Face Image Super-Resolution Using Inception Residual Network and GAN Framework

Septian Dwi Indradi
School of Computing
Telkom University
Bandung, Indonesia
septiandrd@gmail.com

Anditya Arifianto
School of Computing
Telkom University
Bandung, Indonesia
anditya@telkomuniversity.ac.id

Kurniawan Nur Ramadhani
School of Computing
Telkom University
Bandung, Indonesia
kurniawannr@telkomuniversity.ac.id

***Abstract*—Single Image Super-Resolution (SISR) is an image reconstruction technique that aims to generate a high-resolution image from a low-resolution image. One of the SISR implementations is to reconstruct face images in order to gain more facial information from a low-resolution face images. In this paper, we propose a method to reconstruct face images using a Generative Adversarial Network (GAN) framework that able to generate plausible high-resolution images. Inside the GAN framework, we use inception residual network to improve the generated image quality and stabilize the training. Experimental results demonstrated that our proposed method was able to generate visually pleasant face images with the highest PSNR score of 26.615 and SSIM score of 0.8461.**

Keywords—Single Image Super-Resolution, image reconstruction, face image, Generative Adversarial Network.

I. INTRODUCTION

Image reconstruction is a computer vision problem that has been widely discussed and remains an active research. Single-Image Super-Resolution (SISR) is one of the image reconstruction techniques that aims to generate a high-resolution yet plausible output from a low-resolution input. One of the SISR implementations is to reconstruct face images. In some case like surveillance camera footage, if the camera is located too far from a person or the person is under fast motion, the facial area captured often has a bad quality and become difficult to recognize. Therefore, an image reconstruction system is needed to gain detailed information of the image.

The main challenge from this problem is to generate a plausible face image that has realistic details. From the previous researches, earlier SISR methods such as sparsity-based SISR [1] and regression-based SISR [2] was able to generate high-resolution images, but some of the methods still resulting either unrealistic artefacts or over-smooth textures.

Recently, Convolutional Neural Network (CNN) based methods [3, 4] achieved significant improvements over the previous methods, and after Ian Goodfellow *et al.* introduced the Generative Adversarial Network (GAN) in 2014 [5], the combination of deep CNN and GAN framework was proven able to generate better and realistic high-resolution images [6, 7]. In this paper, we used the GAN framework, that consist of generator network and discriminator network trained simultaneously and compete with each other during the training process.

Then we proposed an inception residual block that we used inside the generator network to improve generated image quality. We also apply several loss function besides the main adversarial loss function from the original GAN framework, such as pixel-wise loss, feature-wise loss and PSNR loss in order to gain more realistic details and prevent unnecessary artefacts on the generated images.

II. RELATED WORKS

A. Single-Image Super-Resolution

The target of SISR is to reconstruct a low-resolution image and generate high-resolution version of the image that has more details and high frequency components. The methods used to solve the problem can be categorized into three different approaches. The first one is interpolation-based, which is the simplest technique that apply interpolation on the input images, but this approach can not obtain high-frequency details. The second one is reconstruction-based, that apply smoothing and down-sampling to the high-resolution images, and use that knowledge to generate high-resolution images. For some cases, the generated images often have quite a lot of noises and artefacts. The third approach is example-based, the most successful approach that use machine learning models to reconstruct low-resolution images. Earlier example-based methods *[8, 9]* learn the corelations between low-resolution images and its high-resolution complement. As this approach grows rapidly, many other methods has been developed such as sparsity-based *[1]* that uses sparse coding to learn low and high-resolution image dictionaries, and regression based methods *[2]* that uses the regression techniques for better and faster image reconstruction.

After deep learning became popular, many researcher start to implement CNN based methods for SISR problem. The first deep CNN method for SISR problem known as Super-Resolution Convolutional Neural Network (SRCNN) introduced by Dong *et al.* [10] in 2014 and achieved significant improvements over earlier methods. Nowadays, many CNN architectures are introduced to further improve the reconstructed image quality and gain better training performance.

B. Convolutional Neural Network

CNN has been very popular since the winning of 2012 ImageNet competition by Krizhevsky *et al.* [11] with their network "AlexNet". Since then CNN became a widely used method for image and video classification [12, 13], object detection [14], object segmentation [15], face recognition [16] and human pose estimation [17]. There are several factor that affect the growth of CNN such as: (i) powerful GPUs for the fefficiency of training progress, (ii) proposal of better model regularizations, and (iii) availability of dataset (for example ImageNet [18]) for training larger models.

978-1-5386-8052-0/19/$31.00 ©2019 IEEE

For SISR problem, recent studies has implemented variations of deep CNN architectures. In 2016 Dong *et al*. [3] introduced Fast Super-Resolution Convolutional Neural Network (FSRCNN), accelerating their previous model SRCNN [10] performance by propose a compact hourglass-shaped CNN and achieved faster and better system, succeeded to speed up the training more than 40 times while still has superior restoration quality. Some other advanced architectures such as deeply-recursive CNN (DRCN) by Kim *et al*. [19], succeeded to gain better results while keeping minimum parameters, also combined recursive-supervision and skip-connection to the training process.

C. *Generative Adversarial Network*

Goodfellow *et al*. [5] first introduced the GAN framework that applies a new training procedure for generative models to generate plausible and realistic images starting from random noise. The GAN framework consist of generator and discriminator which compete with each other during a simultaneous training process. The discriminator is trained to differentiate real and fake images, whereas the generator is trained to generate images resembles to the ground-truth in which the discriminator cannot recognize. Ledig *et al*. [6] implemented the GAN framework for SISR and proposed a perceptual loss function which consist of content loss and adversarial loss. Adversarial loss force the generated image close to the real image using discriminator network which trained to learn the difference between generated images from the generator network and real natural images, and the content loss which tend to recover realistic textures motivated by perceptual similarity between generated and natural images.

III. PROPOSED METHOD

A. *Network Architecture*

For both generator and discriminator network, we use deep CNN architecture, and apply inception and residual concept for the generator.

Generator Network. The general architecture of the generator network is shown in Fig.1. The network scheme was inspired by the InceptionResNetV2 architecture by Szegedy *et al*. [20]. It takes 16 x 16 pixel degraded low-resolution images as the input. First, the stem block will take the input images and extract its low level features, followed by the inception-residual and upscaling block alternately. At the end of the network we stack 3 convolution layers to build up the features back into image form. We use ReLU activation function for the convolutions within the network except the last convolution layer we use hyper-tangent function. The detailed scheme of stem, inception-residual and upscaling block can be seen in Fig. 2.

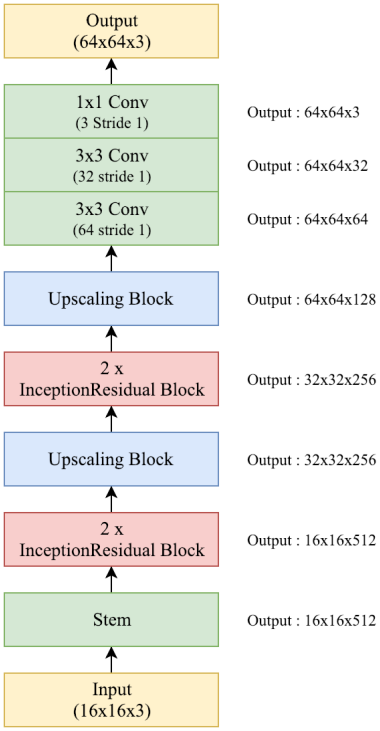
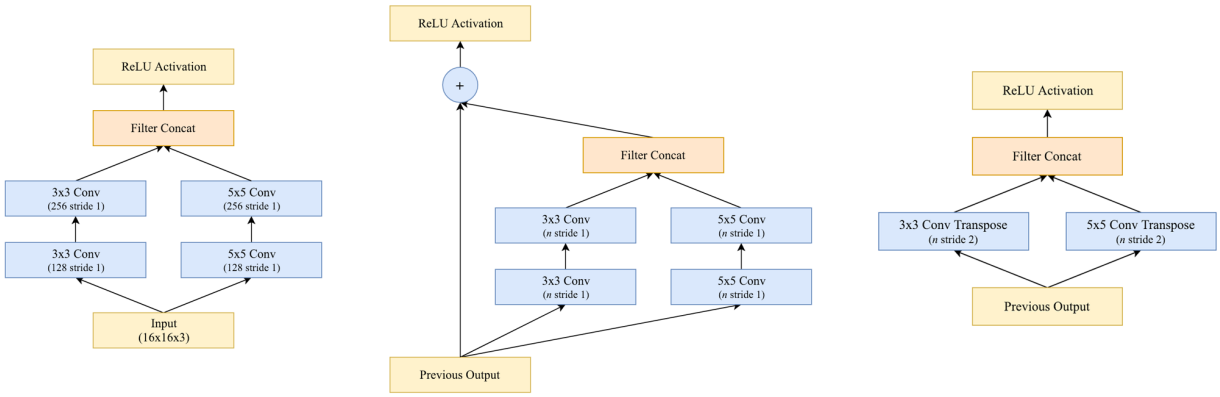


Fig. 1. Generator network scheme.

The inception-residual block depicted in Fig. 2(b) also inspired from InceptionResNetV2 architecture [20]. The inception concept first introduced in 2015 by Szegedy *et al*. [21] increasing the network sparsity, allowing the network to extract visual information at different scales by passing the input into different convolutions at the same time and aggregate them so the following layer can abstract features from different scales simultaneously. Then we implemented the residual connection introduced by He *et al*. [22] to accelerate the training performance.



(a) Stem block (b) Inception-residual block (c) Upscaling block

Fig. 2. Detailed network blocks

For both the stem, inception-residual, and the upscaling block, we use two different kernel size of 3 x 3 and 5 x 5, each convolutions followed by batch normalization [23] and ReLU function. The kernel number for the first and second inception-residual block are 256 and 128, and the kernel number for the first and second upscaling block are 128 and 64.

Discriminator Network. We follow the CNN architecture by Ledig *et al*. [6] for the discriminator network. As shown in Fig. 3, the discriminator network contains seven convolution layers with 3 x 3 kernel, the kernel number is increasing from 64 to 512 as in the VGG network [24], followed by two 1 x 1 convolution and a global average pooling. We also use ReLU activation function for each convolution followed by sigmoid function at the end to obtain the probability for classification.

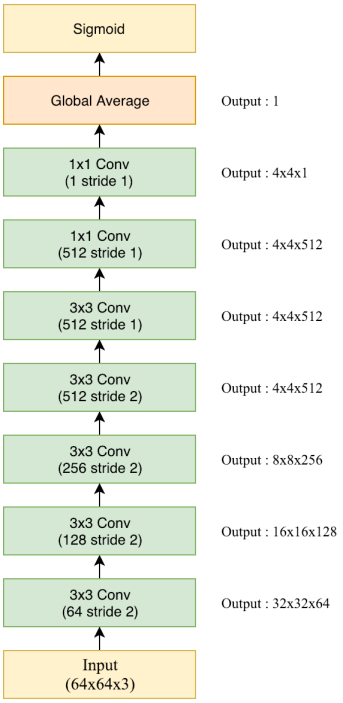


Fig. 3. Discriminator network architecture.

B. Dataset

The dataset that we use in this research is the cropped and aligned version of CelebA dataset *[25]*. The dataset consist of 202500 celebrity face images that has vorious facial type and pose variations. From the dataset we randomly sample 500 image for testing dataset to evaluate image reconstruction performance. The images that used for training and testing are resized into 64 x 64, then degraded by downscaling the images by factor 4 and adding some random noise to the images.

C. Loss Functions

To train the generator network and discriminator network simultaneously, we use the adversarial loss formulation from the original GAN framework [5], where the discriminator D trained to maximize the probability to differentiate between real and generated images from generator G, and train the G to minimize $log(1 - D(G(z)))$ where $D(x)$ represents the probability that x is real image rather than generated image, z represents low-resolution degraded images and $G(z)$ is the reconstructed images. Mathematically, the adversarial loss function is :

$$\min_G \max_D \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1 - D(G(z))\right)\right] \quad (1)$$

The formulation allows the generator to generate images that close to the ground-truth thus difficult to distinguish by the discriminator. Then we apply additional pixel-wise loss, feature-wise loss and PSNR loss for the generator to improve generated images quality.

1) Pixel-wise Loss

To enforce the output images to be close to the ground truth, let $\{z_i, i = 1..N\}$ denote the low-resolution degraded images, and $\{x_i, i = 1..N\}$ denote the ground truth, we measure the distance between the pixel values of the output images and the ground-truth, calculated as:

$$\frac{1}{N}\sum_{i=1}^{N}\sqrt{(G(z_i) - x_i)^2} \quad (2)$$

2) Feature-wise Loss.

To accomplish more realistic images, we measure the distance between features of generated and real images. The features are obtained from the output of the 5th convolution layer activation function, represents as abstractions of the structural information of the images. Different from *[7]*, we measure the feature differences using L2 distance as used in (2), defined as:

$$\frac{1}{N}\sum_{i=1}^{N}\sqrt{\left(\phi_\theta(G(z_i)) - \phi_\theta(x_i)\right)^2} \quad (3)$$

Where $\phi_\theta(x)$ represents the extracted features of input x. This term will enforce the generated images to have similar features as the ground truth.

3) PSNR Loss.

To reduce noises and generate realistic textures on generated images, we apply PSNR loss, defined as:

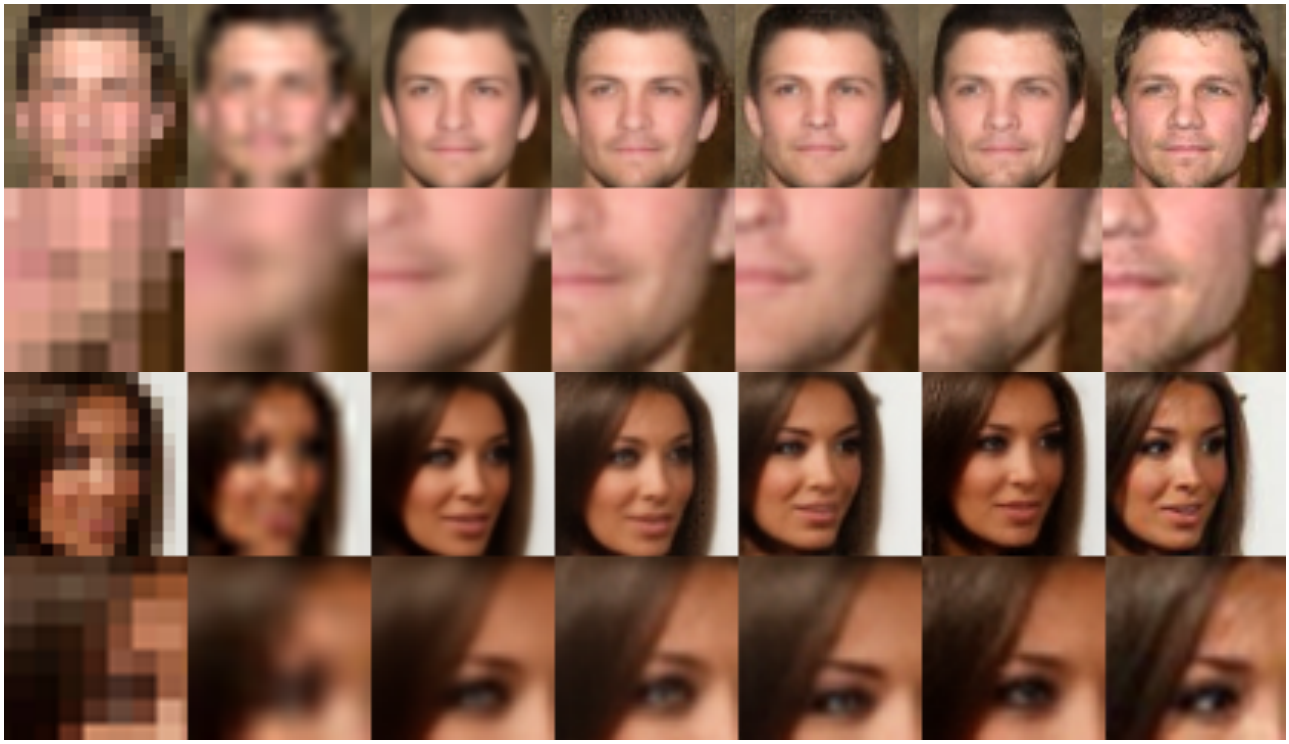
$$MSE = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{m\,n}\sum_{j=0}^{m-1}\sum_{k=0}^{n-1}\left[G(z_i)_{(j,k)} - x_{i(j,k)}\right]^2 \quad (4)$$

$$PSNR = 20 * \log_{10}(MAX_I) - 10 * log_{10}(MSE) \quad (5)$$

Where m and n are width and height of the in and MAX_I is the maximum pixel value of the images. We train the generator G to maximize the PSNR.

IV. Experiments And Discussions

We analyze the performance of our proposed method by performing two different experiments. First we perform experiments on loss function to discover the impact of using different loss formulations. Second we perform experiments on face verification to test the perceptual quality of the generated images from different scenarios.



(a) input (b) bicubic (c) scenario 1 (d) scenario 2 (e) scenario 3 (f) scenario 4 (g) GT

Fig. 4. Generated image results. The input images are upscaled for visualization.

A. Experiments on Loss Function

To discover the effectiveness of training using GAN framework and the impact of applying the additional loss functions, we tested our system in four different scenarios:

1. Train the generator network independently without adversarial process, using only the pixel-wise loss.
2. Train both network simultaneously using adversarial loss and apply pixel-wise loss to the generator.
3. Train both network simultaneously using adversarial loss and apply pixel-wise loss and feature-wise loss to the generator.
4. Train both network simultaneously using adversarial loss and apply pixel-wise loss, feature-wise loss and PSNR loss to the generator.

We train the models using Adam optimizer [26] with momentum terms $\beta_1 = 0.5$, $\beta_2 = 0.999$, initial learning rate $lr = 0.0002$ and halved every 5000 batches, and the batch size is 32. The filters weight are initialized using Glorot initializer [27]. The networks are trained on NVIDIA Tesla P100 16GB GPU for 2 hours, then we measure the performance using PSNR and Structural Similarity (SSIM). The quantitative result is shown in Table I, and the generated image results shown in Fig. 4.

TABLE I. Quantitative results comparison.

Scenario	1	2	3	4
PSNR	**26.615**	26.332	26.018	26.109
SSIM	**0.8461**	0.8357	0.8224	0.8313

In general, all four scenarios have succeeded to super-resolve and reconstruct the degraded face images. As shown in Fig. 4, the generated images from our model has much more facial information and details than simply upscaling the image using bicubic interpolation which has less details and lack of high frequency components. From the quantitative results shown in TABLE I, the highest PSNR and SSIM score are obtained from the first scenario where we only train the generator using the pixel-wise loss. Because the PSNR and SSIM calculate the error based on the pixel value and the pixel-wise loss enforced the pixel value of the generated images to be as close as possible to the ground truth.

However, upon closer inspection to the details, it can be seen in Fig. 4(c) that the result from first scenario has smoother texture. The second scenario generated more high frequency details that can fool the discriminator during the adversarial process, but sometimes the image still contains unrealistic artefacts. As depicted in Fig. 5(a), the nose area of generated image from scenario 2 has unrealistic shape, and it was fixed on scenario 3 where we add the feature-wise loss. The feature-wise loss enforced the generated images to have similar features with the ground truth, for example the shape of a nose or the details of an eye.

(a) Scenario 2 (b) Scenario 3 (c) GT

Fig. 5. Example of unrealistic artefacts on the generated image.

In the last scenario we applied the PSNR loss, which improved the texture quality. The PSNR loss term helped the generator to distinguish between noises and real textures, resulting overall better generated images quality. Although the PSNR and SSIM score didn't significantly increased, as shown in Fig. 6(b) that the generated image from scenario 4 has more realistic facial textures such as the texture of hair, beard and face wrinkles while still remain less noisy, resulting the most visually pleasant images.

(a) Scenario 3 (b) Scenario 4 (c) GT

Fig. 6. Texture differences between generated images.

Besides the generated images quality, we also found that the addition of loss terms increased the computational costs. It can be seen in Fig. 7 that the amount of training batches in 2 hours decreased when we applied additional loss term.

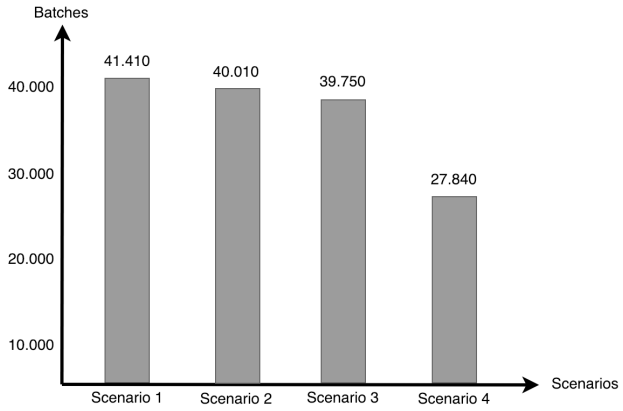
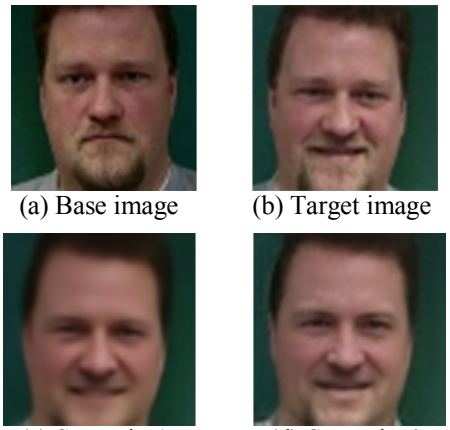


Fig. 7. The amount of training batches in 2 hours.

In the first scenario when we used only pixel-wise loss, the number of training batches in 2 hours reached 41.410, whereas the last scenario when we applied all the loss terms to the training process, 2 hours of training only reached 27.840 batches. However, with the same training time, overall generated images quality from scenario 4 is still better than the first three scenarios.

B. Experiments on Face Verification

The aim of SISR is not only to increase the resolution of an image, but also to recover high frequency components so that the reconstructed image has more realistic details. The reconstructed images should appeared to be real and provided more facial information that could be useful for many cases such as surveillance camera case. As shown in TABLE I that the highest PSNR and SSIM score was achieved from scenario 1. To prove that the results from scenario 4 are perceptually better than the results from scenario 1, we compared the similarity between ground-truth images and output images from both scenario 1 and scenario 4. We randomly sampled 15 images of 5 different person from IMM face dataset, each person consist of 3 images. We chose the first image as a base image, and 2 remaining images was degraded and then reconstructed. An example of base image and the reconstructed images is shown in Fig. 8. Then we encoded both base and reconstructed images. We used face encoding from One Shot Learning using Siamese Network based on FaceNet model [28] to extract facial features. Finally we measured L2 distance between the encoded images.



(a) Base image (b) Target image

(c) Scenario 1 (d) Scenario 4

Fig. 8. Example of face images for verification.

TABLE II. Distances comparison between encoded images from scenario 1 and scenario 4.

	Scenario 1		**Scenario 4**	
	Image 1	**Image 2**	**Image 1**	**Image 2**
Person 1	0.2236	0.1818	0.1216	0.1670
Person 2	0.1692	0.2097	0.1142	0.1768
Person 3	0.1724	0.2254	0.1384	0.1398
Person 4	0.1319	0.1505	0.1252	0.1476
Person 5	0.1979	0.2705	0.1897	0.1953

The distance between encoded images represents similarity of facial features that used to identify someone in face recognition system. As shown in TABLE II that the distances of face encoding from scenario 4 was 22% lower than the distances of face encoding from scenario 1. The results show that the generated images from scenario 4 give more realistic facial features towards the base images, and perceptually better than the generated images from scenario 1.

V. CONCLUSION

In this paper, we propose an inception-residual CNN for face images super-resolution that trained using GAN framework. Our inception-residual network was able to reconstruct better images by extracting visual information at different scales simultaneously. We also implemented the pixel-wise loss, feature-wise loss and PSNR loss. Extensive experimental results show that our proposed method proven able to generate visually pleasant face images that has realistic textures, rich of details and less noises with the highest PSNR score of 26.615 and SSIM score of 0.8461.

In this research we only used 3 x 3 and 5 x 5 convolutions. Some research explained that convolutions with larger spatial filters (e.g. 5 x 5 convolution) tend to increase computational cost and can be solved by using asymmetric convolutions.

Therefore, future research is needed to explore better network architecture and further investigations into the network properties and parameters.

VI. REFERENCES

[1] J. Yang, J. Wright, T. S. Huang and Y. Ma, "Image Super-Resolution Via Sparse Representation," in *IEEE Transactions on Image Processing*, 2010.

[2] R. Timofte, V. D. Smet and L. V. Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *International Conference on Computer Vision (ICCV)*, 2013.

[3] C. Dong, C. C. Loy and X. Tang, "Accelerating the Super-Resolution Convolutional Neural Network," in *European Conference on Computer Vision (ECCV)*, 2016.

[4] J. Kim, J. K. Lee and K. M. Lee, "Accurate Image Super-Resolution Using Very Deep Convolutional Networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] 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 (NIPS)*, 2014.

[6] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[7] X. Xu, D. Sun, J. Pan, Y. Zhang, H. Pfister and M.-H. Yang, "Learning to Super-Resolve Blurry Face and Text Images," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[8] W. Freeman and E. Pasztor, "Learning low-level vision," in *International Journal of Computer Vision (IJCV)*, 2000.

[9] W. Freeman, T. Jones and E. Pasztor, "Example-based super-resolution," in *IEEE Computer Graphics and Applications*, 2002.

[10] C. Dong, C. C. Loy, K. He and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European Conference on Computer Vision (ECCV)*, 2014.

[11] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[12] Q. Li, W. Cai, X. Wang, Y. Zhou, D. D. Feng and M. Chen, "Medical image classification with convolutional neural network," in *International Conference on Control Automation Robotics & Vision (ICARCV)*, 2014.

[13] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar and L. Fei-Fei, "Large-scale Video Classification with Convolutional Neural Networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[14] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[15] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[16] S. Sharma, K. Shanmugasundaram and S. K. Ramasamy, "FAREC — CNN based efficient face recognition technique using Dlib," in *International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, 2016.

[17] A. Toshev and C. Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[19] J. Kim, J. K. Lee and K. M. Lee, "Deeply-Recursive Convolutional Network for Image Super-Resolution," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[20] C. Szegedy, S. Loffe and V. Vanhoucke, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2017.

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[22] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[23] S. Loffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *International Conference on Machine Learning (ICML)*, 2015.

[24] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *ArXiv*, 2014.

[25] Z. Liu, P. Luo, X. Wang and X. Tang, "Deep Learning Face Attributes in the Wild," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[26] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *arXiv preprint arXiv:1412.6980*, 2014.

[27] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

[28] F. Scroff, D. Kalenichenko and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.