A JOINT CONVOLUTIONAL AND SPATIAL QUAD-DIRECTIONAL LSTM NETWORK FOR PHASE UNWRAPPING

Malsha V. Perera *, *Ashwin De Silva* *

Dept. of Electronic and Telecommunication Eng., Univeristy of Moratuwa, Sri Lanka

ABSTRACT

Phase unwrapping is a classical ill-posed problem which aims to recover the true phase from wrapped phase. In this paper, we introduce a novel Convolutional Neural Network (CNN) that incorporates a Spatial Quad-Directional Long Short Term Memory (SQD-LSTM) for phase unwrapping, by formulating it as a regression problem. Incorporating SQD-LSTM can circumvent the typical CNNs' inherent difficulty of learning global spatial dependencies which are vital when recovering the true phase. Furthermore, we employ a problem specific composite loss function to train this network. The proposed network is found to be performing better than the existing methods under severe noise conditions (Normalized Root Mean Square Error of 1.3% at SNR $= 0$ dB) while spending a significantly less computational time (0.054s). The network also does not require a large scale dataset during training, thus making it ideal for applications with limited data that require fast and accurate phase unwrapping.

Index Terms— Phase Unwrapping, Spatial Quad - Directional LSTM, Convolutional Neural Networks

1. INTRODUCTION

The problem of phase unwrapping is prevalent in many applications such as Quantitative Susceptibility Mapping (QSM) in Magnetic Resonance Imaging (MRI) [1], Synthetic Aperture Radar (SAR) interferometry [2], Fringe Projection Techniques (FPT) [3] and digital holographic interferometry [4]. Its objective is to recover the true phase signal from an observed wrapped phase signal that is in the range of $(-\pi, \pi]$. While it may be easy to recover true phase from wrapped phase under ideal conditions, phase unwrapping problem becomes challenging at the presence of noise, phase discontinuities and rapid variation of phase.

Phase unwrapping problem is traditionally addressed in two major approaches: path following approach and minimum norm approach. Path following approaches such as

---

Quality-Guided Phase Unwrapping (QGPU) algorithm [5] and branch cut algorithm [6] perform phase unwrapping by integrating phase along a selected path. Even though path following algorithms are relatively computationally efficient, they are not robust to noise. Minimum norm based algorithms [7] are robust to noise, but they are computationally inefficient than path following approaches.

In recent years, deep learning algorithms have gained popularity and achieved state-of-the-art performance in many computer vision tasks. Following this trend, a few recent studies [8, 9, 10, 11] have attempted to apply deep learning to address the phase unwrapping problem. Out of these, [8, 9, 10] have reformulated the phase unwrapping problem as a semantic segmentation task where Fully Convolutional Networks (FCNs) are trained to predict the wrap count at each pixel. Among these methods, Spoorthi et. al's [8] PhaseNet 2.0, a deep encoder-decoder architecture comprised of dense blocks [12], has the best phase unwrapping performance. To the best of our knowledge only Wang et al [11] has considered phase unwrapping as a regression problem where an FCN inspired by U-NET [13] and ResNet [14] is employed to directly estimate the true phase from wrapped phase. Many of these FCN based phase unwrapping methods have achieved good performance under varying levels of noise while taking significantly less computational times, compared to traditional methods. Although there are benefits, these networks require large scale datasets, thus reducing their suitability in real-world applications.

In addition, the methods based only on FCNs pose another issue. The locally performed convolutional and pooling operations of typical CNNs often ignore the global spatial relationships between different regions of the image. Since most of the real-world phase images contain certain spatial structures, modelling such global spatial relationships is vital when learning the mappings from wrapped phase to true phase. Recurrent Neural Networks (RNNs) [15] are a type of neural networks that can model the contextual relationships within a temporal sequence. However, it is not possible to directly apply an RNN to a feature map of an image. ReNet [16] and C-RNN [17] introduce ways of applying RNNs to feature maps, and inspired by them, Ryu et al [18] attempted to perform phase unwrapping in MRI images using a combination of a convolutional and recurrent networks. However,

this work does not provide any quantitative result, and has not considered the influence of noise on phase unwrapping. Although conventional RNNs may result in some success, due to their limitation of modelling long-term dependencies of a sequence, they would be less suitable in modelling the spatial relationships within a lengthy sequence derived from a feature map. Long Short-Term Memory (LSTM)[19], a special type of RNNs that is capable of modelling long term dependencies would be more adequate in this setting.

To address these shortcomings, in this paper, we propose an encoder-decoder CNN architecture that incorporates a Spatial-Quad Directional LSTM module which combines the power of FCNs and LSTMs in order to perform accurate and fast phase unwrapping, without having to train using a large scale dataset. We then describe a problem specific composite loss function comprised of variance of error and total variation of error losses to train this network. Finally, we report the findings of a comprehensive study which compared the proposed network with PhaseNet 2.0, Ryu et al's method and QGPU at varying levels of noise. These findings affirm that compared to other methods, the proposed network displays a strong robustness to severe noise conditions and a high computational efficiency when performing phase unwrapping.

2. METHODOLOGY

2.1. Data Generation

Datasets used in this study are comprised of synthetic phase images containing random shapes, and their corresponding wrapped phase images. These random shapes are created by adding and subtracting several Gaussians having varying shapes and positions. Mixing Gaussians this way ensures that irregular and arbitrary shapes are formed instead of explicit patterns, which would, in return, enable the proposed network to learn the phase continuities pertaining to any general pattern. In addition, randomly selected slopes are also added to these synthetic phase images along vertical and horizontal directions in order to incorporate the ramp phase.

The wrapped phase image $\psi(x, y)$ of synthetic phase image $\phi(x, y)$ is computed as follows.

$$\psi(x, y) = \angle \exp\left(i\phi(x, y)\right) \quad (1)$$

where, (x, y) denotes the spatial coordinates of a pixel.

Following this method, 2 datasets, each consisting of 6000 phase images (256×256) with values ranging from -44 to 44 were created. The phase images of only one of these datasets were randomly given additive Gaussian noise levels of $0, 5, 10, 20$ and 60 dB before wrapping them in order to simulate the noise prevalent in wrapped phase images of real-world applications. We shall denote this dataset as the "noisy" and the other as the "noise free".
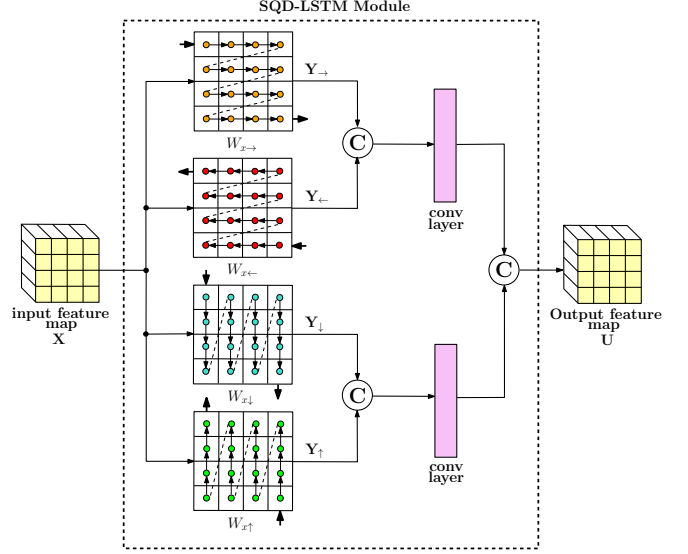

Fig. 1. The proposed SQD-LSTM module

2.2. Spatial Quad-Directional LSTM Module

Let $\mathbf{X} = \{x_{ij}\} \in \mathbb{R}^{w \times h \times c}$ be an input feature map, where w, h and c being the width, height and the feature dimensionality of $\mathbf{X}$. From $\mathbf{X}$, we can derive four distinct sequences as follows.

$$x_\rightarrow = \{\mathbf{r}_i\}_{i=1\ldots h} \, ; \mathbf{r}_i = (x_{i1}, x_{i2}, \ldots, x_{iw})$$
$$x_\leftarrow = \{\mathbf{r}_i\}_{i=h\ldots 1} \, ; \mathbf{r}_i = (x_{iw}, \ldots, x_{i2}, x_{i1})$$
$$x_\downarrow = \{\mathbf{r}_i\}_{i=1\ldots w} \, ; \mathbf{r}_i = (x_{1i}, x_{2i}, \ldots, x_{hi})$$
$$x_\uparrow = \{\mathbf{r}_i\}_{i=w\ldots 1} \, ; \mathbf{r}_i = (x_{hi}, \ldots, x_{2i}, x_{1i})$$

In other words, $x_\rightarrow$, $x_\leftarrow$, $x_\downarrow$ and $x_\uparrow$ represent the sequences obtained when one traverses $\mathbf{X}$ from left-to-right, right-to-left, top-to-bottom and bottom-to-top respectively as illustrated in Fig. 1. Let x be any of the four sequences above. Then $x^{(s)} \in \mathbb{R}^c$ where $s \in [1, \ldots, w \times h]$ is the feature vector that describes a corresponding region of the original image.

In order to model the spatial relationships within these four sequences, we utilize LSTMs which are specialized in modelling long-term dependencies. In simple terms, the input-output relationship of an LSTM layer for sequence x can be written as follows.

$$y^{(s)} = \mathbb{LSTM}\left(x^{(s)}, y^{(s-1)}; W_x, u\right) \quad (2)$$

where, $y^{(s)} \in \mathbb{R}^u$ is the LSTM output at s, W_x is the unified LSTM weights and u is the number of units in the layer. In our SQD-LSTM module, the $x_\rightarrow$, $x_\leftarrow$, $x_\downarrow$ and $x_\uparrow$ are processed by 4 separate LSTMs with unified weights $W_{x_\rightarrow}$, $W_{x_\leftarrow}$, $W_{x_\downarrow}$ and $W_{x_\uparrow}$, respectively. By rearranging the output sequences $y_\rightarrow$, $y_\leftarrow$, $y_\downarrow$ and $y_\uparrow$ from the 4 LSTMs, we obtain the output feature maps $\mathbf{Y}_\rightarrow$, $\mathbf{Y}_\leftarrow$, $\mathbf{Y}_\downarrow$ and $\mathbf{Y}_\uparrow$ each in $\mathbb{R}^{w \times h \times u}$.
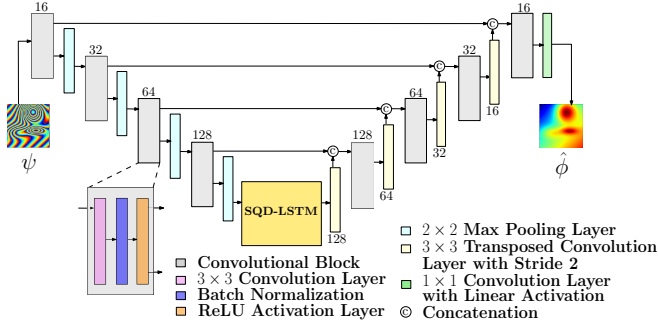
Fig. 2. The proposed network architecture. The number of filters in each convolutional block is stated above it and the number of filters in each transposed convolution layer is stated below it.

The output feature maps pertaining to horizontal ($\mathbf{Y}_\rightarrow$, $\mathbf{Y}_\leftarrow$) and vertical ($\mathbf{Y}_\downarrow$, $\mathbf{Y}_\uparrow$) directions are concatenated as illustrated in Fig. 1 and are passed through separate 3×3 convolutional layers containing d filters each, in order to explore the local association of the spatial features obtained from the LSTM layers. Finally, the output of the SQD-LSTM $\mathbf{U} \in \mathbb{R}^{w \times h \times 2d}$ is formed by concatenating the output feature maps from the two convolutional layers. In this study, u and d were set to 32 and 64 respectively.

2.3. Network Architecture

The proposed network architecture is illustrated in Fig. 2. In a nutshell, the architecture is comprised of a fully convolutional encoder-decoder network where the output of the encoder is passed through the proposed SQD-LSTM module before fed in to the decoder. The output feature map of the encoder is able to represent the local information of the input image. Feeding this encoder output to the SQD-LSTM module allows the network to learn the spatial dependencies between the local features contained in the encoder output. Subsequently, the output of SQD-LSTM module is fed in to the decoder network which increases the resolution of the output through transpose convolutional operations. Furthermore, in order to combine semantic features from the decoder layers and local features from the encoder layers, we add skip connections as shown in Fig. 2. Adding skip connections this way, ensures that the network assembles a more refined output at the later layers.

Every convolutional block in the network contains a 3×3 convolutional layer which is followed by a batch-normalization layer and a ReLU activation. Every encoder convolutional block is followed by a 2×2 max-pooling layer with a stride of 2, while every decoder convolutional block is preceded by 3×3 transposed convolutional layers with a stride of 2. Since the network performs a regression task, the final convolutional block of the decoder layer is succeeded by a 1×1 convolutional layer with a linear activation.

2.4. Loss Functions

Since we formulate the phase unwrapping problem as a regression task, the go to choice for the loss function is Mean Squared Error (MSE) loss. However, our experiments reveal that MSE loss, when employed to the proposed network, shows an insufficient convergence that results in poor performance of phase unwrapping. By (1), it follows that $\phi + 2\pi n$ where $\forall n \in \mathbb{Z}$ gives rise to the same wrapped phase ψ. Therefore, the phase unwrapping problem of ψ does not have a unique solution. Since MSE loss enforces the network to learn a unique solution, it does not fit well in to the nature of phase unwrapping problem. Hence, a loss function which allows for other solutions at convergence while increasing the similarity between the predicted phase $\hat{\phi}$ and true phase ϕ is required. To address these concerns, we adopt the composite loss function $\mathcal{L}_c$ defined below.

$$\mathcal{L}_c = \lambda_1 \mathcal{L}_{var} + \lambda_2 \mathcal{L}_{tv} \tag{3}$$

where,

$$\mathcal{L}_{var} = \mathbb{E}\big[(\hat{\phi} - \phi)^2\big] - \big(\mathbb{E}[(\hat{\phi} - \phi)]\big)^2 \tag{4}$$

$$\mathcal{L}_{tv} = \mathbb{E}\big[|\hat{\phi}_x - \phi_x| + |\hat{\phi}_y - \phi_y|\big] \tag{5}$$

and λ_1, λ_2 are the weights assigned for the two losses and were empirically set to 1 and 0.1 respectively during training. The variance of error loss $\mathcal{L}_{var}$ allows for alternate solutions at convergence while the total variation of error loss $\mathcal{L}_{tv}$ increases the increasing the similarity between $\hat{\phi}$ and ϕ by enforcing the network to match the gradients of them.

3. EXPERIMENTS AND RESULTS

The proposed network was implemented in Keras and was trained and tested on the two datasets mentioned in section 2.3, separately. In both occasions, the models were trained using ADAM optimizer with a learning rate of 0.001 and they converged within 10 epochs taking only ~ 1.5 hours to train. Similarly, Ryu et al's [18] network, PhaseNet 2.0 [8] and QGPU [5] were implemented and tested on the two datasets as well. Out of these, Ryu et al's network and PhaseNet 2.0 were trained on both noisy and noise free datasets. Furthermore, in order to assess the significance of SQD-LSTM module and the loss function $\mathcal{L}_c$, we trained and tested two separate U-NETS (which resemble the convolutional architecture of the proposed network), one with MSE and one with $\mathcal{L}_c$ as the loss function, using only the noise free dataset. All the above training and testing were performed on an NVIDIA Tesla K80 GPU with a fixed train - test split of 5000 - 1000 for each dataset.

To evaluate and compare these methods, we computed Normalized Root Mean Square Error (NRMSE - normalized by the range of the respective true phase image) of the unwrapped phase images and measured the average computational time per output, for each method. These results are
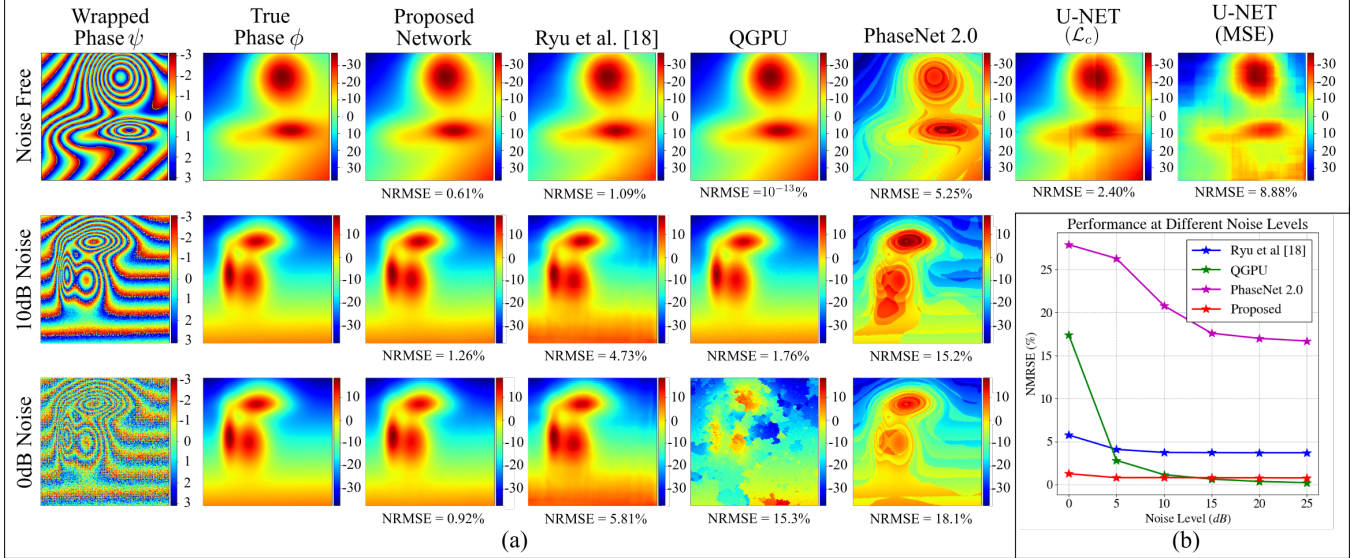
Fig. 3. (a) Unwrapped phase images obtained from different methods for selected noise free and noisy (10 dB and 0 dB) wrapped phase images. (b) Plot of NRMSE of proposed method, Ryu et al [18], QGPU and PhaseNet 2.0 against varying noise levels.

summarised in Table 1. Several selected unwrapped phase images obtained from each method are illustrated in Fig. 3 together with a plot of the noise levels versus NMRSE.

Table 1. Results

Method	Noise Free NRMSE	Noisy NRMSE	Computational Time (s)
UNET (MSE)	14.24%	-	0.234
UNET ($\mathcal{L}_c$)	2.75%	-	0.262
Ryu et al.[18]	2.23%	3.84%	0.687
PhaseNet 2.0 [8]	9.41%	17.53%	0.234
QGPU [5]	10^{-13}%	5.04%	35.42
Proposed Method	0.84%	**0.90%**	**0.054**

It follows from Table 1 that QGPU has achieved a nearly perfect performance in unwrapping noise free images. However, it exhibits poor performance when it comes to noisy images and has a very high average computational time (35.42 s) compared to deep learning approaches. The proposed method, on the other hand achieves a comparable performance (NRMSE = 0.84%) with that of QGPU in terms of noise free images, while attaining the highest performance (NRMSE = 0.9%) in terms of noisy images and the lowest average computational time (0.054 s) among the compared methods. As evident from Fig. 3 (b), the proposed network is able to accurately (NRMSE = 1.26%) unwrap the wrapped phase images with severe noise levels as high as SNR = 0 dB. It is also observed that the proposed network has surpassed the PhaseNet 2.0 which currently holds the state-of-the-art deep learning based phase unwrapping performance. PhaseNet 2.0, is a comparatively deeper network

comprised of dense blocks and therefore, its training process is highly data intensive. Nevertheless, in this study, its phase unwrapping error has increased, and this may be due to the relatively smaller size (5000) of the training dataset used. While being trained only on 5000 images, the proposed network however attained the highest phase unwrapping performance for noisy data among the considered methods, thus making it ideal for real-world applications with limited availability of data.

As shown by Table 1, the U-NET with $\mathcal{L}_c$ has a better phase unwrapping performance compared to the U-NET with MSE. Therefore, it is apparent that $\mathcal{L}_c$ is a better suited loss function for this problem than MSE. Also, it is evident that the proposed method performs better than the U-NET with $\mathcal{L}_c$. During experimentation, we also noticed that the proposed network converges faster than any of the considered methods. These observations point to the conclusion that the success of the proposed network is owed to the the SQD-LSTM module and the $\mathcal{L}_c$ loss function.

4. CONCLUSION

In this paper, we propose a novel convolutional architecture that encompasses a spatial quad-directional LSTM for phase unwrapping, by formulating it as a regression problem. When compared with several existing phase unwrapping methods, it is found that the proposed network attains state-of-the-art unwrapping performance even under sever noise conditions without requiring large scale datasets to train. Furthermore, the network expends a significantly less computational time on average to perform this task, making it ideal for applica-

tions which demand accurate and faster phase unwrapping. Looking forward, we hope to extend the network to address application specific challenges such as phase discontinuities.

5. REFERENCES

[1] J. Dong, T. Liu, F. Chen, D. Zhou, A. Dimov, A. Raj, Q. Cheng, P. Spincemaille, and Y. Wang, "Simultaneous phase unwrapping and removal of chemical shift (spurs) using graph cuts: Application in quantitative suscepti-bility mapping," *IEEE Transactions on Medical Imaging*, vol. 34, no. 2, pp. 531–540, 2015.

[2] 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.

[3] Zhangqiang He, Junning Cui, Jiubin Tan, Xingyuan Bian, and Wenxue Jiang, "Discrete fringe phase un-wrapping algorithm based on kalman motion estimation for high-speed i/q-interferometry," *Opt. Express*, vol. 26, no. 7, pp. 8699–8708, Apr 2018.

[4] P. R. Sukumar, R. G. Waghmare, R. K. Singh, G. R. K. S. Subrahmanyam, and D. Mishra, "Phase unwrap-ping with kalman filter based denoising in digital holo-graphic interferometry," in *2015 International Confer-ence on Advances in Computing, Communications and Informatics (ICACCI)*, 2015, pp. 2256–2260.

[5] Ming Zhao, Lei Huang, Qican Zhang, Xianyu Su, Anand Asundi, and Qian Kemao, "Quality-guided phase unwrapping technique: comparison of quality maps and guiding strategies," *Appl. Opt.*, vol. 50, no. 33, pp. 6214–6224, Nov 2011.

[6] Q. Huang, H. Zhou, S. Dong, and S. Xu, "Parallel branch-cut algorithm based on simulated annealing for large-scale phase unwrapping," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 7, pp. 3833–3846, 2015.

[7] 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.

[8] G. E. Spoorthi, R. K. Sai Subrahmanyam Gorthi, and S. Gorthi, "Phasenet 2.0: Phase unwrapping of noisy data based on deep learning approach," *IEEE Trans-actions on Image Processing*, vol. 29, pp. 4862–4872, 2020.

[9] Junchao Zhang, Xiaobo Tian, Jianbo Shao, Haibo Luo, and Rongguang Liang, "Phase unwrapping in optical metrology via denoised and convolutional segmentation networks," *Opt. Express*, vol. 27, no. 10, pp. 14903–14912, May 2019.

[10] Teng Zhang, Shaowei Jiang, Zixin Zhao, Krishna Dixit, Xiaofei Zhou, Jia Hou, Yongbing Zhang, and Cheng-gang Yan, "Rapid and robust two-dimensional phase unwrapping via deep learning," *Opt. Express*, vol. 27, no. 16, pp. 23173–23185, Aug 2019.

[11] Kaiqiang Wang, Ying Li, Qian Kemao, Jianglei Di, and Jianlin Zhao, "One-step robust deep learning phase un-wrapping," *Opt. Express*, vol. 27, no. 10, pp. 15100–15115, May 2019.

[12] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Wein-berger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269.

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical im-age segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 2015, pp. 234–241.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Con-ference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[15] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recur-rent neural networks and robust time series prediction," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 240–254, 1994.

[16] Francesco Visin, Kyle Kastner, Kyunghyun Cho, Mat-teo Matteucci, Aaron Courville, and Yoshua Bengio, "Renet: A recurrent neural network based alternative to convolutional networks," 2015, `https://arxiv.org/abs/1505.00393`.

[17] Z. Zuo, B. Shuai, G. Wang, X. Liu, X. Wang, B. Wang, and Y. Chen, "Convolutional recurrent neural net-works: Learning spatial dependencies for image rep-resentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2015, pp. 18–26.

[18] Ryu K, Gho S-M, Nam Y, Koch K, and Kim D-H, "Development of a deep learn-ing method for phase unwrapping mr images," 2019, `https://index.mirasmart.com/ISMRM2019/PDFfiles/4707.html`.

[19] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.