original phase as obtaining the wrap-count (integer jump of 2π) at each pixel by semantic segmentation, and this is accomplished through a suitable deep learning framework. The proposed architecture consists of an encoder network, a corresponding decoder network followed by a pixel-wise classification layer. The relationship between the absolute phase and the wrap-count is leveraged in generating abundant simulated data of several random shapes. This deliberates the network on learning continuity in wrapped phase maps rather than specific patterns in the training data. We compare the proposed framework with the widely adapted quality-guided phase unwrapping algorithm and also with the well known MATLAB's unwrap function for varying noise levels. The proposed framework is found to be robust to noise and computationally fast. The results obtained highlight that Deep Convolutional Neural Network (DCNN) can indeed be effectively applied for phase unwrapping, and the proposed framework will hopefully pave the way for the development of a new set of deep learning based phase unwrapping methods.**

***Index Terms*—Phase Unwrapping, Deep Convolutional Neural Network, Encoder, Decoder, Semantic Segmentation.**

I. Introduction

PHASE UNWRAPPING is a classic signal processing problem that refers to recovering the originial phase value from the principal value (-π π]. Two dimensional phase unwrapping problem arises in many applications such as optical measurement techniques(e.g., digital holographic interferometry and fringe projection profilometry [1]), Synthetic Aperture Radar (SAR) [2] and Magnetic Resonance Imaging (MRI) [3]. Phase estimated from these applications is directly proportional to the physical parameter under consideration such as object shape, terrain elevation and magnetic field inhomogeneity.

Ideally phase unwrapping could be accomplished by addition or subtraction of 2π at each pixel depending on the phase difference between the neighboring pixels. However, in practice, phase unwrapping is a very challenging problem because of the presence of severe noise, rapidly varying phase changes and phase discontinuities.

Many phase unwrapping algorithms have been proposed over the years and they can be broadly classified into two categories: path-following approaches and minimum norm approaches. Most of the path-following algorithms perform phase integration along the path chosen to recover the true phase. There are four kinds of path-following algorithms: 1) quality-guided algorithm [4], [5] ; 2) branch cut algorithm [6]; 3) mask cut algorithm [7]; 4) minimum discontinuity algorithm [8], [9]. Generally these algorithms are computationally efficient but are not robust to severe noise as the error present at a point or local region may propagate along the path chosen. Minimum-norm methods [10], [11] minimize the difference between the local derivative of the true phase and that of the wrapped phase to carry out phase unwrapping. Minimum-norm methods are robust to noise but produce over smooth phase and are computationally intensive thus making them unsuitable for real-time measurements. Schwartzkopf et. al., [12] proposed a feed-forward multilayer perceptron neural network that detects discontinuities based on the computation of probabilities over a local patch. The main limitation of that method is that it's output is path dependent.

Deep learning methods have been extensively used in object detection and image classification e.g., [13], [14] and have been achieving state-of-the-art performance. DCNNs have also been applied in various image processing applications such as image super resolution [15], medical image segmentation [16], depth predication in stereo and monocular images [17]. Further, there is a mounting evidence that DCNNs are setting new records in many of these applications. Despite of all the successes in wide variety of vision applications, to our knowledge there is no framework that successfully uses DCNN to unwrap the phase.

In this letter, we try to abridge this gap by proposing a deep learning based phase unwrapping framework that uses DCNN, and it is referred to as "PhaseNet". In the proposed framework, phase unwrapping is formulated as a semantic segmentation problem. Semantic segmentation, which is also referred to as "dense predictions" or "pixel-wise classification," aims at classifying each pixel into one of the predetermined classes depending on the class of its enclosing object or region [18], [19].

The proposed PhaseNet takes wrapped phase as input and wrap-count (i.e., integer multiple of 2π to be added at each pixel of wrapped phase to restore original phase) as semantic label. Furthermore, a clustering-based post-processing is also proposed that explicitly incorporates the information of the locations of $\pm 2\pi$ discontinuities. It is found that the proposed PhaseNet is efficient when compared to the conventional phase unwrapping methods in terms of robustness to noise and computational time.

The authors are with the Department of Electrical Engineering, Indian Institute of Technology, Tirupati, A.P, 517506, India (E-mail: spoorthi.ge92@gmail.com; s.gorthi@iittp.ac.in; rkg@iittp.ac.in).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2018.2879184, IEEE Signal Processing Letters

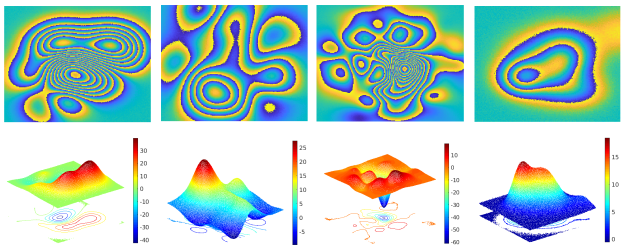


Fig. 1: Sample training data that is generated by repeatedly performing arithmetic operations on Gaussian functions with randomly varying mean and variance values.

II. Proposed Method

Phase unwrapping can also be interpreted as determining the unknown integral multiple of 2π to be added at each pixel of the wrapped phase map to restore the true phase. The true phase Φ can be estimated from wrapped phase Ψ as

$$\Phi_{(x,y)} = \Psi_{(x,y)} + 2\pi\, k_{(x,y)}, \tag{1}$$

where (x, y) denotes the spatial coordinates of a pixel and k denotes integer multiple of 2π referred to as 'wrap-count' to be added to wrapped phase to get the absolute phase.

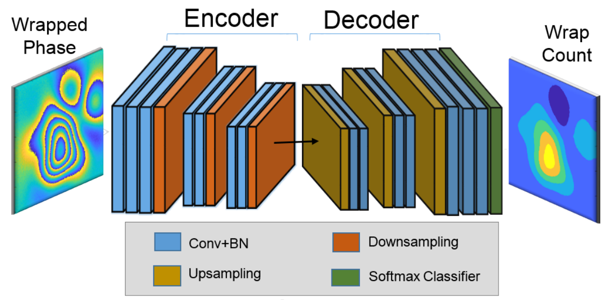
In the proposed framework, instead of training the architecture for directly obtaining true phase from wrapped phase as a regression problem, we effectively convert this into semantic segmentation problem through Eq. 1. The phase unwrapping is learned through framework that takes wrapped phase $\Psi_{(x,y)}$ as input and gives the output as wrap-count $k_{(x,y)}$. The ground truth for the DCNN to obtain wrap-count $k_{(x,y)}$ can be computed using the following equation:

$$k_{(x,y)} = \text{round}\left(\frac{\Phi_{(x,y)} - \Psi_{(x,y)}}{2\pi}\right) \tag{2}$$

Phase unwrapping is a severely ill-posed problem as two regions having same shape, size and position can still belong to different classes depending on the class of neighborhood region or pixel. Hence, understanding the relationship between different classes is the foremost requirement as contrary to most of the deep learning applications that are trained to learn conformity in features or a specific pattern.

Although fully connected neural networks have demonstrated superior results in object recognition and image classification, they are not generally used for semantic segmenation as they could potentially lose the spatial information. On the other hand, Fully Convolutional Neural (FCN) networks [18], [19], [20] can recognize the spatial relationships between different classes. Furthermore, FCNs can also take arbitrary sized input and produce corresponding sized output with efficient interface and learning. Hence, we considered convolutional encoder-decoder architecture that is fully convolutional as that correlates with our problem definition.

The proposed architecture consists of encoder network and corresponding decoder network. Similar to SegNet [19], upsampling layer of decoder network uses max-pooling indices obtained from the corresponding max-pooling layer of the encoder network to upsample the low resolution feature maps.



Conv-Convolution, BN-Batch Normalization

Fig. 2: An illustration of the Convolutional Encoder-Decoder architecture. Wrapped phase is given as input, and wrap-count at each pixel is given as ground truth.

The two key advantages of reusing max-pooling indices in the decoder network are: (i) it improves preciseness of boundaries (ii) it reduces the number of parameters enabling end-to-end training.

A. Generation of Data

One of the compelling requisites for deep learning techniques is to have large labeled dataset for training. In phase unwrapping, unlike in many other applications, there is a definitive input-output relationship between the wrapped phase and the wrap-count, as shown in Eq. 2. We exploit this relationship for generating abundant simulated data containing several random shapes.

More specifically, training data for phase unwrapping is generated by repeatedly performing arithmetic operations such as addition and subtraction on Gaussian functions with randomly varying mean and variance values. This enables the network to learn phase continuities for any general shapes rather than limiting it to certain definitive patterns. Furthermore, Gaussian noise was added to generated data to make the training dataset more practical. Fig. 1 gives a glimpse of training data that is used in training.

B. Model Architecture

The proposed architecture is illustrated in Fig. 2 and the details of the network configuration are shown in Table I. It consists of an encoder network which has three max-pooling layers interleaved between seven convolutional layers. Corresponding to each encoder layer, there is a decoder layer that semantically projects the low resolution features learnt by the encoder onto the pixel space. Upsampling layer in the decoder network upsamples its input feature maps from the indices received from the corresponding max-pooling layer of encoder.

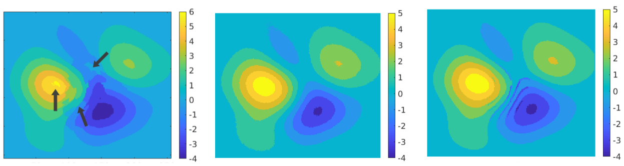
After comprehensive experimentaion, it is found that a kernel size of 5×5 provides optimal receptive field along with channel size of 128. Hence these values are used throughout the network. Regularization is also incoroprtead in the network to prevent over-fitting by introducing dropout layers after max-pooling layers$_{2,3}$ and before upsampling layers$_{2,1}$. Convolutional layers of encoder are followed by element-wise rectified linear nonlinearity operation (ReLU) and Batch

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2018.2879184, IEEE Signal Processing Letters

TABLE I: Proposed network configuration for PhaseNet.

Layer	#Filters	Size	Output Size
Conv_1+ReLU	128	5 ×5	256×256 × 128
$\text{Conv}_{2,3}$+ReLU	128	5 ×5	256×256 × 128
Max-pooling_1		2×2	128×128 × 128
$\text{Conv}_{4,5}$+ReLU	128	5 ×5	128×128 × 128
Max-pooling_2		2×2	64×64 × 128
$\text{Conv}_{6,7}$+ReLU	128	5 ×5	64×64 × 128
Max-pooling_3		2×2	32×32 × 128
Upsampling_1		2×2	64×64 × 128
$\text{Conv}_{8,9}$	128	5 ×5	64×64 × 128
Upsampling_2		2×2	128×128 × 128
$\text{Conv}_{10,11}$	128	5 ×5	128×128 × 128
Upsampling_3		2×2	256×256 × 128
$\text{Conv}_{12,13}$	128	5 ×5	256×256 × 128
Conv_{14}	N	1 ×1	256×256 × N

Conv-Convolution, N-Number of classes



(a) Output of DCNN (b) Ground truth (c) Output after post-processing

Fig. 3: Output of DCNN shown in Fig. (a) is further improved by clustering-based post-processing. Arrows in Fig. (a) indicate the misclassified regions that got corrected through the proposed post-processing.

Normalization(BN) [21]. Notice that no nonlinearity is present in the decoder network. Max–Pooling layers have window size of 2×2 and stride of two. Dimension of feature maps at the final decoder layer is reduced by convolving with $1 \times 1 \times 128 \times N$ trainable filters where N is the number of classes. The resultant of this is fed to softmax classifier that classifies each pixel independently.

C. Post-processing: Clustering-based Smoothness

Fig. 3(a) shows the output of DCNN, and Fig. 3(b) shows the corresponding ground truth. It can be noted from the regions indicated with arrow marks in Fig. 3(a) that (i) the pixels around the closely disconnected regions, and (ii) the pixels around suddenly varying phase region are misclassified. In order to further improve these results, we propose the following clustering-based approach for enforcing smoothness by incorporating complementary information.

The wrapped phase is convoluted with isotropic the Laplacian filter [22] to obtain residual pixels. These residual pixels are excluded from the wrapped phase, and disconnected clusters are formed by 8-connected neighborhood [22]. Disjoined clusters are binarized and the whole cluster is assigned an unique wrap-count by obtaining the mode of the wrap-count from the output of DCNN in that particular region. Output of the DCNN is retained across residual pixel region. Fig. 3(c) shows the wrap-count at each pixel after performing the aforementioned post-processing. Since the wrap-counts at residual pixels locations are preserved, there is still an undesirable classification along the contours of clusters. This is eliminated by passing it through median filter after multiplying wrap-counts with 2π and adding that with wrapped phase. Fig. 4 presents a block diagram of the entire framework of PhaseNet by integrating all the steps.

TABLE II: Mean Square Error (MSE) & processing time for the PhaseNet, QGPU and MATLAB's unwrap function.

Method	MSE for SNR = 0 dB	Time in Seconds
PhaseNet	2	0.18
Quality-guided Unwrap	11	24
MATLAB's Unwrap	17	0.05

III. Experimental Results

Training dataset consists of 10000 training samples, and the size of each sample is $256 \times 256 \times 1$. Wrapped phase is given as input to the DCNN (minimum value is set to zero by shifting the values). The wrap-counts in the training dataset are varying in the range of -15 to 15 (i.e., -90 to 90 radians) constituting a 31 class problem. Due to imbalance of class labels in training dataset, the PhaseNet is able to accurately unwrap phase values ranging up to -36 to 36 radians. The weights of all the layers are initialized from scratch by the initialization described in [23]. Cross-entropy loss and Adam optimizer [24] with momentum 0.9 and initial learning rate of 0.0001 is used. It is found that small learning rate is necessary to ensure that the model converges smoothly. Dropout probability is set to 0.25. The network converges after approximately 100K iterations. Training takes about 10 hours on NIVIDA GTX 1080-Ti GPU with 11 GB memory.

To evaluate the robustness of the PhaseNet for varying noise, *peaks* function of size 256×256 with various noise levels are simulated in MATLAB 2018a. The performance of the method is compared with the well known quality-guided phase unwrapping (QGPU) method [25] and MATLAB's unwrap function. Fig. 5(a) and 5(b) respectively show the wrapped phase at SNR = 0 dB and the corresponding ground truth for unwrapped phase. Fig. 5(c) shows the phase estimated from the PhaseNet. Fig. 5(d), 5(e) and 5(f) show the error plots for the proposed PhaseNet, QGPU and MATLAB's unwrap respectively. It can be noted from the error plot of QGPU that half of the reference plane is wrongly displaced because of high noise level. Moreover, it is found from our experiments that the performance of QGPU is unpredictable at low SNR levels. The error plot of MATLAB's unwrap function has obvious error propagation along the path leading to inferior results. On the other hand, PhaseNet performed well compared to the other methods and it is found to be very robust to noise.

Fig. 6 shows error analysis for all the three methods for varying SNR values. It can be noted from these results that at high SNR values, all the three methods are performing equally well. On the other hand, at low SNR values, the proposed PhaseNet clearly outperforms the other two methods. More specifically, unlike the other two methods, the proposed approach almost reliably reconstructs the phase map even up to a noise level of -2 dB. Table II presents the Mean Square Error (MSE) and processing time values for all the three methods for

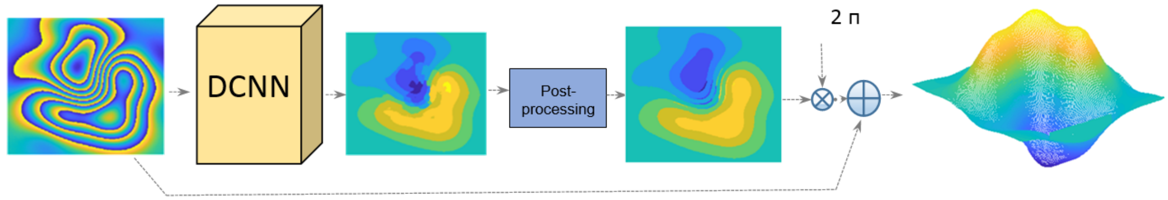
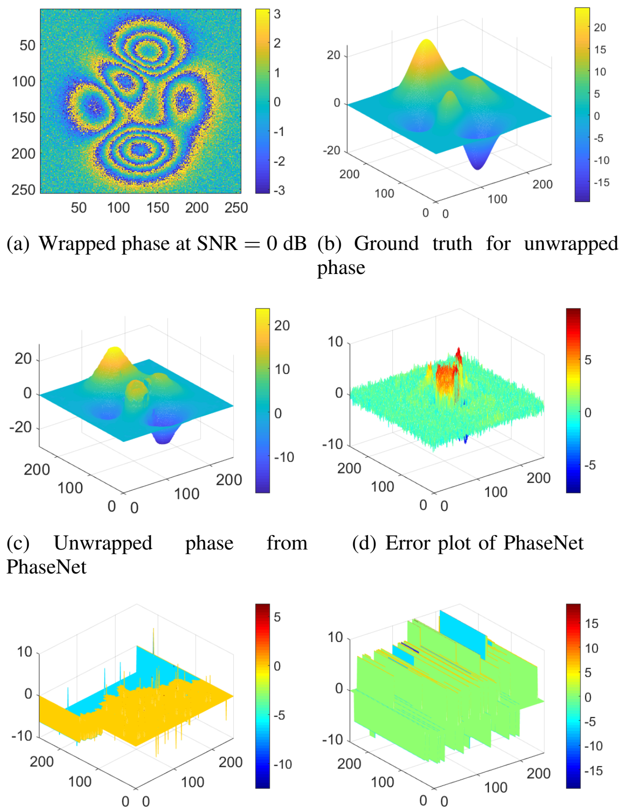


Fig. 4: Illustration of the proposed framework. Wrapped phase is given as input to the DCNN. Output of the DCNN with clustering-based post-processing gives the wrap-count. It is multiplied with 2π and resultant is added with wrapped phase to retrieve the true phase.



(a) Wrapped phase at SNR = 0 dB (b) Ground truth for unwrapped phase

(c) Unwrapped phase from PhaseNet (d) Error plot of PhaseNet

(e) Error plot of QGPU (f) Error plot of MATLAB's unwrap

Fig. 5: Simulation results of PhaseNet, QGPU and MATLAB's unwrap function at SNR = 0 dB.

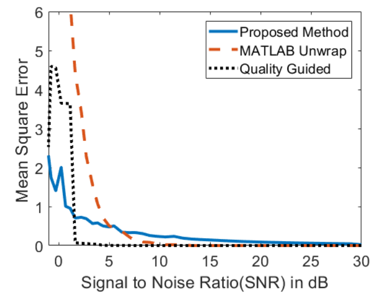
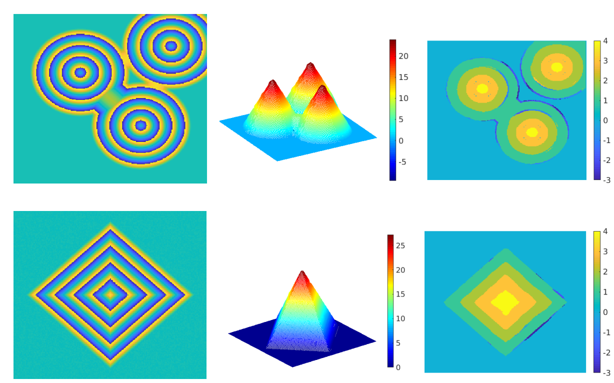


Fig. 6: Error observed using the PhaseNet, QGPU and MATLAB's unwrap as a function of the SNR.



(a) Wrapped phase (b) Unwrapped phase (c) Wrap-count prediction of PhaseNet

Fig. 7: Qualitative assessment of the adaptability of the PhaseNet for unwrapping regular shapes.

the unwrapping of the *peaks* function at a SNR value of 0 dB. It can be noted that the computational time of the proposed method is comparable to MATLAB's unwrap method, and has the least MSE value among all the three methods.

As described in section II-A, the PhaseNet has been trained with randomly varying shapes that are generated by combining Gaussians with varying mean and variance values. The adaptability of the PhaseNet to different regular shapes like cones and pyramids is also evaluated, and the results are shown in Fig. 7. It can be be noted that the proposed framework is able to predict the wrap-counts even for regular shapes and produce accurate results.

IV. Conclusions

We presented a novel Deep Convolutional Neural Network (DCNN) based framework for phase unwrapping, and is referred to as PhaseNet. The classical phase unwrapping problem is reformulated as a semantic segmentation problem, and a fully convolutional architecture is proposed for solving this problem. The training data generation procedure in this paper is quite general, and in future, it can be tailor-made for customizing it to application-specific challenges like rapidly varying phase and discontinuities. The results from the PhaseNet are compared with two other widely used methods, and it is found that the proposed framework achieves good performance even under severe noise conditions with less computational time.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2018.2879184, IEEE Signal Processing Letters

REFERENCES

[1] S. Xing and H. Guo, "Temporal phase unwrapping for fringe projection profilometry aided by recursion of chebyshev polynomials," *Applied optics*, vol. 56, no. 6, pp. 1591–1602, 2017.

[2] Y. Gao, S. Zhang, T. Li, Q. Chen, S. Li, and P. Meng, "Adaptive unscented kalman filter phase unwrapping method and its application on Gaofen-3 interferometric SAR data." *Sensors*, vol. 18, no. 6, 2018.

[3] J. Dong, F. Chen, D. Zhou, T. Liu, Z. Yu, and Y. Wang, "Phase unwrapping with graph cuts optimization and dual decomposition acceleration for 3D high-resolution MRI data," *Magnetic Resonance in Medicine*, vol. 77, no. 3, pp. 1353–1358, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/mrm.26174

[4] Y. Zhang, S. Wang, G. Ji, and Z. Dong, "An improved quality guided phase unwrapping method and its applications to MRI," *Progress In Electromagnetics Research*, vol. 145, pp. 273–286, 2014.

[5] H. Zhong, J. Tang, S. Zhang, and M. Chen, "An improved quality-guided phase-unwrapping algorithm based on priority queue," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 2, pp. 364–368, 2011.

[6] R. M. Goldstein, H. A. Zebker, and C. L. Werner, "Satellite radar interferometry: Two-dimensional phase unwrapping," *Radio science*, vol. 23, no. 4, pp. 713–720, 1988.

[7] D. Gao and F. Yin, "Mask cut optimization in two-dimensional phase unwrapping," *IEEE Geoscience and Remote Sensing Letters*, vol. 9, no. 3, pp. 338–342, 2012.

[8] J. Xu, D. An, X. Huang, and P. Yi, "An efficient minimum-discontinuity phase-unwrapping method," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 5, pp. 666–670, 2016.

[9] Y. Liu, Y. Han, F. Li, and Q. Zhang, "Speedup of minimum discontinuity phase unwrapping algorithm with a reference phase distribution," *Optics Communications*, vol. 417, pp. 97–102, 2018.

[10] D. C. Ghiglia and L. A. Romero, "Minimum Lp-norm two-dimensional phase unwrapping," *J. Opt. Soc. Am. A*, vol. 13, no. 10, pp. 1999–2013, Oct 1996. [Online]. Available: http://josaa.osa.org/abstract.cfm?URI=josaa-13-10-1999

[11] J. Strand, T. Taxt, and A. K. Jain, "Two-dimensional phase unwrapping using a block least-squares method," *IEEE Transactions on Image Processing*, vol. 8, no. 3, pp. 375–386, 1999.

[12] W. Schwartzkopf, T. E. Milner, J. Ghosh, B. L. Evans, and A. C. Bovik, "Two-dimensional phase unwrapping using neural networks," in *Image*

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in

Analysis and Interpretation, 2000. Proceedings. 4th IEEE Southwest Symposium. IEEE, 2000, pp. 274–277.

[13] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks." in *CVPR*, vol. 1, no. 2, 2017, p. 3.

[14] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *International conference on machine learning*, 2014, pp. 647–655.

[15] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[17] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2650–2658.

[18] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[19] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv:1511.00561*, 2015.

[20] G. Lin, A. Milan, C. Shen, and I. D. Reid, "RefineNet: Multi-path Refinement Networks for High-Resolution Semantic Segmentation." in *CVPR*, vol. 1, no. 2, 2017, p. 5.

[21] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[22] R. C. Gonzalez, R. E. Woods, S. L. Eddins *et al.*, *Digital image processing using MATLAB*. Pearson-Prentice-Hall Upper Saddle River, New Jersey, 2004, vol. 624.

Proceedings of the IEEE international conference on computer vision, 2015, pp. 1026–1034.

[24] D. Kingma and J. A. Ba, "A method for stochastic optimization. arxiv 2014," *arXiv preprint arXiv:1412.6980*.

[25] D. C. Ghiglia and M. D. Pritt, *Two-dimensional phase unwrapping: theory, algorithms, and software*. Wiley New York, 1998, vol. 4.