Optical Flow-Guided Multi-Scale Dense Network for Frame Interpolation

Ting Zhang* , Huihui Bai*, Feng Li* and Yao Zhao*
* Beijing Jiaotong University, Beijing, China
E-mail: hhbai@bjtu.edu.cn

***Abstract*—Video frame interpolation is a traditional computer vision task, which aims to generate intermediate frames between two given consecutive frames. Many algorithms attempt to solve this task relying on optical flow to compute dense pixel correspondence. According to the estimated flow, the input images are warped to the location of the interpolated frame, and then blended together to generate synthesis frame. However, due to the difficulty of flow estimation, this method always leads to blurry region and visually unpleasant results. To overcome the limitation of inaccurate flow estimation, we perform an end-to-end neural network to improve interpolation results after warping, which explicitly uses optical flow but not completely depends on it. Moreover, we design a multi-scale dense network for frame interpolation (FIMSDN), which not only makes full use of the multi-scale information for large motion frame interpolation, but also strengthens feature propagation. Specifically, a pre-trained optical flow net is firstly utilized to produce the bidirectional flow between two input frames. The input images are warped to the middle frame by the estimated flow and then fed with the original images into the FIMSDN to directly estimate the in-between frame. Experimental results show the improvement in terms of both objective and subjective quality by comparing with other recent optical flow and convolutional neural network (CNN) based methods.**

I. Introduction

Video frame interpolation, a classic subject of computer vision, aims to generate intermediate frames between given two consecutive frames in a video sequence. This technique is widely used in video frame rate conversion [7], slow motion effects [15], as well as video compression [5]. Due to the effect of motion complexity, occlusions, luminance changes and so on, synthesis in-between frames is still a challenging problem.

Many algorithms attempt to solve the frame interpolation task explicitly or inexplicitly relying on optical flow estimation. Optical flow indicates the two dimensional apparent motion field of space moving objects in image domain, which can be regarded as a sub-problem of frame interpolation. Some traditional methods [3] [10] entirely depend on the flow estimation, which generate interpolated frames through blending the warped input images based on the computed flow fields. However, the quality of interpolation frame is sensitive to the accuracy of estimated flow, which is a complex problem and suffers from the factors of occlusion, illumination changes and large motion. Lately, in order to overcome this limitation, Meyer et al. [7] propose a phase-based image synthesis method without the need for any form of explicit correspondence estimation. While their method provides an efficient alternative to traditional flow based algorithms, it only well suits for small motion interpolation.

In recent years, deep learning, especially convolutional neural networks (CNNs), have been successfully utilized in many computer vision tasks and achieved state-of-the-art results. For optical flow estimation, FlowNet [8] proves the effectiveness of CNNs in learning feature matching, and its successors FlowNet2 [18] generates more smooth and accurate flow fields than before. However, they are supervised methods which need scarce ground truth of optical flow to train. For the frame interpolation task, Long et al. [9] propose an encoder-decoder architecture to synthesis in-between frames directly, but their ultimate goal is to estimate optical flow. As a result, their synthesis results are visually blurry. In [16], flow estimation and pixel synthesis are merged into a single convolution process to generate in-between frames. However, this method needs large kernels to handle large motion. Liu et al. [15] introduce a deep voxel flow (DVF) method to learn a fully differentiable network for frame synthesis. While they design a voxel flow layer like optical flow field instead of directly estimate flow, their method is still limited in the accuracy of voxel flow, and the result of voxel flow layer is inferior to FlowNet [8].

Therefore, according to the flow estimation based frame synthesis algorithm, we propose an optical flow-guided CNN based method, which explicitly uses optical flow but not completely depends on it. Specifically, a pre-trained flow estimation network is cascaded with our frame interpolation network, which can provide more motion information and reduce the complexity of later training. In addition, to avoid the spatial distortion brought by errors in flow fields, we propose a multi-scale dense network for frame interpolation (FIMSDN), which can directly generate visually pleasant in-between frames. In the framework of FIMSDN, a multi-scale model is utilized to make full use of information from fine to coarse level, which can effectively improve the results of large motion frame interpolation. Instead of just inputting different scale images for training, we extract multi-scale perspective information from original input images to preserve high resolution information. And different from the encoder-decoder model, which consists of many sub- and up-sampling layers, we progressively improve features in each scale via a densely connected architecture. The dense connection is adopted to propagate the previous features to the current state, because convolutions only take local information into

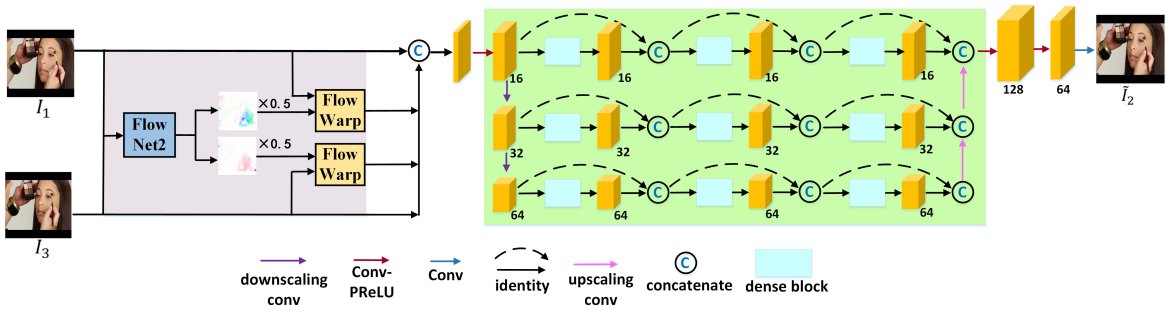


Fig. 1. Overview of the proposed method.

account. Finally, each scale features are gathered to synthesis intermediated frame.

Our method can be trained end-to-end by sampling triplets of consecutive video frames from any natural video sequences. Experimental results show the improvement in terms of both objective and subjective quality by comparing with other recent optical flow and CNN based methods on different video scenarios.

The rest of this paper is organized as follows. Section II elaborates a detailed description of the proposed scheme. The evaluation of our model and experimental results are presented in Section III. Finally, conclusions are drawn in Section IV.

II. Proposed Scheme

Our frame interpolation method is an optical flow-guided multi-scale dense network architecture, which named as OF-FIMSDN.The proposed scheme benefits from the pre-trained flow estimation network and blends the warped frames visually natural through a multi-scale densely connected network. Any video can be processed to be training samples for the task of frame interpolation. The overview of the proposed method is illustrated in Fig. 1. In this section, we will introduce the proposed method in detail.

A. Optical Flow-Guided

Taking two consecutive frames I_i and I_j as an an example, the optical flow from I_i to I_j denotes as $\Delta_{i\to j} = (\mu_x, \nu_y)$, where μ_x and ν_y represent the pixel-wise displacements in the directions of x and y, respectively. According to the flow $\Delta_{i\to j}$, a warping operation $W(\cdot)$ can be adopted to warp I_j to I_i via bilinear interpolation, which can be written as:

$$I_{j\to i}(x,y) = W(I_j, \Delta_{i\to j}) = I_j(x+\mu_x, y+\nu_y). \quad (1)$$

In our scheme, given triplets of consecutive frames I_1, I_2, and I_3, our goal is to reconstruct the in-between frame I_2 from input two frames I_1 and I_3. We first estimate the bidirectional optical flows based on a pre-trained flow net, which provides a great starting point for the following frame synthesis network. We define the computed bidirectional flow fields from I_1 to I_3 and I_3 to I_1 respectively as follows:

$$\Delta_{1\to 3} = \mathcal{F}(I_1, I_3), \Delta_{3\to 1} = \mathcal{F}(I_3, I_1). \quad (2)$$

where $\mathcal{F}$ represents the mapping function of the network for optical flow estimation. Here, we choose FlowNet2 [18] to compute initial bidirectional flows, which generates more smooth and accurate flow fields.

The optical flow provides a set of mapping functions to estimate the in-between behavior of each object from the pixels of input images. Meanwhile, bidirectional flow fields further ensure the accuracy of the unidirectional warping. We then warp the input images to the location of in-between frame as:

$$\begin{aligned} I_{1\to 2} &= W(I_1, \Delta_{3\to 1}/2), \\ I_{3\to 2} &= W(I_3, \Delta_{1\to 3}/2). \end{aligned} \quad (3)$$

B. Multi-scale Dense Network for Frame Interpolation

Though the computed flow fields can be used to synthesize the intermediate frames, the results may generate motion blur and artifacts. Therefore, we proposed a multi-scale dense network for frame interpolation (FIMSDN), which is adopted as a refined network to optimize the quality of synthesized frames. The network is trained to directly generate in-between frames, whose mapping function $g(\cdot)$ can be written as:

$$\tilde{I}_2 = g(I_1, I_3, I_{1\to 2}, I_{3\to 2}, \theta). \quad (4)$$

where θ is the network parameter. To avoid the spatial distortion and occlusions for optical flow warping, except the warped images $I_{1\to 2}$ and $I_{3\to 2}$, the original images I_1 and I_3 are also concatenated as an input to offer more image information for the frame interpolation network.

The network architecture is shown in Fig. 1. For the design of FIMSDN, we utilize the framework of multi-scale feature maps to maintain coarse to fine level features. Meanwhile, dense connection is adopted to eliminate the limitation of short-range dependencies caused by convolutions with small kernel size. Specifically, as shown in Fig. 1, in the first column, we first extract feature maps from input images using a 7×7 convolutional layer followed by a Parametric Rectified Linear Unit [22] activation (red line, denoted as Conv(7×7)-PReLU). It is noted that small kernel size is adverse to the feature extraction of large motions, and PReLU is utilized as an activation function after the convolutional layers to realize nonlinear mapping, except the last layer in our network. And then a downscaling layer (purple line) is utilized to get coarse

level feature with Conv(3×3) of stride 2. Three scale feature maps are extracted with number of 16, 32, and 64, respectively. In the horizontal dimension of each scale, three dense blocks are adopted to maintain features reuse and strengthen feature propagation through corresponding scale. The pattern of dense block is denoted as Conv(1×1)-PReLU-Conv(3×3)-PReLU. Distinct from the design of dense block in DenseNets [19], we remove the Batch Normalization (BN) [13] following the findings in [17], and use PReLU activation to realize nonlinear mapping. At last column, coarse feature maps are upscaled to the size of original image and concatenated with the fine scale features. Finally, the synthesized intermediate frame is obtained by further convolution on the fusion feature maps.

C. Loss Function

FIMSDN is designed in order to make the synthesized frame $\tilde{I}_2$ as similar as possible to the in-between frame I_2. Therefore, a corresponding optimization goal is needed for our network by minimizing a distance $\mathcal{L}_p$ (p=1 or p=2) between the original and reconstructed frame as follows:

$$\mathcal{L}_p(\tilde{I}_2, I_2) = ||\tilde{I}_2 - I_2||_p^p. \tag{5}$$

In [11], it has been reported that $\mathcal{L}_2$ loss always leads to unnatural blurriness of the output image. Meanwhile, the Charbonnier loss, a differentiable variant of $\mathcal{L}_1$ norm, is commonly used in flow estimation tasks [12] and has been confirmed can lead to a robust result than the $\mathcal{L}_2$ loss. The Charbonnier loss function is:

$$\rho(x) = \sqrt{x^2 + \epsilon^2}. \tag{6}$$

where x represents the difference between the original and synthesized frame, and ϵ is a small hyperparameter to control the Charbonnier penalty is always non-zero. Here, the loss function can be denoted as:

$$L(\tilde{I}_2, I_2) = \sqrt{||\tilde{I}_2 - I_2||^2 + \epsilon^2}. \tag{7}$$

where ϵ set to 0.01 to train our frame interpolation network.

III. Experimental Results

In this section, we first introduce training datasets and details of our network. Then, in order to evaluate the performance of our video frame interpolation scheme, we conduct experimental comparisons against others, including a few optical flow based methods [1], [10], [18], the recent phase-based interpolation method [7] and state-of-the-art deep learning based method [16].

A. Datasets

The proposed method can be trained end-to-end using any video data by sampling triplets of consecutive video frames. UCF101 [4] video dataset, which has been split into training and testing set, is used in our scheme. It contains of 101 action classes that is benefit for the network to learn motion features.

For our training, triplets are extracted by taking three consecutive frames from all videos, where the first and last frames serve as inputs to our network and the in-between image as ground truth to train. It is worth mentioned that not all triplets help the model to do frame interpolation, because they may show no or less difference between consecutive images. Therefore, we first choose frame triplets consist of obvious motion by removing the peak signal-to-noise ratio (PSNR) values less than a certain threshold between pairs in triplets group. Data augmentation is adopted to increase the diversity of samples by flipping the images vertically and horizontally. Our training set finally includes approximately 200,000 triplets. Following [15], parts of the UCF101 [4] and THUMOS-15 [2] test sets are used as benchmarks.

TABLE I
PSNR and SSIM comparisons on UCF101 test dataset

Methods	PSNR/dB		SSIM
	full	mask	
Farneback[1]	34.779	36.637	0.966
EpicFlow[10]	34.817	36.531	0.967
Phase-based[7]	34.094	36.357	0.961
FlowNet2[18]	34.523	36.178	0.964
SepConv $\mathcal{L}_f$[16]	34.908	36.939	0.968
SepConv $\mathcal{L}_1$[16]	34.946	36.987	0.969
FIMSDN(no-flow)	34.981	37.126	0.969
Proposed	**35.104**	**37.215**	**0.971**

TABLE II
PSNR and SSIM comparisons on THUMOS-15 test dataset

Methods	PSNR/dB		SSIM
	full	mask	
Farneback[1]	34.677	37.087	0.971
EpicFlow[10]	34.627	37.192	0.971
Phase-based[7]	33.596	36.354	0.959
FlowNet2[18]	33.974	36.463	0.969
SepConv $\mathcal{L}_f$[16]	35.207	37.409	0.974
SepConv $\mathcal{L}_1$[16]	**35.419**	37.766	**0.975**
FIMSDN(no-flow)	34.938	37.750	0.969
Proposed	35.089	**37.888**	**0.975**

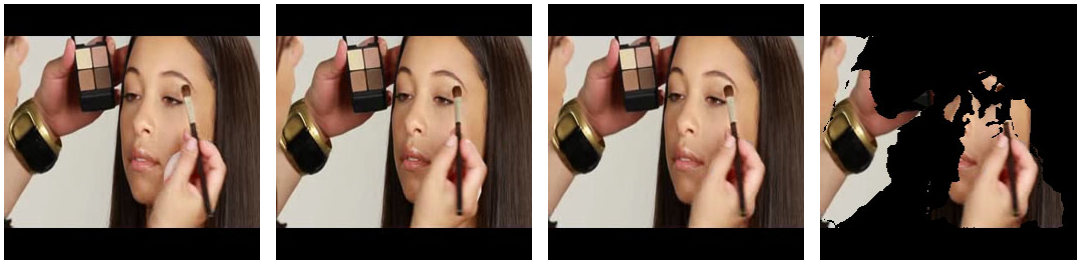
B. Training Details

To train our neural network, we initialize its parameters via the approach of Xavier [6], and then use ADAM [21] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, a learning rate of 0.0001 and 8 samples per mini-batch to minimize the loss function. The training is performed via TensorFlow [23] on 1 Titan Xp GPU.

C. Results Comparison

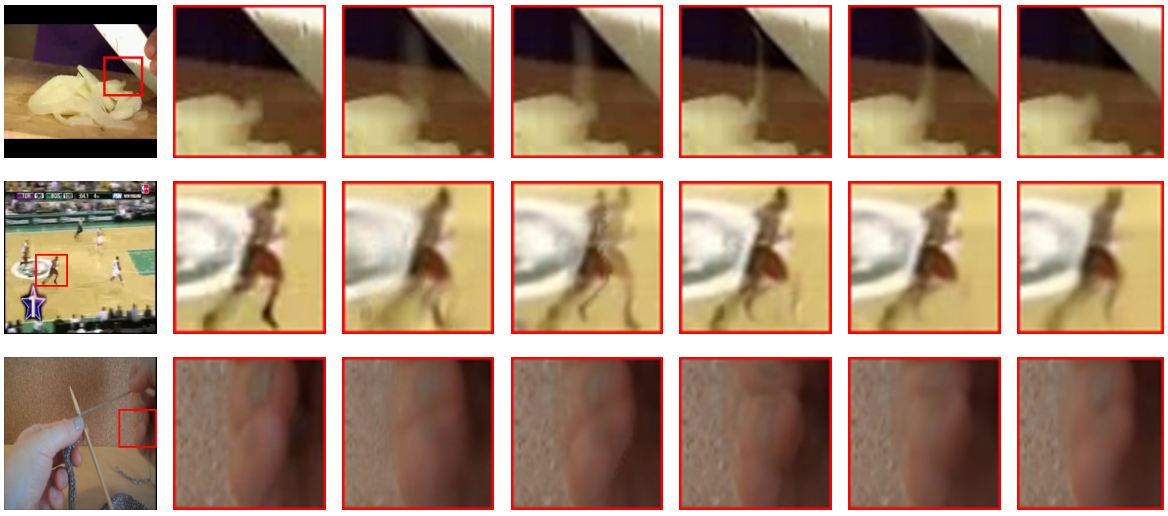
To evaluate the performance of our proposed method, we have compared with several state-of-the-art methods from the aspects of objective and visual quality. More details about the comparison experiments and results are explained in the following parts.

In order to evaluate the benefits of training based on estimated optical flow, we first feed the input frames directly into our synthesis network, which is trained as the same condition as proposed method and denoted as FIMSDN (no-flow). Several optical flow estimation methods are also served as the comparison experiments, including traditional methods Farneback [1] and EpicFlow [10], as well as the CNNs based approach FlowNet2 [18]. Given the estimated



Before Image I_1 After Image I_3 Target Image I_2 Masked Image

Fig. 2. Example of masked image for evaluation.



GT GT-crop Phase-based EpicFlow SepConv $\mathcal{L}_f$ SepConv $\mathcal{L}_1$ Proposed

Fig. 3. Visual comparisons with other frame interpolation methods, where "GT" gives the ground truth of in-between frame with highlighted crop, and "GT-crop" refers to the cropped regions.

optical flow fields, we adopt the frame interpolation algorithm introduced in the Middlebury benchmark [3] to synthesize the in-between frame. We then compare with some directly frame interpolation methods, including the recent phase-based [7] technique and a deep learning based method SepConv [16]. For the two trained models proposed in SepConv, we refer to their as SepConv $\mathcal{L}_1$ and $\mathcal{L}_f$ based on the different training loss.

For objective evaluation, the PSNR and structural similarity (SSIM) are utilized as image quality assessment metrics for interpolation accuracy. And the higher their values are, the better the frame synthesis quality is. We perform our evaluation not only on full images, but also on the masked images, as shown in Fig. 2, which only contain large motion regions in target images. Specifically, we use EpicFlow [10] to compute flow fields between pairs of triplets frame, and then choose pixels where the flow values higher than 0.2 as masked areas. The performance metrics are denoted as full and masked PSNR, respectively. The objective quality comparisons on UCF101 and THUMOS-15 test datasets are shown in Tab. I and Tab. II respectively, where the numbers of rough bodies mean the best. We can see that the proposed method outperforms other schemes in terms of both average PSNR and SSIM on UCF101 test frames. Although our method is inferior to SepConv [16] in terms of full PSNR on THUMOS-15, our average masked PSNR is better, which indicates the effectiveness of proposed method on motion estimation. Meanwhile, the results demonstrate the efficiency of optical flow-guided and the proposed frame interpolation network by comparing with methods Flownet2 [18] and FIMSDN(no-flow), respectively.

Examples of visual comparisons are exhibited in Fig. 3, where some motion parts are spotlighted by red rectangles. It is worth to mention that optical flow based methods easily generate blur and artifacts in results owing to the inaccurate flow estimation and blend algorithm. The proposed method shows more precise and visual pleasing interpolation results.

IV. Conclusions

In this paper, we propose an optical flow-guided frame interpolation method, which explicitly uses optical flow but not completely depends on it. The pre-trained flow fields provide a great starting point for latter training, and the following

frame synthesis network, namely FIMSDN, blends the warped frames visually natural. Meanwhile, in the framework of FIMSDN, a multi-scale model is utilized to make full use of information from fine to coarse level, which can effectively improve the results of large motion frame interpolation. And dense connection is adopted to strengthen feature reuse and propagation. Experimental results demonstrate that the proposed scheme achieves better performance in both objective and subjective quality by comparing with other recent optical flow and CNN based frame interpolation methods.

ACKNOWLEDGMENT

This work was supported in part by Key Innovation Team of Shanxi 1331 Project (KITSX1331) and Fundamental Research Funds for the Central Universities (2018JBZ001).

REFERENCES

[1] G. Farnebck, "Two-Frame Motion Estimation Based on Polynomial Expansion," in *Scandinavian Conference on Image Analysis,* vol. 2749, pp. 363-370, 2003.

[2] A. Gorban, H. Idrees, Y. G. Jiang, A. Roshan Zamir, I. Laptev, M. Shah, and R. Sukthankar, "THUMOS challenge: Action recognition with a large number of classes," 2015.

[3] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," in *International Journal of Computer Vision,* vol. 92, no. 1, pp. 1-31, 2011.

[4] K. Soomro, A. Roshan Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," in *Center for Research in Computer Vision,* 2012.

[5] Z. Liu, L. Yuan, X. Tang, M. Uyttendaele, and J. Sun, "Fast burst images denoising," *Acm Transactions on Graphics,* vol. 33, no. 6, pp. 1-9, 2014.

[6] X. Glorot, Y. Bengio., "Understanding the difficulty of training deep feedforward neural networks," *Journal of Machine Learning Research,* vol. 9, pp. 249-256, 2010.

[7] S. Meyer, O. Wang, H. Zimmer, M. Grosse, and A. Sorkine-Hornung, "Phase-based frame interpolation for video," in *IEEE Conference on Computer Vision and Pattern Recognition,* pp. 1410-1418, 2015.

[8] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *International Journal of Computer Vision,* pp. 2758-2766, 2015.

[9] G. Long, L. Kneip, J. M. Alvarez, H. Li, X. Zhang, and Q. Yu, "Learning image matching by simply watching video," in *European Conference on Computer Vision,* pp. 434-450, 2016.

[10] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Epicflow: Edge-preserving interpolation of correspondences for optical flow," in *IEEE Conference on Computer Vision and Pattern Recognition,* pp. 1164-1172, 2015.

[11] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," in *International Conference on Learning Representations,* 2016.

[12] P. Krhenbhl, V. Koltun, "Efficient Nonlocal Regularization for Optical Flow," in *European Conference on Computer Vision,* vol. 7572, pp. 356-369, 2012.

[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning,* vol. 37, pp. 448-456, 2015.

[14] L. Zhao, H. Bai, A. Wang, and Y. Zhao, "Learning a Virtual Codec Based on Deep Convolutional Neural Network to Compress Image," *arXiv preprint arXiv:1712.05969,* 2017.

[15] Z. Liu, R. Yeh, X. Tang, Y. Liu, and A. Agarwala, "Video frame synthesis using deep voxel flow," in *IEEE Conference on Computer Vision,* pp. 4473-4481, 2017.

[16] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive separable convolution," in *IEEE International Conference on Computer Vision,* pp. 2270-2279, 2017.

[17] S. Nah, T. H. Kim, and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *IEEE Conference on Computer Vision and Pattern Recognition,* pp. 257-265, 2016.

[18] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *IEEE Conference on Computer Vision and Pattern Recognition,* pp. 1647-1655, 2017.

[19] G. Huang, Z. Liu, K. Weinberger, and L. Maaten, "Densely connected convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition,* 2016.

[20] L. Zhao, H. Bai, A. Wang, and Y. Zhao, "Multiple description convolutional neural networks for image compression," *arXiv preprint arXiv:1801.06611,* 2018.

[21] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations,* 2015.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human level performance on ImageNet classification," in *IEEE Conference on Computer Vision,* pp. 1026-1034, 2015.

[23] M. Abadi, A. Agarwal, P. Barham, and et al., "Tensorflow: Large-Scale machine learning on heterogeneous distributed systems," *arXiv: 1603.04467,* 2016.

[24] G. Huang, D. Chen, T. Li, and et al., "Multi-Scale Dense Networks for Resource Efficient Image Classification," *arXiv preprint arXiv: 1703.09844,* 2017.

[25] L. Zhao, H. Bai, A. Wang, and Y. Zhao, "Simultaneously color-depth super-resolution with conditional generative adversarial network," *arXiv preprint arXiv:1708.09105,* 2017.