Image Super-Resolution using General Adversarial Network

CSCI-B657: Computer Vision
Spring 2019
Final Project Report

Darshan Shinde[1], Pei-Yi Cheng[1], Virendra Wali[1]

***Abstract*— Super resolution is one of the hot research topics in computer vision. We re-implemented this technique which was first proposed by Bulat et al. [1]. This technique comprises two GAN (General Adversarial Network) models named High-to-Low and Low-to-High. A High-to-Low model which generates paired dataset whereas a Low-to-High model is used to resolve the image. The results show that the GAN model successfully super resolves face images with some exceptions. We aim to validate the author's claim via testing the model with another dataset. In addition, we also aim to explore the feasibility of using only Low-to-High GAN model with a loss function proposed by author on artificially generated paired data set via bi-linear down sampling.**

I. INTRODUCTION

Nowadays, many crime incidents get captured using Closed Circuit Television (CCTV) Cameras but with a poor footage quality which creates difficulties during investigation. This problem can be solved using image quality improvement algorithms [4], [2] (and many more). Out of these we are trying to re-implement the idea of increasing the resolution of low-resolution images using GAN Model proposed by [1]. In the past, due to lack of readily available low-resolution images, techniques like bilinear / cubic down sampling, Gaussian blur etc. were employed to artificially generate low resolution images. However, these artificially generated low resolution images bore no resemblance to the ones encountered in the real-world scenarios. In order to overcome this shortcoming, Adrian Bulat proposed a two-model technique which comprised of 2 GAN models named High-to-Low & Low-to-High, in which, the a High-to-Low GAN model could be utilized to generate the low-resolution images.

The author proposed infusing a high-resolution image with adjusted noise component so that the same can be on par with the low-resolution images encountered in the real-world scenarios. The author concludes that this process of generating low resolution images and using the same for training GAN model to output high resolution images outperforms all other previous techniques. To corroborate his assertion, he has furnished numeric results of some experiments, comparing the same with ones obtained from different approaches mentioned above.

[1]Darshan Shinde and Pei-Yi Cheng are data science graduate students and Virendra Wali is Computer Science graduate student at the School of Informatics, Computing, and Engineering, Indiana University, Bloomington. All of them have equally contributed to this project.

In this project, we aim to replicate experiments conducted by author on different data sets and confirm the author's conclusions. In addition, we also aim to improve the performance of the approach proposed by author by combining it with some other techniques.

II. BACKGROUND & RELATED WORK

A vast majority of work has been done towards solving the problem of low resolution to high resolution. Prior to this, majority of super-resolution methods used low resolution images created by bi-linear down-sampling. Bi-linear down-sampling was followed by blurring or GAN with L2-pixel loss. The results of those methods are noisier and blurrier. To overcome this flaw, GAN + L2 loss is used but L2 loss alone failed to de-noise the input and generate a good output. So, to solve this problem, a High-to-Low network GAN is used to down-sample high-resolution images first followed by Low-to-High GAN which is used for super-resolving images into high-resolution ones.

In addition to this, Dong et al. [2] proposed a Convolutional Neural Network model to jointly optimize three operations for super-resolution: extract feature maps of low-resolution image, non-linearly map low-resolution feature maps to high-resolution patch representation and reconstruct high- resolution image by combining the predictions within a spatial neighborhood. Mean Squared Error is used as the loss function. The results show that increasing the number of filters and using reasonably larger filter size would improve performance at the cost of running time. However, deeper structure is not always a good choice for super-resolution.

Another deep learning model which was employed to optimize images was Super Resolution Convolutional Neural Networks (SRCNN) [2]. It was one of the techniques that bettered the traditional models in place. This model comprised of 3 convolution layers: patch extraction & representation, non-liner mapping & reconstruction. A yet another deep learning model named Very Deep Super Resolution (VDSR) [5] employed a similar structure as SRCNN. However, it dived much deeper to obtain significant accuracy. Both the models employ bi-cubic sampling at the input stage and deal with feature maps to achieve parity with the output.

However, none of the models mentioned & discussed above tend to produce quality low resolution images in the real-world scenarios. Hence, a two-model technique as mentioned above was proposed by Adrian Bulat to generate

low resolution images on par with the ones encountered in real world instances.

III. METHODS

Our implementation uses General Adversarial Network to super-resolve the low- resolution image of size 16x16 into high-resolution of size 64x64. This network is trained on paired dataset of low- resolution images and high-resolution images. This approach differs from the past related work in sense of how it generated paired dataset.

The main architecture proposed in this paper has two main sub components (GANs). The first one (High-to-Low) is used for learning the down-sampling and another (Low-to-High) is to achieve a primary objective of improving quality of low- resolution image to high-resolution. The complete architecture of the model is shown in Fig. 1. A High-to-Low sub component is trained on two disjoint and unpaired datasets which can be used effectively to simulate the image degradation process. The first dataset contains the high-resolution facial images. The second dataset contains the blurry and low-quality low- resolution images. For our experiments, we are using Celeb-A dataset (High resolution) [6] and Widerface dataset (low resolution) [8] which is used to contaminate the high- resolution images with noise and artefact to create low resolution images. In addition, we also tested the model on Indian Movie Face Database (IMFDB) [7].

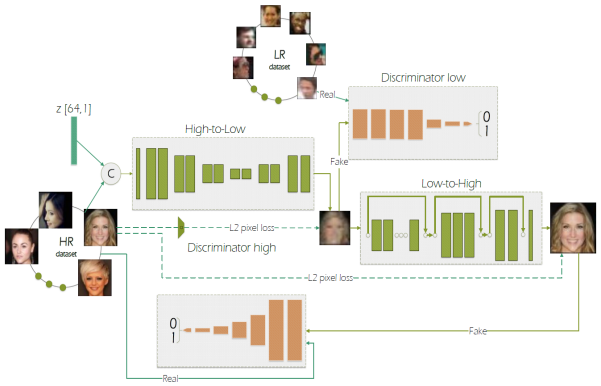

Fig. 1. Overall architecture and training pipeline (image is reprint from [1])

Below is the detailed description of our model:

A. High-To-Low

We are using this High-to-Low network to generate paired dataset of high-resolution images and their corresponding low-resolution images which we will use to train Low-to-High network. As a final model, both networks are combined to generate a single network. The first generator module of this single network takes a high-resolution image of size 64x64 along with a random noise of size 64x1 as input and generates a low-resolution image of size 16x16. This low-resolution image is further passed on to a first discriminator module (which is trained on low-resolution image dataset) which delivers a judgement whether the image is real or fake. Then, this generated low-resolution image is passed on to a second generator module (which is trained on paired dataset of high-resolution images and low-resolution images) and converts it into high- resolution image of size 64x64 which is further passed on to a second discriminator module which classifies the ground truth image and fake images.

1) High-To-Low Generator: For High-to-Low generator, the first layer is a fully connected layer that takes high-resolution image concatenated with noise as model input. An encoder-decoder structure contains 6 groups of residual blocks (2 blocks in each group) so that the image resolution is reduced from size 64x64 to size 4x4 using 4 times resolution degradation via pooling layers, and image resolution is increased from size 4x4 to size 16x16 using two times resolution improvement via pixel shuffle layers. Fig. 2 shows the architecture of generator model.

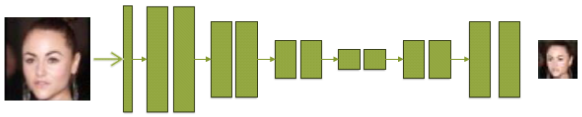

Fig. 2. High-To-Low generator (image is reprint from [1])

2) High-To-Low Discriminator: High-to-Low discriminator contains 6 residual blocks excluding batch normalization. This is followed by using a fully connected layer. The input size 16x16 resolution is reduced at the last two blocks by using max-pooling. Fig. 3 shows the architecture of discriminator model.

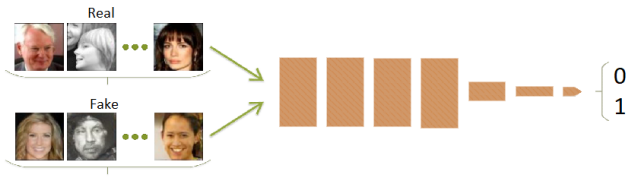

Fig. 3. High-To-Low discriminator (image is reprint from [1])

B. Low-To-High

The output of the High-to-Low network is fed to Low-to-High generator network which generates high resolution images. In addition, the high-resolution images which are used as an input for the High to Low architecture are used as a ground truth during this phase.

1) Low-To-High Generator: The Low-to-High generator network consists of 3 groups of residual blocks each with a varying number. First group has 12 residual blocks, second comprises of 3 whereas the remainder consists of 2. Image resolution is increased from size 16x16 to size 64x64 by employing bilinear interpolation 2 times between groups of residual blocks. Fig. 4 shows the architecture of Low-to-High generator.

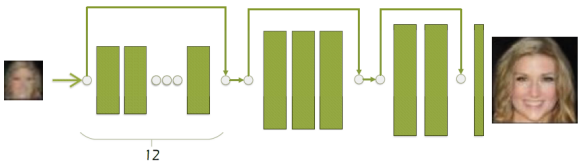

Fig. 4. Low-To-High generator (image is reprint from [1])

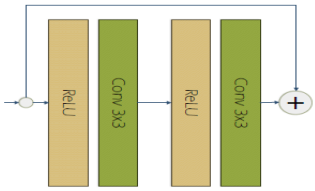

Fig. 7. Residual block used in discriminator (image is reprint from [1])

2) Low-To-High Discriminator: The Low-to-High discriminator is like High-to-Low discriminator with exception of decreasing resolution only twice using max-pooling layers. The structure of Low-To-High discriminator is shown in Fig. 5

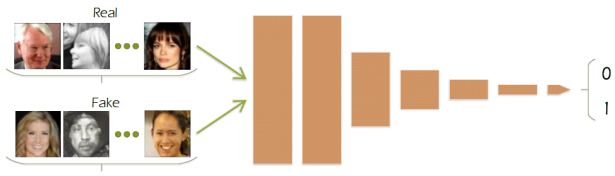

Fig. 5. Low-To-High discriminator (image is reprint from [1])

C. Residual blocks

Generator and discriminator are deep neural networks. There is a common problem in deep neural network of vanishing gradients. To resolve this issue of vanishing gradients, residual blocks are introduced in [3]. A residual block here consists of batch normalization, ReLU, Conv3x3. Each layer feeds into the next layer while utilizing skip connection to jump over some layers to counter the effect of gradient vanishing in very deep neural network. The structure of residual blocks used in generators and discriminator are shown in Fig. 6 and 7.

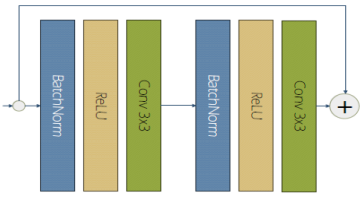

Fig. 6. Residual block used in generator (image is reprint from [1])

D. Loss Function

The combination of pixel loss and adversarial loss is used as loss function for each network. Pixel loss (eq. 3) is calculated by L2 distance between real image and generated image, motivating the generator to preserve the characteristics in the images. Adversarial loss (eq. 2) helps the image generation process. Total loss (eq. 1) is weighted sum of pixel loss and adversarial loss. These loss functions are defined as follows:

$$l = \alpha l_{pixel} + \beta l_{GAN} \tag{1}$$

$$l_{GAN} = \mathbb{E}_{x\sim\mathbb{P}_r}[min(0,-1+D(x))] + \mathbb{E}_{x\sim\mathbb{P}_g}[min(0,-1-D(\hat{x}))] \tag{2}$$

$$l_{pixel} = \frac{1}{WH}\Sigma_{i=1}^{W}\Sigma_{j=1}^{H}(F(I^{hr})_{i,j} - G_{\theta_G}(I^d)_{i,j})^2) \tag{3}$$

Where D(x) is discriminator output for real images, D($\hat{x}$) is discriminator output for fake images, W is width of generated image, H is the height of generated image, F is the function which maps resolution of real image and generated image.

E. Training

We have built 2 GAN Models i.e. High-to-Low & Low-to-High GAN using fully connected blocks and residual blocks. Here, we are using Adam optimizer with hyper parameter value $\beta_1 = 0$ and $\beta_2 = 0.9$ and a default alpha value. The learning rate for both the models turns out to be $1e-4$. We are using two loss functions i.e. pixel loss (pytorch's MSELoss) and custom GAN loss. For complete model, we are using the combined weighted loss which is eq. 1. Here, we are using $\alpha = 1$ and $\beta = 0.05$. In addition, we are also using 1:1 discriminator to generate update ratio instead of authors 5:1 update ratio. In order to train the model, we first trained the 2 GAN models separately each with 125 epochs and followed by 2500 epochs for the both the models combined.

F. Evaluation

"Frechet Inception Distance" (FID) [4] is used in this project to measure the similarity of generated images to real images. "Inception Score" is a performance measurement that has positive relationship with human judgment. "Frechet distance" is used to measure the difference of two Gaussians given mean and covariance. Combining these two concepts of "Inception Score" and "Frechet distance", FID is more consistent with the noise level and human judgment.

In order to calculate the distance between given two images, the same are passed as an input to the inception network one at a time. The output extracted from the third layer of the inception network is a feature map which is used to calculate the mean and the co-variance of the given image.

The distance between two images is calculated using following formula (eq. 4):

$$FID = \| \mu_r - \mu g \|^2 + Tr(\Sigma_r + \Sigma_g - 2\sqrt{\Sigma r \Sigma g}) \quad (4)$$

Where,
μ_r = Mean of Real Image,
μ_g = Mean of Generated Image,
Σ_r = covariance of Real Image,
Σ_g = covariance of Generated Image

IV. EXPERIMENTS & RESULTS

In this project we have re-implemented the image super resolution technique proposed by Adrian Bulat. Our main goal is to cross validate the claim made by author. In addition to this, we tried to test the universality of the model. For the same purpose, we trained our model on a dataset and attempted to test the same on a different dataset in order to analyze the performance of our model. Author Bulat proposes to use a new GAN model to generate paired dataset and a loss function to train the model. We conducted an experiment to verify the necessity of 2 techniques as we wanted to investigate whether a lone loss function would suffice to super resolve the image.

The goal of the first experiment is to cross validate the claim made by author. For this we tried to replicate the author's network structure. We trained our model under the same training conditions as specified by author and we also made use of the data set mentioned by him. We trained our High-To-Low generator on Celeb-A dataset and used Widerface dataset to train High-To-Low discriminator. A few images from celeb-A data set were set aside for the testing purpose. We used these images for generating paired dataset. We trained our Low-To-High GAN model using this paired dataset. Remaining few images from celeb-A dataset were used to test this whole model. The results of this experiment are shown in Fig. 8.

Fig. 8. Results of experiment 1 (1.Real Image 2.Down-sampled Image 3.Generated Image)

The FID metric proposed by the author has been used to measure the performance of the model. We measured the FID metric for our model and compared it with the values furnished by author as well as FID values for few other state-of-the-art image super resolution techniques. Results are shown in Table I

Method	FID
SRGAN	104.8
CycleGan	19.01
DeepDeblur	294.96
Wavelet-SRNet	149.46
FSRNet	157.29
Author's	14.89
Our	94.83

TABLE I
FID BASED COMPARISON(ABOVE FID VALUES ARE TAKEN FROM [1])

To test the universality of the model, we trained our model in the similar fashion on celeb-A and Widerface datasets and tested it on IMFDB dataset. The results of this experiments are shown in figure 9.

Fig. 9. Results of experiment 2 (1.Real Image 2.Generated Image)

In our third experiment, we have used only Low-To-High GAN model. It is trained on paired dataset of high-resolution images from celeb-A dataset and the corresponding low-resolution image is generated using bilinear down sampling. We used the same loss function which is proposed by author and tested it under the same testing conditions and test images.

V. DISCUSSION

A. Limitations

The author has not published a complete source code which makes it difficult to know the exact setting and structure of model. As a result, we might not be able to re-implement the technique as done by the author.

Another limitation is related to materializing the stability of GAN. There are some techniques that help deal with instability issue. For example, in order to improve stability, the author used spectral normalization, a weight normalization method, that normalizes weights to satisfy Lipschitz constraint. This project does not make use of it due to time and computation constraints.

VI. CONCLUSION AND FUTURE WORK

We reimplemented the GAN models for super resolution to validate the authors claim. In addition, we tried to use only low-to-high GAN model with author's loss function on artificially generated paired dataset by using bilinear down-sampling. We also tested the model on different datasets (IMFDB) to verify the universality of the model. We believe we can use this model for other applications by training model on different data sets.

Super Resolution is an important topic in the field of Computer Vision. The two GAN model approach has paved a way to super resolve real world images with ease. This approach can find application in many purposes like CCTV surveillance, Medical & Geology Imaging Processes and to improve photo quality in the mobile devices embedded with a low quality hardware.

For future work, researchers can test the model with other datasets. Another scope is changing the architecture of High-to-Low and Low-to-High model. For example, redesign the skip structure, try different number of channel or kernel size, or simplify the model by eliminating trivial parameters or weights. Finally, we can explore whether better control can improve our model. For example, the current model uses encoder-decoder like structure, whether cycle GAN architecture can provide more information of image characteristics or help us capture specific features.

VII. Acknowledge

Thanks for AIs of computer vision course giving us lots of valuable and timely assistance in implementation. Thanks to Professor Kim and Yi-Lin Lee, a math PhD student, for helping us understand the intuition of loss functions.

References

[1] Yang J. Bulat A. and Tzimiropoulos G. "To learn image super-resolution, use a GAN to learn how to do image degradation first". In: *"ECCV"* ("2018"). URL: http://arxiv.org/abs/1807.11458.

[2] Chao Dong et al. "Image Super-Resolution Using Deep Convolutional Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38 (2016), pp. 295–307.

[3] Kaiming He et al. "Identity Mappings in Deep Residual Networks". In: *ECCV*. 2016.

[4] Martin Heusel et al. "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium". In: *NIPS*. 2017.

[5] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. "Accurate Image Super-Resolution Using Very Deep Convolutional Networks". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 1646–1654.

[6] Ziwei Liu et al. "Deep Learning Face Attributes in the Wild". In: *Proceedings of International Conference on Computer Vision (ICCV)*. 2015.

[7] Parisa Beham Shankar Setty Moula Husain. "Indian Movie Face Database: A Benchmark for Face Recognition Under Wide Variations". In: *National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*. Dec. 2013.

[8] Shuo Yang et al. "WIDER FACE: A Face Detection Benchmark". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.