Journal of the Institute of Engineering
Volume 16, No. 1 Published: April 2021
ISSN: 1810-3383

Printed in Nepal

Devnagari Handwritten Characters Image Super-Resolution based on Enhanced SRGAN

Prasiddha Siwakoti [a], Sharad Kumar Ghimire [b]

Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering, Tribhuvan University, Nepal
Corresponding Authors: [a] prasiddha.siwakoti43@gmail.com, [b] skghimire@ioe.edu.np

Received: 2020-08-31 Revised: 2021-01-25 Accepted: 2021-02-19

Abstract:
The difficulty in machine learning-based image super-resolution is to generate high-frequency component in an image without introducing any artifacts. In this paper, Devnagari handwritten characters image using a generative adversarial network with a classifier is generated in high-resolution which is also classifiable. The generator architecture is modified by removing all batch normalization layers in generator architecture with a residual in residual dense block. Batch normalization is removed because it produces unwanted artifacts in the generated images. A Devnagari handwritten characters classifier is built using CNN. The classifier is used in the network to calculate the content loss. The adversarial loss is obtained from the GAN architecture and both of the losses are added to obtain total loss. Generated HR images is validated using six different evaluation metrics among which MSE, PSNR determines pixel-wise difference and SSIM compares images perceptually. Similarly, FID is used to measure the statistical similarity between the batch of generated images and its original batch. Finally, the Gradient similarity is used to assess the quality of the generated image. From the experimental results, we obtain MSE, PSNR and SSIM as 0.0507, 12.95(dB) and 0.8172 respectively. Similarly, the FID value obtained was 27.5 with the classification accuracy of image data of 98%. The gradient similarity between the generated image and the ground truth obtained was 0.9124.

Keywords: Super Resolution, RRDB, Generator, Discriminator, SSIM, FID

1. Introduction

Low-resolution images mean there are fewer details in an image that are detectable by the human visual system and same applies to the computer vision. Low resolution images when not accounted for could significantly decrease the classification accuracy of deep neural network. Image super-resolution means translating a low resolution image into a visually pleasing high-resolution image. There are various methods to enhance a low-resolution image into a high-resolution one. Some common image enhancement methods include image color adjustment, noise reduction and image interpolation but obtaining the high-frequency components from low-resolution image is still a difficult task. Image super-resolution technique using GAN can generate high resolution images with high-frequency details but those details could be fabricated details. In such cases, classification and recognition of images will be difficult. Robust algorithms are developed including ANN, CNN and deeper neural networks in recognizing and reconstructing images using various kinds of large datasets. In case of Nepali characters, DHCD dataset is made available, published by P. K. Gyawali et al. [15]. DHCD contains 46 classes of images of handwritten characters. Each class contains 2000 images, 1700 images for training and 300 images for testing. SRCNN [1] is one of the pioneers that demonstrated the potential for convolutional network to be applied to this research domain achieving the state of the art reconstruction quality while maintaining the light weight structure. VDSR [3] on the other hand is one of the fastest models while also capable of providing highly accurate image construction. Dong et al. [5] down-scaled an image by using bi-cubic interpolation and fed that image and trained on a 3 layered convolution network. To limit the numbers of

parameters (DRCN) Deeply-Recursive Convolutional Network [6] is considered best. A function called perceptual loss function was conceptualized by Jonshon et al. [2] and Bruna et al. [7] to regenerate more realistic high resolution image. These researches helped us get the concept of how a CNN model can be deeper yet fast. GANs (generative adversarial networks) proposed by Goodfellow [8] have been considerably successful as a framework for generative models in recent years. In addition to that, a variant of GAN called conditional GAN proposed by Mirza et al. [10] uses labels as another input to the generator which helps to control the generator output avoiding the necessity of changing generators architecture. SRGAN by Ledig et al. [4] is one of the most influential image super resolution concept which produces visually superior images in high resolution from given low resolution counterpart, but with low PSNR value. It achieves such quality by using Resnets [11]. ESRGAN [12] is the latest improvisation in the field of image super resolution by making residual learning denser by introducing RRDB block.

A method to generate high resolution handwritten Nepali characters images from a low resolution image employing the concept of SRGAN [4] but removing the batch normalization layer in the generator architecture and adding a residual in residual block (RRDB) instead of normal residual block as stated in ESRGAN [12]. The purpose of using RRDB instead of normal residual block is to enhance feature mapping as well as removing vanishing gradient problem. In contrast to ESRGAN we introduce a Devnagari handwritten character classifier as a third player in the GAN's architecture.

2. Method

A discriminator network is described by $D_{\theta D}$ and updated alternately with generator network $G_{\theta G}$ to solve the following adversarial function as stated in the paper by Goodfellow et al.[8] and SRGAN [4]:

$$\min G\theta_{\max} D\theta E_{I^{HR}} \sim pGT(I^{HR})[\log D_{\theta D}(I^{HR})] + E_{I^{LR}} \sim pG(I^{LR})[\log(1 - D_{\theta D}(G_{\theta G}(I^{LR})))] \quad (1)$$

In the training process, the objective function makes generator G try to fool the discriminator D by generating images similar to that of the real image. D is trained initially on the real dataset but after that it is also trained on the images given by the G.G tries to outperform D by producing similar looking image to the real image, at the same time discriminator tries to determine whether the image is from real data set or not. As the training proceeds both G and D gets better in their job. Finally, at the end of the training, a perceptually superior looking image compared to the image generated by traditional methods which just use pixel-wise calculation is generated. The Generator is composed of 8 blocks including RRDB (Residual in Residual Block) block. The first convolution (pre-residual) layer is followed by 3 RRDB blocks (residual and residual dense block) and RRDB is followed by a post residual convolution layer. Two sub-pixel convolution layers is used to upsapmle the image and a final convolution layer at the output layer. Every convolution block is Leakey Relu activated except the output layer .The output layer has TanH activation. The Generator G is implemented with kernel size of 3*3 and number of filters in each convolution layer is 8.

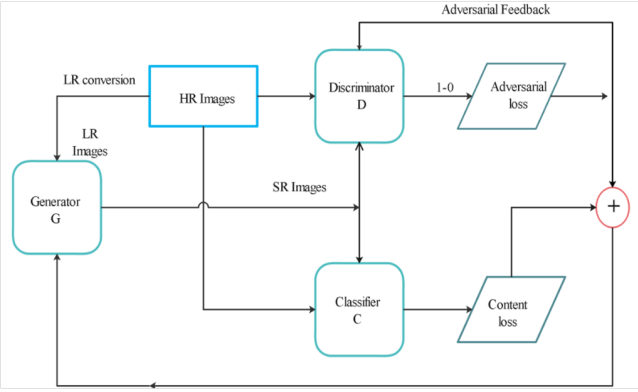


Figure 1: An illustration of overall block diagram: G creates a reconstructed image ISR, C classifies the real image IHR and ISR, minimizing cross-entropy loss and the D discriminates that weather the imager generated corresponds to true images

In discretization of the image from real world into a digital imaging device each pixel approximate color based on light intensity and each cmos sensor has pixel pitch of about 5 microns. So there is a gap between two pixels. These gaps are filled using different algorithm digitally and they are called sub pixel. The concept of sub pixel convolution is introduced to maximize the image information and improve the resolution. It is done by obtaining low-resolution feature maps using convolution and multi-channel reconstruction to

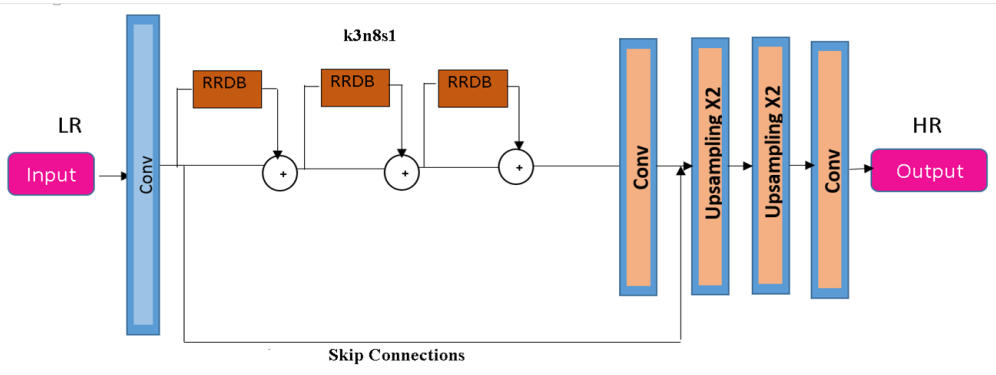


Figure 2: Generator architecture (G in Figure 1)

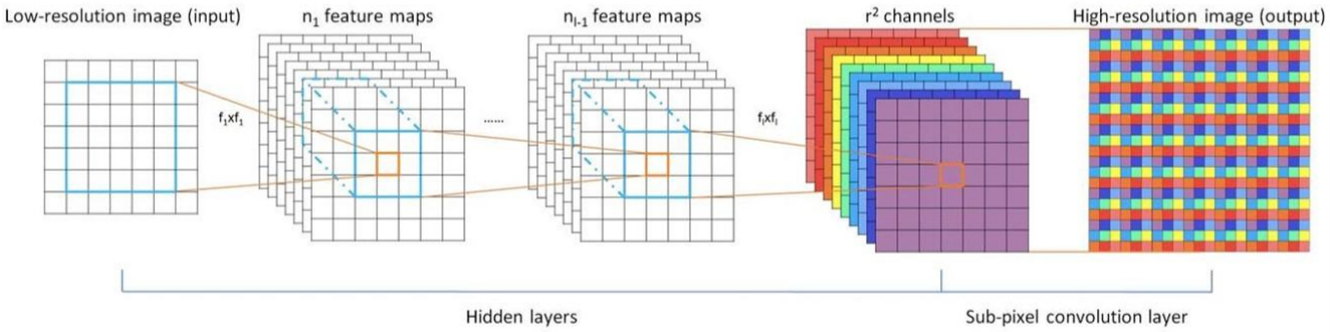


Figure 3: Sub-pixel convolution [16]

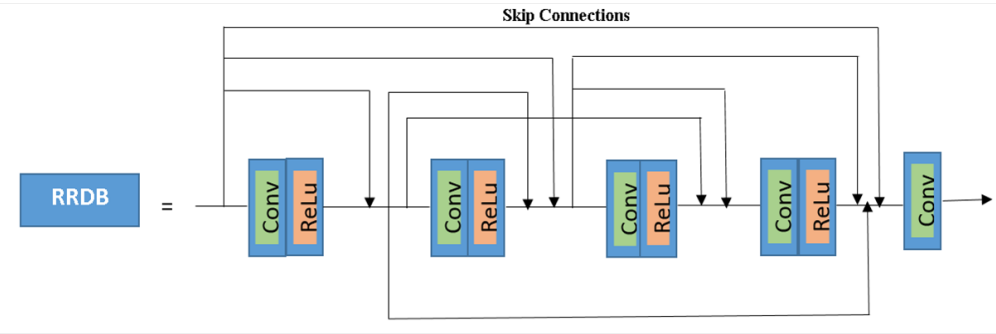


Figure 4: RRDB block employing dense skip connections

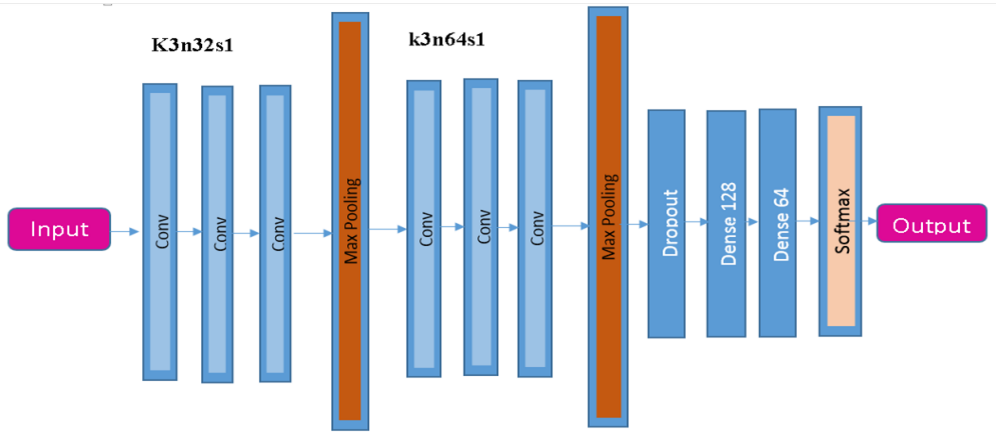


Figure 5: Classifier architecture (C in Figure 1)

construct a high resolution feature map. In ESPCN [16] r*r channel feature maps before the up sampling layers is extracted, where r is the up sampling factor. Now the r*r feature map in the previous layer is combined to form a new dimension of W*r, H*r (where W=width, H=height and C=channels). It is realized by following function:

$$N*(C*r*r)*W*H \longrightarrow N*C*(H*r)(W*r)$$

Inside RRDB, each block contains 5 convolution layers employing a dense skip connection. RRDB is used to enhance the performance of generator by using dense connection and multilevel architecture.

The Discriminator is composed of 8 convolution blocks, each convolution block is followed by a batch normalization layer and activated using a Leaky ReLU function. The output is obtained from a layer with a sigmoid function after going through 2 dense layers. The sigmoid function is used to obtain the probability value between 0 and 1. All the convolution layers have kernel size of 3*3 but filter size doubles after each 2 layers starting from 64 to 512. The classifier used is a general CNN handwritten character classifier trained on DHCD dataset. The architecture of classifier is such that, it contains 6 layers of convolution layer. There is a max-pooling layer after each 3 convolution layer and a dropout layer is used to randomly deactivate some neurons. Two dense layers are followed by an output layer with softmax activation. It is trained and used as a pre-trained classifier to get content loss later.

For the generator to perform well, the loss function plays a very important role. The loss function consists of two losses. The content loss and the adversarial loss, Content loss is essentially the MSE loss and the adversarial loss is modified GAN loss. The total loss is the addition of content loss and adversarial loss.

$$l^{SR} = l^{SR}\,mse + 10^{-3} \cdot l^{SR}\,adv \tag{2}$$

The content loss is calculated as pixel-wise MSE loss:

$$l^{SR}\,mse = 1/rWH(\Sigma_{=1}\Sigma_{=1}(I^{HR}{}_{x,y} - G_{\theta G}(I^{LR})_{x,y})^2) \tag{3}$$

Here, H, W and C represent height, width and channels of the image that we are dealing respectively. Where r is scale factor. Similarly, I^{HR} and I^{LR} are high resolution and low-resolution images respectively. The value of C can be 1 or 3. From the GAN architecture, we add adversarial loss to the overall loss function. It is weighted by some small value. By applying adversarial loss in updating generator and the discriminator we are able to generate perceptually realistic and natural images. The formula for adversarial loss is as follows:

$$l^{SR}\,adv = \Sigma_{=1} - \log D_{\theta D}(G_{\theta G}(I^{SR}n)) \tag{4}$$

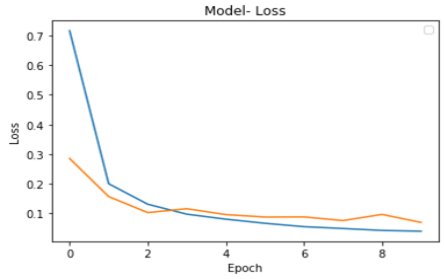
Here, $D_{\theta D}(G_{\theta G}(I^{SR}n))$ is the probability that the reconstructed image $G_{\theta G}(I^{SR}n)$ is distinguished as real image by discriminator. This is not an original GAN loss function. To get the proper result, we need to minimize above equation instead of $\log[1 - D_{\theta D}(G_{\theta G}(I^{SR}n)]$.

One of the challenging tasks during the implementation of the proposed model is training. Dataset is prepared by separating the whole dataset into train and test set (1700 for training and 300 for testing).First the classifier is trained with the given image to classify DHC by applying conventional training methodology. After achieving certain accuracy on classifier, data is prepared for adversarial training. Low resolution image is fed to generator to create high resolution (32*32). Discriminator is trained with real images, and then trained on both real and fake high resolution image. The discriminator loss is then calculated. Also the pre-trained classifier network is fed with high resolution image to generated content loss in the image. Train the generator by applying so obtained losses through back propagation. Repeat step 4 until the no more optimization is possible.

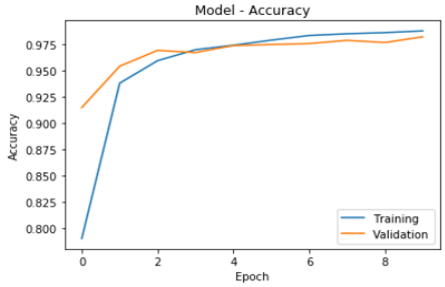
3. Results and Discussion

The classifier is set for the training process with batch of 32 images and 10 epochs. After the training of classifier, training loss of 0.03998, training accuracy 0.9877 and validation loss of 0.696 and validation accuracy of 0.9820 is obtained.

Adversarial training was done on google colaboratory cpu for 10000 epochs (12 hours limit). Since the model is deep, fast convergence and lower generator loss is achieved using Adam optimizer [17] with learning rate of 0.0002. The learning rate is determined using hit and trial method starting from 0.0001 to 0.0002.

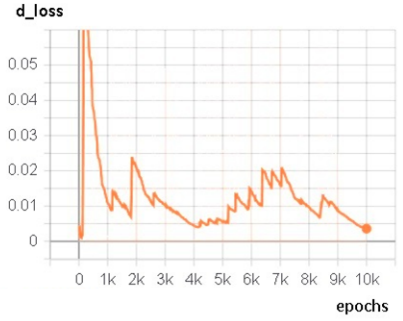


(a)

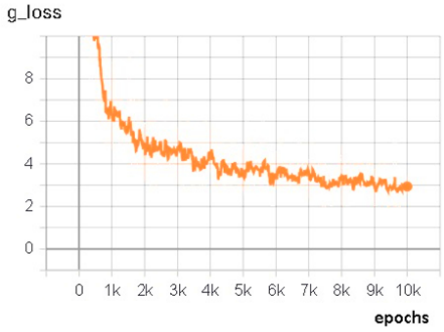


(b)

Figure 6: (a) Training and Validation loss and (b)Training and Validation accuracy of the classifier



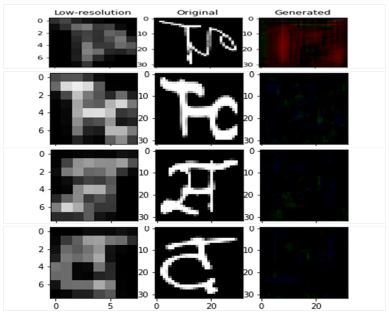
(a)



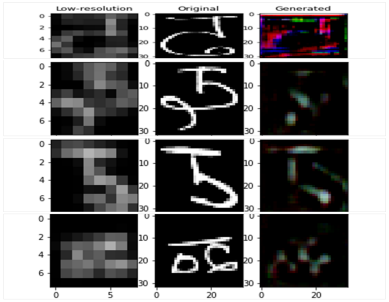
(b)

Figure 7: (a) Graph of discriminator loss and (b) generator loss for 10000 epochs

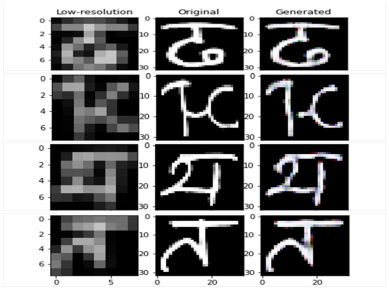
With the completion of training, we found the discriminator loss as 0.0097 and the generator loss as 2.877. The GANs losses are not so intuitive like other models. Since generator and the discriminator are competing against each other, increment in loss of one network means the other network is improving and minimizing loss and hence the unstable looking graph.



(a)



(b)



(c)

Figure 8: Generated images along with its ground truth and low-resolution counterpart
(a) results after 500 epochs
(b) after 3000 epochs
(c) after 9500 epochs

Table 1: Obtained metrics compared with benchmark MNIST dataset

Metrics	Datasets	
	DHCD	MNIST
MSE	0.0507	0.08425
PSNR	12.95(dB)	10.741(dB)
SSIM	0.8172	0.85063

Table 2: FID for different datasets to measure the performance of DCGAN [13]

DCGAN Image dataset	method	b, a	updates	FID
CelebA	TTUR	1e-5, 5e-4	225k	12.5
CIFAR-10	TTUR	1e-5, 5e-4	75k	36.9
SVHN	TTUR	1e-5, 5e-4	165k	12.5
LSUN	TTUR	1e-5, 5e-4	340k	57.5

Table 3: Gradient Similarity of different dataset [14]

Database	q	g
A57	0.9002	0.9004
IVC	0.9294	0.9297
CSIQ	0.9126	0.9126
LIVE	0.9554	0.9555
TID	0.8554	0.8532
Toyoma	0.9233	0.9241

The pre-trained model was used to extract features from generated high resolution images and obtain pixel-wise MSE. The adversarial loss and content loss was fed back to the generator and the training iterated to 10000 iterations. Above are the results in which MSE=0.0507, PSNR=12.95(dB), SSIM=0.8172, was obtained. From the above results (SSIM=1 means perfectly match) we can say that the generated high resolution image is nearly equivalent to original high resolution image. The values resulted images were in the range [-1,1] the maximum pixel value is 1 which is the reason the psnr value seems low. That means the system was able to generate images equivalent to the original resolution from given low resolution input. Similarly FID (Frechet Inception Distance) was used to find the statistical similarity between two sets of data the real and the generated data set in different stages of training. The final batch of generated image had obtained 27.2 as FID value. Classification accuracy of generated images determined using the pre-trained classifier was 98%. Finally one more metrics was used to quantify the image quality, gradient similarity of 0.9124 was obtained from the generated images. For and additional experiment two different alignment change was performed in test images, for shear =0.2 and rotation =15°, results were comparable to fundamental results. But for larger change in orientation (shear=0.5 and rotation=30°) the results were degraded.

4. Conclusion

This research aims to put a stepping stone on generating handwritten characters (Nepali) in high resolution classifiable form. A modified GAN is created to generated very original looking handwritten characters from its low resolution images. Visual assessment is not enough for the validation of generated results. So the result is validated using 6 different metrics, MSE, PSNR, SSIM, FID, Classification Accuracy and Gradient Similarity. MSE and PSNR are the contemporary methods of image similarity measurement. It can only make pixel-wise comparisons. The average MSE and PSNR obtained is 0.0507 and 12.95(dB). So SSIM is considered to validate the structural similarity of the generated image with the real image and the value obtained is 0.8172. Similarly, batch of generated images was used to compare their statistic similarity, which showed that not only the single selected image but whole batch is statistically similar to the original dataset and the FID value obtained is 27.2. Classification Accuracy of 98% was observed which is very close to classification accuracy of the original dataset. This means that the generated image is classifiable by computer. Finally gradient similarity was calculated between the real and generated image to measure the quality of the generated image for which the similarity value of 0.9124 was obtained.

References

[1] C. Dong, K. He, C. C. Loy, and X. Tang X, "Image Super-Resolution Using Deep Convolutional Networks," *arXiv:1501.00092v3*, 2015.

[2] C. Dong, C. Loy, X. and Tang X, "Accelerating The Super-Resolution Convolutional Neural Network," in *European Conference on Computer Vision*, 2016, pp. 391–407.

[3] J. Kim, J. K. Lee, and K. M. Lee, "Accurate Image Super-Resolution Using Very Deep Convolutional Networks," *arXiv:1511.04587v2*, 2016.

[4] A. Acosta, A. Aitken, J. Caballero, A. Cunningham, F. Husz, C. Ledig, L. Theis, A. Tejani, J. Totz, and Z. Wang Z, "Photo-Realistic Single ImageSuper-Resolution Using a Generative Adversarial Network," *arXiv:1609.04802v5*, 2016.

[5] C. Dong, K. He, C. C. Loy, and X. Tang, "Learning a Deep Convolutional Network for Image Super-Resolution" in *European Conference on Computer Vision*, 2014, pp. 184—199.

[6] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-Recursive Convolutional Network for Image Super-Resolution," *arXiv:1511.04491v2*, 2016.

[7] J. Bruna, Y. LeCun, and P. Sprechmann, "Super-Resolution with Deep Convolutional Sufficient Statistics," *arXiv:1511.05666*, 2016.

[8] Y. Bengio, A. Courville, I. J. Goodfellow, M. Mirza, S. Ozair, J. Pouget-Abadie, Warde-Farley, and B. Xu, "Generative Adversarial Networks," *NIPS*, 2014.

[9] S. Chintala, L. Metz, and A. Radford, "Un-Supervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *arXiv:1511.06434*, 2016.

[10] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," *arXiv:1411.1784*, 2015.

[11] K. He, S. Ren, J. Sun, and X. Zhang, "Deep Residual Learning for Image Recognition," *arXiv:1512.03385v1*, 2016.

[12] D. Chao, C. L. Chen, G. Jinjin, Y. Ke, W. Shixiang, W. Xintao, Q. Yu, and L. Yihao L, "Enhanced Super-Resolution Generative Adversarial Networks," *ECCV*, 2018.

[13] N. Bernhard, R. Hubert, H. Martin, H. Sepp, and U. Thomas, "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium," *arXiv:1706.08500*, 2017.

[14] L. Anmin, N. Manish, and L. Weisi L, "Image Quality Assessment Based on Gradient Similarity," *IEEE Transaction on image processing*, vol. 21, no. 4, 2012.

[15] S. Achrya, P. K. Gyawali, and A. K. Pant, "Deep Learning Based Large Scale Handwritten Devanagari Character Recognition," *IEEE SKIMA*, 2015.

[16] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, A. Wang Z, "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural," *arXiv:1609.05158*, 2016.

[17] D. P. Kingma and J. Ba, "Adam: A Method For Stochastic Optimization," *arXiv:1412.6980*, 2017.