A Survey on Image Super Resolution and Colorization

Berat Biçer
Computer Engineering
Bilkent University, Ankara, Turkey
berat.bicer@bilkent.edu.tr

Ergün Batuhan Kaynak
Computer Engineering
Bilkent University, Ankara, Turkey
batuhan.kaynak@bilkent.edu.tr

I. Introduction

This survey presents a literature survey for two computer vision problems in context of machine learning: image colorization and super resolution, and joint study of the two. Due to the similarities between the tasks (i.e. training a network to obtain an image that is hopefully similar to a ground truth image), the same evaluation and quality assessment metrics are leveraged. We explain these metrics in Section II during colorization discussion, and refer to them in Section III in super resolution.

II. Image Colorization

Image colorization (IC) is the task of obtaining a colored image from a source image. Source images can be grayscale, or be subjected to degradation and color loss. In literature, IC is mostly studied as a separate task yet some researchers tackled IC in relation to other restoration problems as well. The remaining of this section is split into two parts: reference colorization and automatic colorization.

A. Reference (exemplar-based) Colorization

Reference colorization has a reference (Ref) that the network seeks to learn from the input. This method usually selects a colored image as Ref, converts it into a grayscale image and, feeds the pair into the network wherecitee network is designed to learn the reference image. The problem is basically formulated as a regression task, where the network learns a parametric mapping between the input grayscale and the RGB counterpart [1]. To this end, networks generally choose mean squared error (MSE) as the loss function:

$$MSE = \frac{1}{P}\sum_{i=1}^{P}(Ref(i) - Output(i))^2 \tag{1}$$

In some studies, researchers experimented with the reference image and its effect on what the network learns. Some architectures leveraged user inputs as reference images [2] and experimented with various reference images [3]. Both studies have shown that the output image is heavily affected by the high-level similarity between the input and the reference image (such as object boundaries) and the color scheme of the reference image. Some studies employed text input as reference [4] as well. It is clear that there is no single way of referencing colorization procedure, and the selection of the reference input remains as one of the challenges for reference colorization.

Another challenge is the evaluation of network outputs. Most research such as [5]–[9] uses inspection and perceptual quality, basically displaying comparative results obtained from various networks and their outputs, drawing conclusions from apparent differences. This approach is simple, yet lacks objectivity. A more robust method of evaluation includes one of two following metrics: peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM). PSNR is the ratio between the maximum power of a signal and the power of the corrupting noise affecting the fidelity of the representation. In dB, given an image I and its approximation K, it is computed as follows:

$$PSNR = 10 \times log_{10}\frac{{MAX_I}^2}{MSE} \tag{2}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(I(i) - K(i))^2 \tag{3}$$

$$MAX_I = max_{i=1...n}I(i) \tag{4}$$

The higher PSNR index the better the obtained representation. PSNR is sometimes said to fall short as a metric because it lacks the ability to asses human perception of the reconstruction, but it is still very commonly used.

Another metric called structural similarity index (SSIM) is proposed to solve the perception criterion problem [10]. SSIM improves upon PSNR and MSE and is computed on sampled windows from two images. SSIM also takes luminance, contrast, and structures into consideration. These metrics are individually calculated using I and K and then multiplied to obtain a decimal value in range $[-1, 1]$ where -1 indicates no similarity and 1 indicates identical sets of data, or a perfect match. The reader is kindly directed to the original paper for details on the individual calculations.

The next challenge is the network design. While there are methods employing conventional computer vision techniques [11], we will focus on machine learning applications on image colorization. Earlier studies such as [2], [6], [7] focus on encoder-decoder based architectures: First part of network uses convolutions and maxpooling, reducing the spatial size of the input while increasing number of channels in representation. This way, the network learns a small-sized representation of common trends in the data which can be used in classification, or can be fed into another network to obtain samples resembling the input. Decoder network does the latter, converting the encoded representation into images having the same size as the input. The term upsampling is significant in decoder networks, and corresponds to increasing spatial dimensions of the input. Traditionally, this is done with interpolation whereas recent approaches use transpose convolutions. Upsampling is explained in detail in Section III-A of this survey.

Encoder-decoder networks are relatively successful, reaching to PSNR index 22-38 [2] in some datasets. However, a major breakthrough in image colorization occurred with generative adversarial networks(GAN) and their applications in image colorization. A GAN consists of two networks competing against each other: a generative network which produces samples based on input, and a discriminator trained to detect generated samples from the real ones. The network is jointly trained to minimize discriminator loss and maximize generator performance. Generators can be designed as an encoder-decoder network as in [12], where authors applied colorization to black and white sketch images to obtain an auto painter, utilizing a generator designed as an encoder-decoder network and discriminator as an encoder vector followed by dense layers for classification. Some studies also formulate reference colorization using a non-grayscale input such as [13] where the inputs are underwater images having blue-green color scheme overall. It is possible to employ design choices from earlier neural networks in GANs as well: In [14], authors connect two sets of generator-discriminator pairs in a coupled fashion such that generators and discriminators share the same weights pairwise; allowing the network to learn a joint representation for multiple image domains(datasets). In [15], authors apply skip connections to the generator as well, achieving significant improvements over the state of the art.

B. Non-exemplar Based Image Colorization

Non-exemplar based image colorization (NEIC) has no reference to learn from and seeks to obtain a believable colorized output using unsupervised learning algorithms. These methods usually involve generative adversarial networks (GAN) as described previously. However, since there is no reference to compare to, the network is trained to optimize the joint adversarial loss with the expectation that the quality of the output increases as training progresses. Most studies employ the following joint adversarial loss or its variations introduced in [16]:

$$\begin{aligned}\min_G \max_D V(D,G) &= \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] \\ &\quad + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]\end{aligned} \tag{5}$$

where $G(z)$ is the generator output and $D(x)$ is the discriminator output. Compared to exemplar-based GAN approaches, most non-exemplar methods optimize only the adversarial loss whereas exemplar-based approaches optimize the joint loss of generative loss and reference loss in (1).

Since there is no reference image to narrow down the search space, non-exemplar image colorization is an ill-defined problem since output has a large degree of freedom. Recent studies in [2], [17] aim to solve this problem by combining limited user interaction with the automatic coloring algorithm, however, this is still an active field of research.

C. Datasets

It is relatively easy to obtain a dataset for IC task, since training samples can be obtained by computationally converting color images into grayscale. Most influential factor in colorization performance is the size of the dataset, which most research easily handles by using a large-scale dataset for other fields such as object recognition (ImageNet [18]), or segmentation and scene understanding (MSCOCO [19]), or super resolution (DIV2K [20]–[23]).

III. Image Super Resolution

Image Super Resolution (SR) is the task of improving the resolution of low resolution (LR) images and converting them to high resolution (HR) images. With SR, challenging data can be converted to a more interpretable resolution and this can help in domains such as medical imaging and security/surveillance. Entertainment venues such as games and movies can enjoy quality increases. SR tasks often declare a magnification amount Mx that dictates the factor of scale difference between the LR and HR images (e.g. a 4x SR application can try to convert 25x25 LR images to 100x100 HR images). Most common magnification amounts are 2x and 4x, although a 2020 paper by Bühler et. al. attempted as much as 32x [24]. Super resolution is also used for videos, but we only discuss images within the scope of this survey.

It is relatively trivial to obtain datasets for training in SR task, since any image can be labeled HR, and then downsampled to acquire LR counterparts. Very common datasets for training include ImageNet, MSCOCO and DIV2K among many others. The first two datasets are desirable since they contain many images (350K and 14M respectively) and the DIV2K dataset contains a modest 1000 images, but they are nearly 2K quality in resolution. Most papers evaluate on some agreed upon datasets, some of which are listed in Table I.

Pixelwise mean squared error (MSE) between the reconstructed image SR (obtained from the source low resolution image LR) and the ground truth high resolution image HR (both containing P pixels) is a historically common evaluation and quality assessment metric for SR tasks (Eq. 3). More recently, the use of peak signal-to-noise ratio (PSNR) as an evaluation metric became more common (Eq. 2). Recent papers have also started including SSIM as a metric.

SR is an old task with first applications as early as 1980s with basic interpolation techniques. Most notable improvements are made fairly recently starting from around 2016 with the help of deep learning. Another jump in performance is obtained with generative adversarial networks (GAN) and the

TABLE I

Table of PSNR values of different networks and datasets. We only consider 4x magnification networks here. The networks are mostly in chronological order from left to right columnwise, so we can expect an increase in PSNR from left to right. Data collected from individual papers. First number in each column is the PSNR value, while the other is the SSIM value.

		Bicubic	SRCNN	DRCN	SRDenseNet	SRGAN	ESRGAN
Datasets	Set5	28.43 / 0.811	30.48 / 0.863	31.53 / 0.885	32.02 / 0.893	32.05 / 0.891	**32.73 / 0.901**
	Set14	26.00 / 0.702	27.50 / 0.751	28.03 / 0.767	28.50 / 0.778	28.53 / 0.780	**28.99 / 0.792**
	BSD100	25.96 / 0.668	26.90 / 0.710	27.24 / 0.723	27.53 / 0.734	27.57 / 0.735	**27.85 / 0.746**
	Urban100	23.14 / 0.657	24.52 / 0.723	25.14 / 0.751	26.05 / 0.782	26.07 / 0.784	**27.03 / 0.815**
	DIV2K	28.11 / 0.775	29.33 / 0.809	29.83 / 0.823	- / -	**28.92 / 0.896**	- / -
	Manga109	25.15 / 0.789	27.66 / 0.858	28.97 / 0.886	- / -	- / -	**31.66 / 0.920**
Up-Sampling Method		Bicubic	Bicubic	Bicubic	TConv	Sub-Pixel	Sub-Pixel
Up-Sampling Location		Pre	Pre	Pre	Post	Post	Post

use of residual networks. Due to the abundance of methods in super resolution task, it is not feasible to touch upon all of them in this survey. Instead, we will focus on a smaller subset of the network architectures and present a comparative outlook to the advantages and disadvantages of these networks, and briefly introduce them to the reader. We focus on two main topics. First, we talk about the up-sampling methods themselves and then move on to the different points in the network where up-sampling operations can take place. We choose focusing on these because they are specifically related to SR problem, not general computer vision. It should still be noted that a lot of parts of these networks are always changing and these improvements cannot only be ascribed to a change in specific operations only, but it is the only comparison we can make by looking at the literature without ablation studies.

A. Up-Sampling Methods

Upsampling methods are studied in two different categories: One of them is the classical interpolation based methods. Most notable example here is the bicubic interpolation. The second school of methods is the more recent, learning based techniques.

1) Interpolation Based Up-Sampling: We briefly discuss the historical approaches here for the sake of completeness. Among the methods here bicubic is the most relevant. That said, its relevancy is mostly to be used as a baseline by the never approaches.

a) Nearest Neighbour: For every new pixel to be predicted in the SR image, this algorithm looks at the neighbors of the pixel (in the LR image) within a window picks the most common pixel as its result. LR images are usually low on pixel count and therefore it is easy to see individual pixels and perceive a blocky structure. Although we obtain a higher resolution image with nearest neighbors algorithm, we cannot remedy the blockiness.

b) Bilinear interpolation: Linear interpolation is widely used in mathematics to approximate unknown values. It is done by drawing a line between two known data points (x_0, y_0) and (x_1, y_1) and assigning unknown points to be on the line. Since a line is only a first degree polynomial, it has very bad performance. Bilinear interpolation is the two dimensional counterpart of linear interpolation (i.e. linear interpolation of two variables).

c) Bicubic interpolation: Cubic interpolation (also referred as cubic Hermite interpolation) tries to fit a third degree polynomial spline. Following the same logic as in bilinear interpolation, bicubic interpolation is the two dimensional version of cubic interpolation. Bicubic interpolation performs the best among interpolation methods. One of the first SR networks, SRCNN, used bicubic interpolated LR image and tried to reconstruct the SR from it directly [25]. The reconstructions are done via convolutional layers. Later, DRCN improves upon SRCNN with recursive convolutions and becomes the best performing bicubic up-sampling network until EnhanceNet. Among recent works, EnhanceNet combines the bicubic interpolated image with the residual of the image they obtained in a 10 layer fully convolutional network [26]. We can see that bicubic interpolation is not used by itself as a method, but rather a starting template or a preprocessing method.

2) Learning Based Up-Sampling: The problem with interpolation methods was that although we were increasing our resolution, we were not able to approximate non-existent information in the image to obtain smooth results. Learning based methods hope to alleviate this problem by also considering the vast amount of previous examples they have seen to make better and more complex pixel assignments.

a) Transposed Convolution: **Transposed convolution** (also referred as fractionally strided convolutions or up convolution) can be thought of as the inverse of yje convolution operation. This layer is used to increase image dimensions and learns filters that will predict the newly created empty pixels. For a 2x increase in resolution, the source pixels of the feature map is first *expanded* to create new empty pixels. Then, instead of multiplying a group of pixels with the kernel to obtain a single value, we multiply the input group of pixels with the kernel for each empty pixel in that group (overlapping values are added together). More recent architectures mostly follow the transposed convolution (TConv) layer by another convolution to obtain an intermediate representation rather than stacking TConv layers on top of each other. One of the first promising network to use TConv is SRDenseNet, where they improve the state of the art over the best performing bicubic network, DRCN [27]. SRDenseNet also uses dense connection leading up to the TConv operation, meaning that each convolution operation produces its output by partially considering the outputs of all preivous convolutional layers. TConv by itself did not result in groundbreaking improvement, but it paved the way for leaving pre up-sampling in favor of post up-sampling (more details on section III-B).

b) Sub-Pixel: **Sub-Pixel** convolution, proposed by Shi et. al. [28] and most notably used by Ledig et. al. [29], is the most common up-sampling method used in image super resolution. The idea is fundamentally the same as TConv, but up-sampling is handled differently. TConv does upsampling and fills the missing pixels directly with convolutions. Sub-Pixel, on the other hand, applies convolution to the $H \times W \times C$ input pixels to obtain r^2 convolution outputs (for up-sampling scale factor r). These outputs are then processed through what they call a **shuffle** operation to turn $H \times W \times r^2 C$ to $rH \times rW \times C$. The inventors of Sub-Pixel prove that their shuffle operation reshapes the feature maps in a way that no information is lost, unlike TConv [30]. The reader is referred to Shi et. al. for the in-depth formulation of shuffle operation [28]. One of the first high performance architectures that use Sub-Pixel idea is SRResNet [29] (and also SRGAN in the same paper), where they create a dense representation of LR image by passing it through several residual blocks and use two Sub-Pixel layers to up-sample their representation by 4x, and pass this through a 9x9 convolutional layer for fine tuning the up-sampled result. Another impressive set of results with Sub-Pixel is obtained by ESRGAN [31]. They improve upon SRGAN by using a deeper model. They can do so by replacing the batch normalization layers by dense connections between the residual blocks, and using relativistic average GAN rather than vanilla GAN. ESRGAN also earned quite a popularity with a couple of old game graphics improvement projects.

B. Up-Sampling Locations

1) Pre Up-Sampling: Pre up-sampling is the class of up-sampling operations where the input LR image is up-sampled to a primitive SR image in the first couple of layers in the network. The rest of the network tries to restore this primitive SR to a good HR candidate. We have already mentioned examples of these when discussing bicubic interpolation [25], [26]. Note that these examples of pre up-sampling were not learnable. One network to use a learnable and still considered to use pre up-sampling is DSRN [32], but we believe this network should be considered under iterative up and down sampling. Pre up-sampling was done because the up-sampling operation itself was not learnable. Almost all learnable up-sampling methods do not use this anymore.

2) Post Up-Sampling: In contrast to pre up-sampling, post up-sampling puts the up-sampling operation to the very end of the architecture. This method is by far the most common location to do up-sampling within the literature. Up-sampling is mostly done by learnable Tconv or Sub-Pixel layers. The networks first learn a latent representation of the LR image and then convert this representation to SR via a learnable post up-sampling operation. It makes sense to learn the representation in lower dimensions and then use up- in the end, since the spatial complexity is much more reduced before up-sampling. Most architectures stack these post up-sampling operations to increase magnification rather than relying on a single layer to learn the whole magnification. With this logic, most 4x networks use two post up-sampling layers [27], [29].

3) Iterative Up and Down Sampling: The idea behind this network scheme is to incorporate the transition between HR and LR images within the network and capture the dependencies between them. Harris et. al. uses this idea as blocks within the network and treats each output SR as a residual. These residuals are then used to construct the final output SR image [33]. Although there are not many other networks with exactly this behaviour (to our knowledge), there exist recursive networks that show similar behaviour. DSRN is a dual-state recurrent network, where one state keeps the history of input LR image and the other state holds the information of the predicted SR image at timestep t. At each timestep, an SR_t is obtained by applying transposed convolution to LR_t. To obtain LR_{t+1}, SR_t is downsampled via a convolution operation [32].

C. Noteable Mentions

Here, we briefly discuss papers that we enjoyed due to their ideas, but could not include due to the scope of our comparative settings.

1) Variational Degradation: Most SR applications try to predict HR from an LR image with a single magnification ratio or donwsampling method and train different networks for different combinations of these. Xu et. al. add another dimension to their convolutional layers to also learn the best refinements for different donwsampling operations [34].

2) Progressive Up-Sampling: This is another up-sampling location we could have mentioned. One problem with the most common post up-sampling location is that we force the whole up-sampling operation to happen in a single sequential layer group. This can make the learning process harder. Instead of doing the up-sampling at a certain point, these networks have *up-sampling blocks* that are mini networks within themselves, and they try to focus on a small scale magnification. In theory, connecting these blocks can achieve better learning for big magnifications. LapSRN used this idea with bicubic up-sampling at the end of each up-sampling block [35]. ProSR then improves over LapSRN by also considering the intermediate outputs of these blocks on each subsequent block to increase training stability [36]. Progressive Up-Sampling proves to be a good idea theoretically, but in practice, the networks get too dense due to stacked blocks, and training gets less and less stable. That said, MS-LapSRN (improved version of LapSRN by the same authors) can achieve up to 32x SR results [37].

IV. Joint Image Colorization and Super Resolution

Joint study of these two problems is an active area of research. To our knowledge, the first and only study to jointly achieve super-resolution and colorization goals, published in 2019, is [38], compared to previous work which studied the problems individually and separately. Authors designed a GAN-based network where generator consists of multiple subnetwork cascaded together. The generator works as follows: low resolution input samples(color and depth images) are fed into two feature extraction subnetworks (color and depth feature extraction) and outputs are fed into a feature merge subnetwork, obtaining two sets of features. Next, fused features and individual outputs of feature extraction subnetworks are fed into separate reconstruction networks (depth and color) to obtain output color and depth images.

References

[1] M. Sharma, M. Makwana, A. Upadhyay, A. Pratap Singh, A. Badhwar, A. Trivedi, A. Saini, and S. Chaudhury, "Robust image colorization using self attention based progressive generative adversarial network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[2] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros, "Real-time user-guided image colorization with learned deep priors," *arXiv preprint arXiv:1705.02999*, 2017.

[3] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *European conference on computer vision*. Springer, 2016, pp. 649–666.

[4] H. Bahng, S. Yoo, W. Cho, D. Keetae Park, Z. Wu, X. Ma, and J. Choo, "Coloring with words: Guiding image colorization through text-based palette generation," in *Proceedings of the european conference on computer vision (eccv)*, 2018, pp. 431–447.

[5] S. Gu, R. Timofte, and R. Zhang, "Ntire 2019 challenge on image colorization: Report," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[6] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification," *ACM Transactions on Graphics (ToG)*, vol. 35, no. 4, pp. 1–11, 2016.

[7] A. Deshpande, J. Lu, M.-C. Yeh, M. Jin Chong, and D. Forsyth, "Learning diverse image colorization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6837–6845.

[8] B. Li, Y.-K. Lai, and P. L. Rosin, "Example-based image colorization via automatic feature selection and fusion," *Neurocomputing*, vol. 266, pp. 687–698, 2017.

[9] T. Mouzon, F. Pierre, and M.-O. Berger, "Joint cnn and variational model for fully-automatic image colorization," in *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer, 2019, pp. 535–546.

[10] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[11] B. Li, Y.-K. Lai, M. John, and P. L. Rosin, "Automatic example-based image colorization using location-aware cross-scale matching," *IEEE Transactions on Image Processing*, vol. 28, no. 9, pp. 4606–4619, 2019.

[12] Y. Liu, Z. Qin, Z. Luo, and H. Wang, "Auto-painter: Cartoon image generation from sketch by using conditional generative adversarial networks," *arXiv preprint arXiv:1705.01908*, 2017.

[13] C. Fabbri, M. J. Islam, and J. Sattar, "Enhancing underwater imagery using generative adversarial networks," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7159–7165.

[14] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Advances in neural information processing systems*, 2016, pp. 469–477.

[15] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[17] M. He, D. Chen, J. Liao, P. V. Sander, and L. Yuan, "Deep exemplar-based colorization," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–16, 2018.

[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[20] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 126–135.

[21] R. Timofte, S. Gu, J. Wu, and L. Van Gool, "Ntire 2018 challenge on single image super-resolution: Methods and results," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 852–863.

[22] J. Cai, S. Gu, R. Timofte, and L. Zhang, "Ntire 2019 challenge on real image super-resolution: Methods and results," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[23] S. Gu, R. Timofte, and R. Zhang, "Ntire 2019 challenge on image colorization: Report," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[24] M. C. Bühler, A. Romero, and R. Timofte, "Deepsee: Deep disentangled semantic explorative extreme super-resolution," 2020.

[25] 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.

[26] M. S. M. Sajjadi, B. Schölkopf, and M. Hirsch, "Enhancenet: Single image super-resolution through automated texture synthesis," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 4501–4510.

[27] T. Tong, G. Li, X. Liu, and Q. Gao, "Image super-resolution using dense skip connections," in *ICCV*, 2017.

[28] 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 *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1874–1883.

[29] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 105–114.

[30] W. Shi, J. Caballero, L. Theis, F. Huszar, A. Aitken, C. Ledig, and Z. Wang, "Is the deconvolution layer the same as a convolutional layer?" 09 2016.

[31] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *The European Conference on Computer Vision Workshops (ECCVW)*, September 2018.

[32] W. Han, S. Chang, D. Liu, M. Yu, M. Witbrock, and T. Huang, "Image super-resolution via dual-state recurrent networks," 06 2018, pp. 1654–1663.

[33] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," 2018.

[34] Y.-S. Xu, S.-Y. R. Tseng, Y. Tseng, H.-K. Kuo, and Y.-M. Tsai, "Unified dynamic convolutional network for super-resolution with variational degradations," 2020.

[35] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[36] Y. Wang, F. Perazzi, B. McWilliams, A. Sorkine-Hornung, O. Sorkine-Hornung, and C. Schroers, "A fully progressive approach to single-image super-resolution," 2018.

[37] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Fast and accurate image super-resolution with deep laplacian pyramid networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[38] L. Zhao, H. Bai, J. Liang, B. Zeng, A. Wang, and Y. Zhao, "Simultaneous color-depth super-resolution with conditional generative adversarial networks," *Pattern Recognition*, vol. 88, pp. 356–369, 2019.