Research Article | Vol. 28, No. 4 / 17 February 2020 / Optics Express 4876
Optics EXPRESS

RedCap: residual encoder-decoder capsule network for holographic image reconstruction

TIANJIAO ZENG, HAYDEN K.-H. SO, AND EDMUND Y. LAM*

Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam, Hong Kong SAR, China
*elam@eee.hku.hk

Abstract: A capsule network, as an advanced technique in deep learning, is designed to overcome information loss in the pooling operation and internal data representation of a convolutional neural network (CNN). It has shown promising results in several applications, such as digit recognition and image segmentation. In this work, we investigate for the first time the use of capsule network in digital holographic reconstruction. The proposed residual encoder-decoder capsule network, which we call RedCap, uses a novel windowed spatial dynamic routing algorithm and residual capsule block, which extends the idea of a residual block. Compared with the CNN-based neural network, RedCap exhibits much better experimental results in digital holographic reconstruction, while having a dramatic 75% reduction in the number of parameters. It indicates that RedCap is more efficient in the way it processes data and requires a much less memory storage for the learned model, which therefore makes it possible to be applied to some challenging situations with limited computational resources, such as portable devices.

1. Introduction

As a powerful interferometric imaging modality, digital holography (DH) is able to capture the diffracted wavefront from an object or a three-dimensional scene, by encoding the interference patterns in a hologram utilizing object light and reference light [1]. Both amplitude and phase information can be reconstructed numerically from the hologram [2]. Given such features, DH is a fast and non-invasive method for amplitude and phase measurement, and has been applied in various domains such as surface topography [3], microscopic imaging [4], 3D recognition [5], and particle measurement [6]. To ensure the quality of further applications, holographic reconstruction has been considered as an important task for decades.

The traditional physics-based methods, including Fresnel approach [7], angular spectrum approach (ASM) [8], and convolution approach (CONV) [9], require detailed information about the experimental setup (such as the wavelength of the laser, and the pixel pitch of the sensor). In the last few years, superior performance of deep learning-based methods, especially CNN-based networks, have been reported in various imaging problems, such as in phase aberration compensation [10,11], ghost imaging [12], light field [13], scatter imaging [14], autofocusing [15], super-resolution [16], denoising [17], despeckling [18,19], classification [20,21], and depth-of-field extension [22]. Emerging data-driven holographic reconstruction approaches have also been investigated [23–26]. Designed on the extension of various CNN models including U-Net [27], deep residual network (ResNet) [28], and other encoder-decoder models [29], these approaches enable direct reconstruction from raw holograms without any prior knowledge of physical parameters in the imaging process, and are more robust and less time-consuming compared with traditional physics-based methods.

Despite the remarkable performance and successful applications in various fields, CNNs still have their limitations. The pooling operation, especially max-pooling, while serving as an important contributor to its robust and strong feature extraction capability, discards valuable information. Moreover, CNNs do not preserve the spatial relationship between features in

#383350 https://doi.org/10.1364/OE.383350
Journal © 2020 Received 18 Nov 2019; revised 26 Jan 2020; accepted 27 Jan 2020; published 7 Feb 2020

Optics EXPRESS

adjacent layers, due to the scalar nature of their internal data representation. To overcome these drawbacks, a new concept of building block to represent features, which is named capsule, has been introduced by Sabour et al. [30], where complex information at each capsule is stored in vectors instead of scalars. Moreover, the similarity, also interpreted as agreement between lower- and higher-level capsule vectors, is encoded in a transformation matrix in the operation of dynamic routing. With all information encapsulated in vector form and agreement between these vectors taken into account, the capsule network can model the hierarchical relationships inside the architecture better, and is able to learn features in a more efficient way. For illustration, a three-layer capsule network model, named CapsNet, is established in [30] and has been demonstrated to work well in classification on MNIST dataset [31]. This concept is then further investigated and adjusted to different applications and tasks [32–35].

In this work, we aim at extending the use of capsule network to digital holographic reconstruction. The CapsNet [30] uses fully-connected capsules with dynamic routing performed globally along all dimensions, including capsule types (depth), height, and width, which however will result in parameter explosion of the transformation matrix, and consequently incur an expensive computational cost when handling much larger images. Therefore, we adopt the convolutional and deconvolutional capsule layers introduced by [32], where a sliding transformation kernel is applied and shared spatially across blocks similar to a convolutional kernel. The dynamic routing is only performed depth-wise along different capsule types, and the number of parameters, as well as computational cost, can be dramatically cut down. To compensate for the loss that no routing operation is performed in the spatial dimension (along height and width), we propose a novel windowed spatial dynamic routing that is only performed within a block of capsules between adjacent layers for parameter reduction.

With the use of the proposed routing algorithm, a capsule layer mimicking the function of pooling operation, named pooling capsule layer, is introduced to reduce the spatial size of the data transmitting through the network, which thus enlarges the receptive field of capsules in higher layers. In this way, we divide the global dynamic routing into two ways, namely, depth-wise and spatial dynamic routing. We then form a capsule-based residual block composed of a pooling capsule layer and a convolutional capsule layer, with a skip connection adding the input data with the output data in order to learn a deeper network. The framework of our proposed capsule-based holographic reconstruction network, residual encoder-decoder capsule network (RedCap), leveraging residual capsule blocks for the encoder process and deconvolutional capsule layers for the decoder, is exhibited in Fig. 1.

The contributions of our work can be summarized as follows:

- The proposed RedCap architecture is the first attempt to extend capsule network to digital holographic reconstruction.
- We design a new mechanism replacing the original fully-connected dynamic routing with spatial and depth-wise routing, and propose a novel windowed spatial dynamic routing to address the issue of the expensive computational cost of the original capsule layer.
- We introduce the concept of pooling capsule layer using the windowed spatial dynamic routing to downsample feature map, and establish a residual capsule unit combining the idea of the residual block with the capsule layers to enable deeper training.
- We evaluate the reconstruction performance of RedCap on holograms obtained from amplitude objects, and compare with both traditional convolution method CONV [9] and learning-based holographic reconstruction network (HRNet) [23]. RedCap shows better results in both objective measures and visual inspection with an approximately 75% reduction in the number of parameters.

Optics EXPRESS

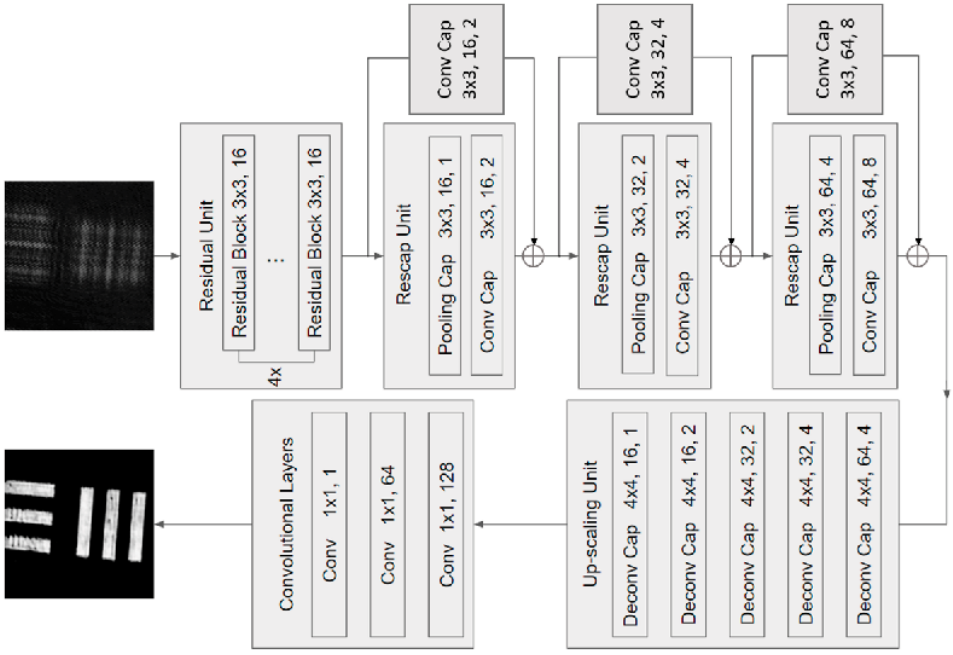


Fig. 1. Architecture of proposed RedCap model for holographic reconstruction, indicating the details of layers including kernel sizes, channel numbers, as well as number of capsule types, etc.

2. Method

Digital holographic reconstruction aims at numerically reconstructing the wavefront information of an object, especially amplitude and phase information from the interference patterns stored in a captured hologram. Such patterns, with complex imaging information to be extracted, is formed by recording the interference of two waves, namely, a reference wave and an object wave carrying the object information. The mapping from a captured hologram to a reconstructed image can be modeled as an inverse imaging problem, and the learning-based approaches attempt to achieve this computation through training a neural network.

2.1. Capsule network

In a wide range of tasks that involve images, CNN-based architecture has become a popular learning approach with very good performance [36]. The idea of a convolutional layer lies in the fact that local image pixels contain important features such as edges, and these local features are shared among remote locations all over the image. Different feature patterns can be detected with different weight matrices, and the detected features at lower layers are combined as a weighted sum; they are then passed through a nonlinear function to form more complicated higher-order features at deeper layers.

However, due to the scalar nature of neurons and additive operations, the spatial relationships of lower-level features and higher-order features are not well described by CNN [37]. Moreover, to enable neurons from the higher layers to learn higher-order features in a wider range of input data, CNN enlarges the field of view by downsampling the feature map. This is usually accomplished with pooling layers, especially max-pooling, or convolutional layers that reduce the spatial dimension of the data with a stride larger than one. Despite the good performance in achieving invariance of neuronal activities, which makes the network more robust against small

Optics EXPRESS

changes of the input, max-pooling is a primitive operation that discards valuable information. Furthermore, it only passes a scalar output indicating the position of the most active features, without taking spatial relationships of features into consideration.

Capsule network with dynamic routing has been designed to address these issues [30]. Inspired by how internal hierarchical relationships of geometrical data are represented in computer graphics, the fundamental units of the network are denoted as capsules, which are in the form of vectors instead of scalars. They can encode a broader range of information of the detected features, including pose (such as orientation and position), magnitude, and some other attributes [38]. Specifically, the pose of the capsules denotes the spatial information of the extracted features, and the magnitude (length of the capsule vectors) represents the existence probability of the detection of such features. Thus, as the feature detector is applied on the input data, identical features that have a slight change in the position or orientation would result in a different pose but the same magnitude. With the spatial relationship of lower- and higher-level features encoded in the translation process between capsules at adjacent layers, the changes in child capsules, as the feature rotates or moves over the image, will be translated into an equivalent variance in the state of the parent capsules. Yet, the feature can still be detected since the existence probability stays the same. Such property is referred to as translation equivariance [38], which is more powerful than the invariance of neuronal activities. This robustness of network performance is therefore superior to what can be obtained in standard CNNs.

Furthermore, the vector-form data representation of a capsule network also enables a novel dynamic routing algorithm, where the essential idea is referred to as "routing by agreement" [30]. This operation can be intuitively interpreted as a clustering mechanism, which ensures that each parent capsule at the higher layers receives inputs from the child capsules at the lower layers that agree with it, and therefore forms a "part-whole" relationship between features at adjacent layers.

Mathematically, by multiplying a corresponding transformation matrix W, the vector $\boldsymbol{u}_i$ of each child capsule i at the lower layers will generate a prediction vector, also called a vote, i.e.,

$$\hat{\boldsymbol{u}}_{j|i} = W\boldsymbol{u}_i, \tag{1}$$

for each parent capsule j at the next layer. All votes from different child capsules are then clustered to generate the vector of each parent capsule, i.e.,

$$\boldsymbol{p}_j = \sum_i \beta_{ij}\hat{\boldsymbol{u}}_{j|i}, \tag{2}$$

with the corresponding coupling coefficient β_{ij} determined by the level of agreement or similarity between the vote generated by each child capsule i and parent capsule j. A vector-to-vector nonlinear function is then applied to $\boldsymbol{p}_j$ to obtain an output vector $\boldsymbol{v}_j$, with length between 0 and 1, for the parent capsule j, since the length indicates the existence probability. The similarity is computed by a scalar product $\boldsymbol{v}_j \cdot \hat{\boldsymbol{u}}_{j|i}$, which iteratively updates the value of the coupling coefficient β_{ij}. A larger β_{ij} represents a higher probability that the features from a child capsule i should be passed to a parent capsule j. In this way, the existence probability of the higher-level capsule relies on the agreement of the votes. The spatial relationships between the lower-level capsules and the higher-level capsules are therefore learned accordingly, which enables a more efficient way of training.

2.2. RedCap

The main challenge in extending the capsule network for holographic reconstruction lies in the computationally expensive full connection of such a network, which not only limits the size of holograms to be processed but also constraints the depth of the network that is considered practical. Considering the input child capsules to the routing operation to be 128×128, 16-dimensional, with one capsule type, and they are being routed to a set of 2 capsule

Optics EXPRESS

Algorithm 1 Windowed Spatial Dynamic Routing

Require: $\hat{\boldsymbol{u}}_{j|i}$: prediction vector
Require: r: number of routing iterations
Require: l: layer
Require: h^l, w^l, h^{l+1}, w^{l+1}: height and width of a user-defined window for layer l and $(l+1)$
1: for all capsule i within a $h^l \times w^l$ window in layer l and capsule j within $h^{l+1} \times w^{l+1}$ window in layer $(l+1)$: $b_{ij} \Leftarrow 0$
2: **for** r iteration **do**
3: for all capsule i within a $h^l \times w^l$ window in layer l: $\boldsymbol{\beta}_i \Leftarrow \text{softmax}(\boldsymbol{b}_i)$
4: for all capsule j within a $h^{l+1} \times w^{l+1}$ window in layer $(l+1)$: $\boldsymbol{s}_j \Leftarrow \sum_{(h^l, w^l)} \beta_{ij} \hat{\boldsymbol{u}}_{j|i}$
5: for all capsule j within a $h^{l+1} \times w^{l+1}$ window in layer $(l+1)$: $\boldsymbol{v}_j \Leftarrow \text{squash}(\boldsymbol{s}_j)$
6: for all capsule i within a $h^l \times w^l$ window in layer l and capsule j within $h^{l+1} \times w^{l+1}$ window in layer $(l+1)$: $b_{ij} \Leftarrow b_{ij} + \hat{\boldsymbol{u}}_{j|i} \cdot \boldsymbol{v}_j$
7: **end for**
8: **return** $\boldsymbol{v}_j$

types, 64×64, 32-dimensional parent capsules. The fully-connected dynamic routing requires $128 \times 128 \times 16 \times 64 \times 64 \times 32 \times 2 = 68,719,476,736$ parameters, which makes the implementation of the network impossible on holographic reconstruction task. To address this issue, we introduce a novel mechanism to divide the fully connected routing into two separate ones along different dimensions, namely, spatial routing and depth-wise routing.

For the former, we develop a windowed spatial dynamic routing approach, which is described in Algorithm 1 and Fig. 2, instead of the conventional global routing. Full connection is only implemented among capsules within a user-defined local block at adjacent layers. Mathematically, for any given layer l, a moving window of size $h^l \times w^l$ gathers the input data with c^l channels, and forms a set of capsule blocks. Let $N = h^l \times w^l$; then, this set contains a total of N capsule vectors, each of which is c^l-dimensional, i.e., $U = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_N\}$. In each single block, instead of routing to all capsules at the next layer, capsule vectors will only be routed locally, corresponding to a $h^{l+1} \times w^{l+1}$ grid of higher-order capsules. Similarly, we can let $M = h^{l+1} \times w^{l+1}$, and the M capsule vectors are $V = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_M\}$, with c^{l+1} channels at layer $(l+1)$. Each low-level capsule $\boldsymbol{u}_i$ will generate a prediction vector $\hat{\boldsymbol{u}}_{j|i}$ for each parent capsule $\boldsymbol{p}_j$ by multiplying a transformation weight matrix W_{ij} of size $c^{l+1} \times c^l$.

Note that the transformation matrices are different for different low-level and high-level capsule pairs within each block. Nevertheless, they will be shared across different blocks, in order to reduce the number of parameters significantly and lighten the computational complexity of the network. For a parent capsule $\boldsymbol{p}_j$ where j lies within the $h^{l+1} \times w^{l+1}$ window, we obtain a set of prediction vectors, denoted as

$$\hat{U}_j = \{\hat{\boldsymbol{u}}_{j|1}, \hat{\boldsymbol{u}}_{j|2}, \ldots, \hat{\boldsymbol{u}}_{j|N} \mid \hat{\boldsymbol{u}}_{j|i} = W_{ij}\boldsymbol{u}_i\}. \tag{3}$$

The output capsule $\boldsymbol{v}_j$ is computed as a weighted sum of all prediction vectors with routing coefficient β_{ij}, and it then passes through a nonlinear vector-to-vector activation function (called a "squash" function here, which we will explain below), which can be expressed as

$$\boldsymbol{v}_j = \text{squash}\left(\sum_i \beta_{ij}\hat{\boldsymbol{u}}_{j|i}\right), \qquad \beta_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})}. \tag{4}$$

The parameter b_{ij} indicates the prior log probability between a higher-level capsule j in the $h^{l+1} \times w^{l+1}$ grid and one of its votes $\hat{\boldsymbol{u}}_{j|i}$, and it undergoes a softmax operation to obtain a non-negative routing coefficient β_{ij} with a total probability summed to 1. Similar to the original

Optics EXPRESS

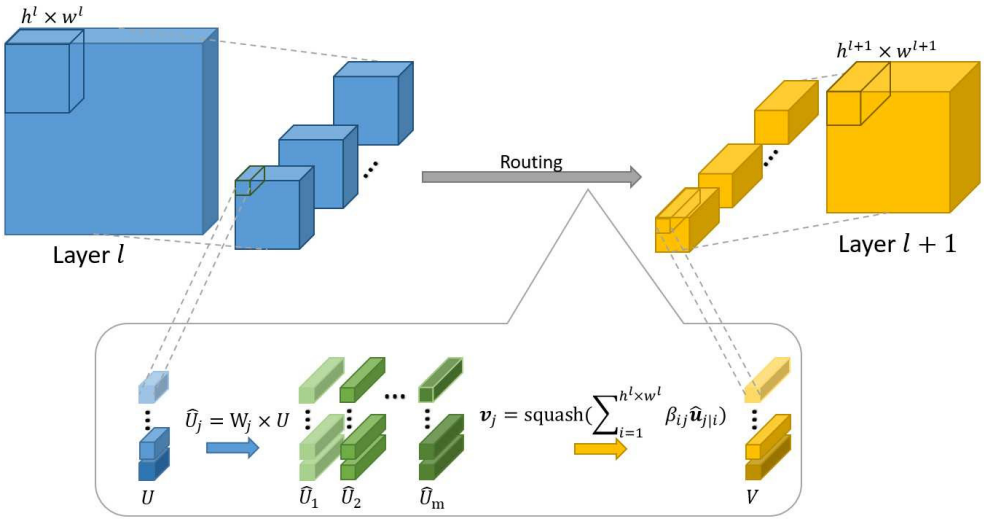


Fig. 2. Windowed spatial dynamic routing. In layer l, a window of size $h^l \times w^l$ will slide across the entire capsule tensor and generate local capsule blocks. For each block in layer l, $h^l \times w^l$ capsules will be used to predict capsules within a $h^{l+1} \times w^{l+1}$ in layer $(l+1)$ towards a weighted combination. All weights are initialized to be equal and will be updated iteratively based on the agreement between $\hat{\boldsymbol{u}}_{j|i}$ and $\boldsymbol{v}_j$.

capsule network, b_{ij} is initialized to be zero and updated according to $\boldsymbol{v}_j \cdot \hat{\boldsymbol{u}}_{j|i}$ in the iterative windowed spatial dynamic routing process. The "squash" function, as introduced in [30], is applied to ensure that the length of the capsule vector is restricted within an interval between 0 and 1 to indicate the existence probability of this capsule. We can express this as

$$\boldsymbol{v}_j = \mathrm{squash}(\boldsymbol{p}_j) = \frac{\|\boldsymbol{p}_j\|^2}{1 + \|\boldsymbol{p}_j\|^2} \cdot \frac{\boldsymbol{p}_j}{\|\boldsymbol{p}_j\|}. \tag{5}$$

As for depth-wise dynamic routing, we use the same routing approach proposed in [32], where in the convolutional and deconvolutional capsule layers, the child capsules will first undergo convolution or deconvolution operations. The output vectors are then viewed as votes. Afterwards, the routing procedure between the votes and the parent capsules is only carried out in the dimension of the capsule types, which we interpret as depth-wise dynamic routing. Mathematically, suppose that the input data to layer l is of size $H^l \times W^l \times c^l \times T^l$, where H^l and W^l denote the height and width of the input data, c^l represents the channel number, and T^l indicates the number of capsule types. For each parent capsule type t_j from T^{l+1} number of capsule types, at layer $l+1$, a convolution or deconvolution operation is applied, and a vote tensor of size $H^{l+1} \times W^{l+1} \times c^{l+1} \times T^l$ is obtained. Then, $H^{l+1} \times W^{l+1}$ parallel depth-wise dynamic routings are performed across capsule types. Each parent capsule $\boldsymbol{p}_{t_j}$ receives a group of votes, described as $U_{t_j} = \{\hat{\boldsymbol{u}}_{t_j|t_1}\}, \ldots, \hat{\boldsymbol{u}}_{t_j|t_{T^l}}$. Therefore, the predicted vector $\boldsymbol{v}_{t_j}$ of each parent capsule is computed as a weighted sum of all votes in group U_{t_j} with weight $\beta_{t_j t_j}$, and then passes through the squashing function,

$$\boldsymbol{v}_{t_j} = \mathrm{squash}\left(\sum_{t_i} \beta_{t_i t_j} \hat{\boldsymbol{u}}_{t_j|t_i}\right). \tag{6}$$

Similarly, the routing coefficient $\beta_{t_i t_j}$ is updated iteratively according to $\boldsymbol{v}_{t_j} \cdot \hat{\boldsymbol{u}}_{t_j|t_i}$. Therefore, the routing operation is only implemented depth-wise between the lower- and higher-level capsule types.

Optics EXPRESS

The procedure of our proposed RedCap network is illustrated in Fig. 1, which consists of a convolutional residual unit and three residual capsule cells to extract features for the encoder part, and three deconvolutional units to reconstruct images for the decoder network. Since the size of an input hologram is large, it is not computationally efficient to apply the routing operation directly at the beginning of the network. Thus, we use a convolutional residual unit with four residual blocks and each of them consists of two $3 \times 3 \times 16$ convolutional layers, followed by a batch normalization layer and a rectified linear unit (ReLU) activation function right after each convolutional layer. They are stacked successively as the initial feature detector and meanwhile downsample the feature map to increase the receptive field. The output is then processed by three residual capsule units (denoted as Rescap), containing a pooling capsule layer with windowed spatial dynamic routing, and a 3×3 convolutional capsule layer with depth-wise dynamic routing algorithm. For pooling capsule layer, balancing between the performance and the computational cost, the windows of spatial dynamic routing are set to be of size 3×3 and 1×1 for adjacent layers with a sliding step of 2, which can be interpreted as a pooling operation with local spatial hierarchies included to increase the field of view of higher layers. The input to each Rescap Unit is passed to the output of the unit with a shortcut connection, in order to avoid the possible gradient vanishing or exploding problem in a deep network.

In the decoder network, there is an up-scaling unit composed of five dilated deconvolutional capsule layers [32] with a stride of 2 and a kernel size of 4×4 to reshape the data to its original size. In the end, three 1×1 convolutional layers are applied for fine-tuning the final reconstructed output images. The backpropagation operation is performed based on the loss calculated by the mean-square error between the reconstructed image and the corresponding ground truth image.

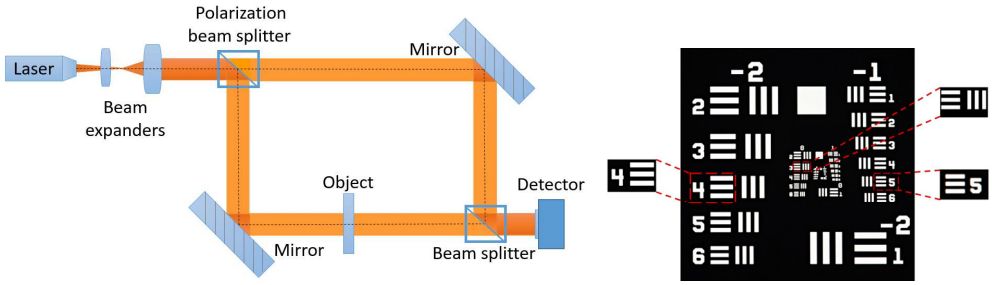
3. Experiment and results

The hologram data used are a subset of data from [23], which are various amplitude objects acquired by imaging different local areas on the negative USAF 1951 test target exhibited in Fig. 3(b). The experiment is conducted using an off-axis digital holography imaging system, as shown in Fig. 3(a), with a lens-free Mach-Zehnder interferometer and two linear motion controllers (Newport, CONEX-LTA-HL) for axial movement. The angles between the object light and reference light are manually adjusted via the mirror, and the setup uses no objective lens. The interference patterns on the hologram plane are recorded as the raw hologram data. The collected samples of original size 1024×1280 are cropped to size 800×800 due to memory limitation, and we then divide the dataset into 80% for training, 10% for validation and 10% for testing. The label images are obtained by backpropagating the imaging process using the CONV method and then manually removing the artifacts as well as suppressing background noise in the images towards thresholding. The network is trained at an initial learning rate of 10^{-2} with a decaying factor of 0.9 for every 10 epochs, and stopped at the stagnation of validation loss. We adopts Adam as the optimizer [39] and a batch size of 2 for parameter update. A weight decay of 10^{-4} is used for regularization. The training process is implemented using Pytorch on an Nvidia GTX 1070.

The proposed RedCap is compared with the conventional method CONV, which is implemented with detailed parameters of the optical system including pixel pitch of the detector, object distance, and laser wavelength, and a CNN-based holographic reconstruction neural network HRNet [23]. As an extension of ResNet [28], it is composed of a batch normalization (BN) and a rectified linear unit (ReLU) activation, six residual blocks and a sub-pixel convolutional layer [40]. Each residual block in HRNet consists of two successive parts containing a convolutional layer followed by a BN and a ReLU activation, and three of the residual blocks utilize a max-pooling layer at the beginning of the block.

The objective assessments, the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) [41], are used to quantitatively evaluate the reconstruction performance of all three

Optics EXPRESS



(a) Experiment setup (b) Sample

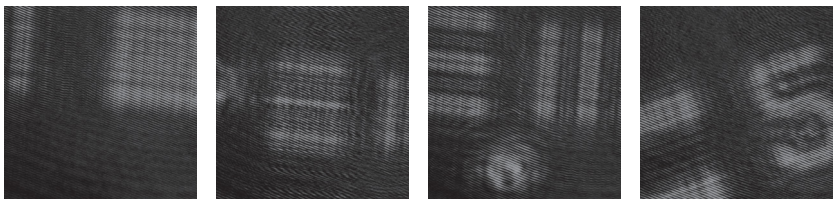
Fig. 3. The digital holography experiment setup and amplitude sample (negative USAF 1951 target).

methods. As shown in Table 1, the learning-based methods outperform the CONV method in both metrics, and overall, RedCap presents the highest PSNR and SSIM values (emphasized in red) than the other two methods. It is worth noting that HRNet contains more than 2.8 million parameters, while RedCap cuts this number down to approximately 0.7 million.

Table 1. Comparison of reconstruction performance for the amplitude object among CONV, HRNet and RedCap.

Measure	Methods	Test
PSNR (dB)	CONV	20.54
	HRNet	24.62
	RedCap	24.79
SSIM	CONV	0.2644
	HRNet	0.9081
	RedCap	0.9234
Number of parameters	HRNet	2,86 M
	RedCap	0.71 M

For visual inspection, four sample holograms are selected (Fig. 4) and reconstructed through RedCap, HRNet and CONV. The reconstructed images of different approaches together with the ground truth images are presented in Fig. 5. It is obvious that both RedCap and HRNet perform much better results in suppressing background noise than the CONV method, especially concerning the artifacts around the edges of the objects. Moreover, the images reconstructed by RedCap are much less blurry, and appear to restore more details compared with those obtained by HRNet. As can been seen, RedCap exhibits a better capability of preserving sharper edges than



(a) (b) (c) (d)

Fig. 4. Sample holograms.

Optics EXPRESS

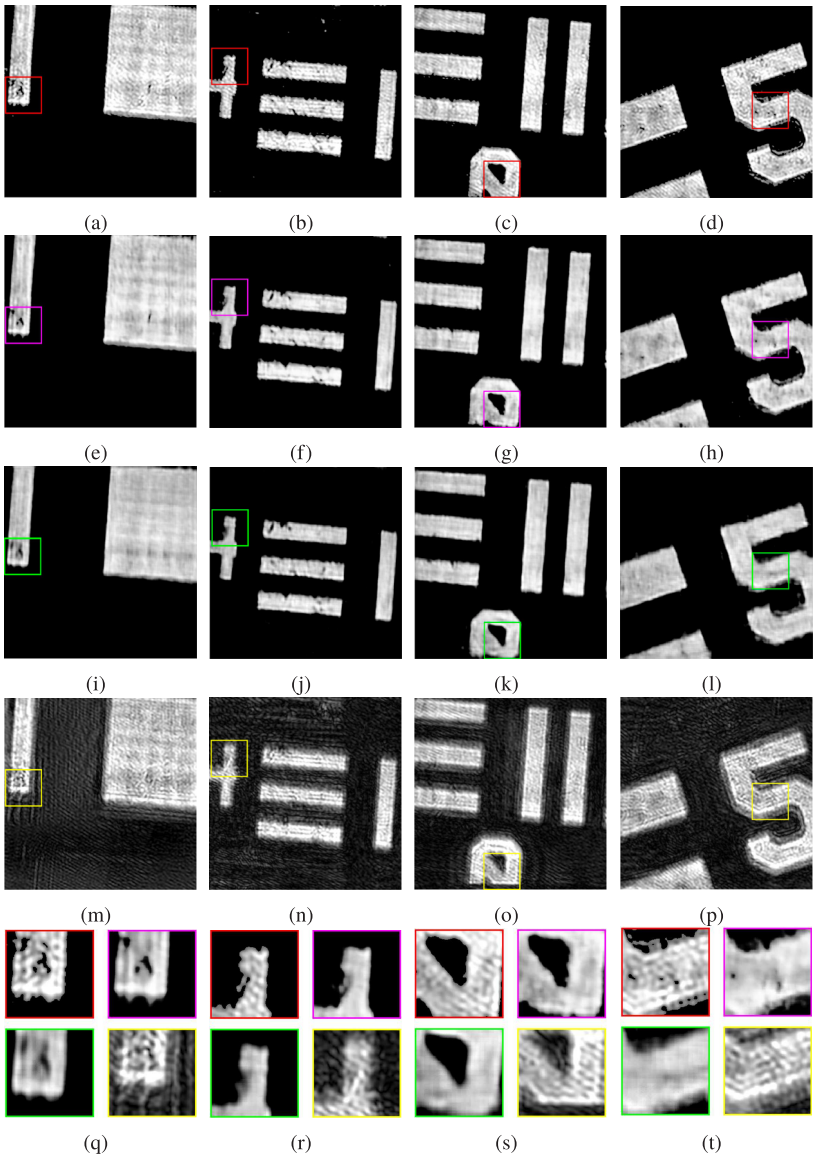


Fig. 5. Ground truth images (a)–(d), and reconstructed images using RedCap (e)–(h), HRNet (i)–(l) and CONV(m)–(p). (q)–(t) are zoomed-in regions of Ground Truth images (red), and reconstructed images by RedCap (magenta), HRNet (green) and CONV method (yellow).

Optics EXPRESS

HRNet. The superior performance and the significant reduction in total parameters demonstrate the high efficiency of RedCap in the learning process of holographic reconstruction.

Table 2. Detailed calculation of parameters of RedCap.

Layer Number	Layer Type	Configuration	Number of Parameters
Layer 1: Residual Unit	Residual Block	$3 \times 3 \times 1 \times 16$ $3 \times 3 \times 16 \times 16$	2,448
	Residual Block	$3 \times 3 \times 16 \times 16$ $3 \times 3 \times 16 \times 16$	4,608
	Residual Block	$3 \times 3 \times 16 \times 16$ $3 \times 3 \times 16 \times 16$	4,608
	Residual Block	$3 \times 3 \times 16 \times 16$ $3 \times 3 \times 16 \times 16$	4,608
Layer 2: Rescap Unit	Pooling Cap	$3 \times 3 \times 16 \times 16$	2,304
	Conv Cap	$3 \times 3 \times 16 \times 2 \times 16$	4,608
	skip connection	$3 \times 3 \times 16 \times 2 \times 16$	4,608
Layer 3: Rescap Unit	Pooling Cap	$3 \times 3 \times 16 \times 32$	4,608
	Conv Cap	$3 \times 3 \times 32 \times 4 \times 32$	36,864
	skip connection	$3 \times 3 \times 16 \times 4 \times 32$	18,432
Layer 4: Rescap Unit	Pooling Cap	$3 \times 3 \times 32 \times 64$	18,432
	Conv Cap	$3 \times 3 \times 64 \times 8 \times 64$	294,912
	skip connection	$3 \times 3 \times 32 \times 8 \times 64$	147,456
Layer 5: Up-scaling Unit	Deconv Cap	$4 \times 4 \times 64 \times 4 \times 32$	131,072
	Deconv Cap	$4 \times 4 \times 32 \times 2 \times 16$	16,384
	Deconv Cap	$4 \times 4 \times 16 \times 1 \times 16$	4,096
	Deconv Cap	$4 \times 4 \times 16 \times 1 \times 16$	4,096
	Deconv Cap	$4 \times 4 \times 16 \times 1 \times 1$	256
Layer 6	convolution	$1 \times 1 \times 1 \times 128$	128
Layer 7	convolution	$1 \times 1 \times 128 \times 64$	8,192
Layer 8	convolution	$1 \times 1 \times 64 \times 1$	64
Total parameters			712,784

4. Conclusion

In conclusion, we propose a novel capsule-based deep learning network RedCap for holographic reconstruction, which is the first work expanding the recently emerged capsule network to the numerical reconstruction of DH. In RedCap, global dynamic routing of the capsule network is separated into depth-wise and spatial dynamic routing. We introduce a novel windowed spatial dynamic routing to cut down the number of parameters and apply it into pooling capsule layers, replacing the pooling operation in CNN-based networks, to increase the receptive field of the network. We then use a convolutional capsule layer with depth-wise dynamic routing and pooling capsule layer with windowed spatial dynamic routing to form a residual capsule unit, which eases the training of a deeper network. By using depth-wise and windowed dynamic routing, RedCap significantly reduces the number of parameters in the network and breaks the limit in the image size of the capsule network due to high computation complexity and memory burden. We also demonstrate the superiority and efficiency of proposed RedCap in reconstructing high-quality images from raw holograms compared to the traditional method and CNN-based

Optics EXPRESS

reconstruction neural network. Experimentally, RedCap outperforms the other two methods in both evaluation metrics (PSNR and SSIM) and shows better performance in details restoration, especially in preserving sharp edges. Moreover, RedCap achieves better results with 75% fewer parameters in comparison with CNN-based HRNet. Our work demonstrates that the capability of encoding spatial relationships of objects possessed by capsules, helps improve the efficiency and performance of the network in the holographic reconstruction task.

Appendix: Details of RedCap

The detailed parameter calculation of the proposed RedCap has been shown layer by layer in Table 2.

Funding

Research Grants Council, University Grants Committee (17200019, 17201818, 17203217); the University of Hong Kong (104005009, 104005438).

Acknowledgments

The authors would like to thank Dr. Zhenbo Ren at Northwestern Polytechnical University in Xi'an, China, for sharing DH datasets and codes of HRNet.

Disclosures

The authors declare no conflicts of interest.

References

1. D. Gabor, "A new microscopic principle," Nature **161**(4098), 777–778 (1948).
2. U. Schnars, C. Falldorf, J. Watson, and W. Jüptner, *Digital Holography and Wavefront Sensing* (Springer, 2016).
3. E. Cuche, P. Marquet, and C. Depeursinge, "Simultaneous amplitude-contrast and quantitative phase-contrast microscopy by numerical reconstruction of Fresnel off-axis holograms," Appl. Opt. **38**(34), 6994–7001 (1999).
4. M. K. Kim, *Digital Holographic Microscopy: Principles, Techniques, and Applications* (Springer Science and Business Media, 2011).
5. B. Javidi and E. Tajahuerce, "Three-dimensional object recognition by use of digital holography," Opt. Lett. **25**(9), 610–612 (2000).
6. S. Murata and N. Yasuda, "Potential of digital holography in particle measurement," Opt. Laser Technol. **32**(7-8), 567–574 (2000).
7. P. Picart and J. Leval, "General theoretical formulation of image formation in digital Fresnel holography," J. Opt. Soc. Am. A **25**(7), 1744–1761 (2008).
8. J. W. Goodman, *Introduction to Fourier Optics* (Roberts and Company, 2004), 3rd ed.
9. T. M. Kreis, M. Adams, and W. P. Jüptner, "Methods of digital holography: a comparison," in *Optical Inspection and Micromeasurements II*, vol. 3098 (1997), pp. 224–233.
10. T. Nguyen, V. Bui, V. Lam, C. B. Raub, L.-C. Chang, and G. Nehmetallah, "Automatic phase aberration compensation for digital holographic microscopy based on deep learning background detection," Opt. Express **25**(13), 15043–15057 (2017).
11. Z. Ren, Z. Xu, and E. Y. Lam, "Phase aberration compensation in digital holographic microscopy using regression analysis," in *OSA Topical Meeting in Computational Optical Sensing and Imaging*, (2018), p. JTh3B.5.
12. M. Lyu, W. Wang, H. Wang, H. Wang, G. Li, N. Chen, and G. Situ, "Deep-learning-based ghost imaging," Sci. Rep. **7**(1), 17865 (2017).
13. N. Meng, X. Sun, H. K.-H. So, and E. Y. Lam, "Computational light field generation using deep nonparametric Bayesian learning," IEEE Access **7**, 24990–25000 (2019).
14. S. Li, M. Deng, J. Lee, A. Sinha, and G. Barbastathis, "Imaging through glass diffusers using densely connected convolutional networks," Optica **5**(7), 803–813 (2018).
15. Z. Ren, Z. Xu, and E. Y. Lam, "Learning-based nonparametric autofocusing for digital holography," Optica **5**(4), 337–344 (2018).
16. Z. Ren, H. K.-H. So, and E. Y. Lam, "Fringe pattern improvement and super-resolution using deep learning in digital holography," IEEE Trans. Ind. Inf. **15**(11), 6179–6186 (2019).
17. A. Shah, L. Zhou, M. D. Abrámoff, and X. Wu, "Multiple surface segmentation using convolution neural nets: Application to retinal layer segmentation in OCT images," Biomed. Opt. Express **9**(9), 4509–4526 (2018).

Optics EXPRESS

18. W. Jeon, W. Jeong, K. Son, and H. Yang, "Speckle noise reduction for digital holographic images using multi-scale convolutional neural networks," Opt. Lett. **43**(17), 4240–4243 (2018).
19. T. Zeng, H. K.-H. So, and E. Y. Lam, "Computational image speckle suppression using block matching and machine learning," Appl. Opt. **58**(7), B39–B45 (2019).
20. S.-J. Kim, C. Wang, B. Zhao, H. Im, J. Min, H. J. Choi, J. Tadros, N. R. Choi, C. M. Castro, and R. Weissleder, "Deep transfer learning-based hologram classification for molecular diagnostics," Sci. Rep. **8**(1), 17003 (2018).
21. N. Meng, H. K.-H. So, and E. Y. Lam, "Computational single-cell classification using deep learning on bright-field and phase images," in *IAPR Conference on Machine Vision Applications*, (2017), pp. 164–167.
22. Y. Wu, Y. Rivenson, Y. Zhang, Z. Wei, H. Günaydin, X. Lin, and A. Ozcan, "Extended depth-of-field in holographic imaging using deep-learning-based autofocusing and phase recovery," Optica **5**(6), 704–710 (2018).
23. Z. Ren, Z. Xu, and E. Y. Lam, "End-to-end deep learning framework for digital holographic reconstruction," Adv. Photonics **1**(1), 016004 (2019).
24. Y. Rivenson, Y. Zhang, H. Günaydın, D. Teng, and A. Ozcan, "Phase recovery and holographic image reconstruction using deep learning in neural networks," Light: Sci. Appl. **7**(2), 17141 (2018).
25. A. Sinha, J. Lee, S. Li, and G. Barbastathis, "Lensless computational imaging through deep learning," Optica **4**(9), 1117–1125 (2017).
26. Z. Ren, T. Zeng, and E. Y. Lam, "Digital holographic imaging via deep learning," in *OSA Topical Meeting in Computational Optical Sensing and Imaging*, (2019), p. CTu3A.4.
27. 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*, (2015), pp. 234–241.
28. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2016), pp. 770–778.
29. X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Advances in Neural Information Processing Systems*, (2016), pp. 2802–2810.
30. S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in Neural Information Processing Systems*, (2017), pp. 3856–3866.
31. Y. LeCun, "The mnist database of handwritten digits," http://yann.lecun.com/exdb/mnist/ (1998).
32. R. LaLonde and U. Bagci, "Capsules for object segmentation," arXiv preprint arXiv:1804.04241 (2018).
33. J. Rajasegaran, V. Jayasundara, S. Jayasekara, H. Jayasekara, S. Seneviratne, and R. Rodrigo, "Deepcaps: Going deeper with capsule networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2019), pp. 10725–10733.
34. A. Mobiny and H. Van Nguyen, "Fast capsnet for lung cancer screening," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (2018), pp. 741–749.
35. P. Afshar, A. Mohammadi, and K. N. Plataniotis, "Brain tumor type classification via capsule networks," in *IEEE International Conference on Image Processing (ICIP)*, (2018), pp. 3129–3133.
36. S. Mallat, "Understanding deep convolutional networks," Philos. Trans. R. Soc., A **374**(2065), 20150203 (2016).
37. E. Oyallon and S. Mallat, "Deep roto-translation scattering for object classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2015), pp. 2865–2873.
38. G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in *International Conference on Artificial Neural Networks*, (2011), pp. 44–51.
39. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980 (2014).
40. W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2016), pp. 1874–1883.
41. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," IEEE Trans. on Image Process. **13**(4), 600–612 (2004).