Image Super-resolution using Generative Adversarial Networks

PG-Independent Study Report

Ankit Pant
2018201035
M.Tech. Computer Science & Engineering
International Institute of Information Technology, Hyderabad
Supervised by: Dr. Pawan Kumar

***Abstract*—Artificial Intelligence (AI) and Machine Learning (ML) have taken the world by storm. AI and ML are being applied to new applications at a very fast rate. One such application is Image Super-resolution. This application involves enhancing the details in an image and converting a low-resolution image to a higher resolution image. Generative Adversarial Networks (GANs) are novel models that show a lot of promise in such applications. The PG-Independent Study undertaken attempts to explore GANs and apply them to the field of image super-resolution. Images of people (portraits, selfies, etc.) are used as datasets and the study attempts to enhance such images.**

***Index Terms*—Image Super-resolution, Generative Adversarial Networks, image-enhancement, portraits, GANs**

I. Introduction

Artificial Intelligence (AI) and Machine Learning (ML) has come a long way since their inception. Various models, finding various applications have been developed and have also increased tremendously in their complexity. The greatest advantage of this rapid development has been the application of AI and ML in various fields, from Computer Vision to Speech Processing, from Expert Systems to Natural Language Processing.

Among the various models developed, Generative Adversarial Models (GANs) are of significant interest. GANs work on the premise of having two models that are competing with each other in a form of a game. GANs have come a long way since they were proposed in 2014 by Goodfellow et al. [1]. One of the major problems with GANs had been the slower learning rate. This can be circumvented by pre-training the generator and the discriminator.

GANs have been used in a variety of applications. They fare well in tasks that require creation and enhancements. One such application is *image super-resolution* which involves enhancing low resolution images and converting them to higher resolution images containing more detail. With the amount of data being generated, it is critical for various services to compress the data. Such compression leads to loss of detail and once at source, this ill effect of compression can be neutralised by using image super-resolution at source.

This PG-Independent Study aims to cover Generative Adversarial Networks and Image Super-resolution (in moderate detail) and create a ML model that can enhance portrait images (including selfies) of (majorly) human subjects. A web application has also been created (using Python) that allows the user to input a low quality image and enhance it. Further details can be found in the subsequent sections.

II. Literature Review

A. Generative Adversarial Networks (GANs)

Generator Adversarial Networks (GANs) are ML models that are composed of a *generator* and a *discriminator* working in tandem to learn the features of the task given to it and has a variety of applications [2]. The *generator* model that attempts to recreate the ground truth (often from random noise) and the *discriminator* (or critic) that attempts to distinguish between the ground truth and the output of the generator. Both the generator and the discriminator may be pre-trained to improve performance. GANs work by repeatedly running a training loop which can be loosely described as follows:

1) The generator attempts to create the ground truth.
2) The discriminator takes this generated image and compares it to the ground truth
3) Repeat 1. based on the feedback from 2.

Ideally the training loop needs to be stopped when the discriminator outputs 0.5 i.e. it can no longer differentiate between the ground truth and the generated image.

Multiple variations of GANs have been developed including Deep convolutional GANs (DCGANs) [3], Wasserstein GAN (WGANs) [4], and Softmax GAN [5].

B. Image Super-resolution

Image super-resolution [6] can be simply defined as conversion of one or more low resolution images to a high resolution image. The resultant images are larger and have more detail. This technique can be used to enhance videos as well.

1) Techniques: The various techniques for image super-resolution includes:

- *Single Frame Super Resolution:* This method involves smoothing, interpolation and sharpening to estimate the details that are not present.
- *Multi Frame Super Resolution:* This method involves using multiple low-resolution images of the same scene

albeit each of those images taken from different angles (shifted with sub-pixel precision). These images are then combined to get an enhanced image.

2) Applications: The various applications of image super-resolution includes:

- Enhancing photographs and self-portraits of people
- Enhancing surveillance footage
- Enhancing medical diagnostic images
- Enhancing astronomical and remotes sensing images
- Enhancing low resolution videos

III. Methodology

Image enhancement is attempted during the course of the study using a Generative Adversarial Network. The approach consists of these steps:

- Creating and training the ML model
- Creating the Web application to use the model

The ML model has been created in Python using fast.ai libraries (that in-turn uses Pytorch) [7]. The code is in the form of Jupyter Notebook and that allows for easy reproduction. Google Colab was used as the platform to train and test the model. The ML model currently can scale low-resolution images to a resolution of $512 * 512$ pixels. This limitation is due to the computing resources that were available during the training of the model. The model can be easily extended to upscaling to higher resolutions in the presence of more powerful hardware.

The Web Application has been created in Python and is run by deploying it in a Docker container (environment). Docker ensures that all the necessary libraries and dependencies are installed in the first run of the application.

The following sub-sections provide further details.

A. Overview of the algorithm

The algorithm to create the ML model consists of the following steps:

1) The high resolution images ($1024 * 1024$ pixels) [8] are first taken and converted to low resolution images ($128 * 128$) pixels. The high resolution images act as the ground truth and the low resolution images act as training data.
2) The generator model is then individually trained on this training dataset. The resultant generated images are then saved.
3) The discriminator model is then individually trained on the generated images and original images so that it can learn to determine the generated images from original images.
4) Both the generator and discriminator are then combined as a GAN and trained.
5) The generated images obtained from the GAN are then saved for comparison with original noisy images. The trained model was saved as well.
6) Another step to upsample the images was applied that converts the images to the resolution of ($512 * 512$ pixels). This model was also saved.
7) A test dataset [9] consists of low resolution images ($306 * 306$) pixels and was used to visually see the output of the model.
8) The saved models were then imported to the web application as used to enhance the images provided by the user.

B. Architecture of the ML model

The ML model consists of three sub-models. The generator, the discriminator, and the up-sampler. All the three models are based on the models created in the fast.ai library [7].

The *generator* model is based on the UNet model [10]. It uses Resnet34 [11] as the base architecture and Mean Squared Error (MSE) [12] as the loss function. This model is pre-trained for 7 cycles with varied learning rated (from 10^{-5} to 10^{-3}).

The *discriminator* model uses the basic *gan_critic* provided by the fast.ai library [7]. It uses Cross Entropy [13] as the loss function and is pre-trained for 8 cycles with varied learning rated (from 10^{-6} to 10^{-3}).

The *up-sampler* model is based on the UNet model [10]. It uses VGG-16 [14] as the base architecture and Mean Squared Error (MSE) [12] as the loss function. This model is pre-trained for 11 cycles with varied learning rated (from 10^{-6} to 10^{-3}).

There were certain issues with saving the aforementioned *up-sampler* model to be used with the web-application. Hence another up-sampler model was created to be used specifically for the web-application. This model was created using the upsampling model available in the fast.ai library [7] with modifications. However, this model does not perform as well as the aforementioned up-sampler model based on UNet.

The architecture can be summarized in the figure 1.

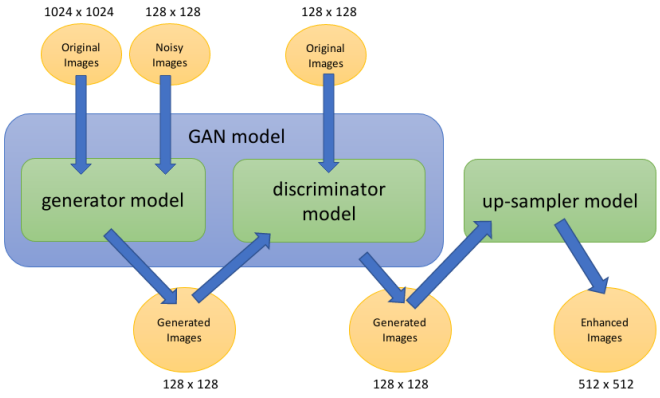

Fig. 1. Architecture of ML model

C. Training and Test Datasets

Different open source datasets were used for training and testing purposes.The details of the datasets is as follows:

1) Training Dataset: The training dataset comprised of 1600 images from the Flickr-Faces-HQ Dataset. It contains high quality images and was created originally as a benchmark for GANs [8].

2) Test Dataset: The test dataset comprised of 300 images from the Selfie Data Set [9]. It contains low quality images.

D. Measures to compare images

Different measures were used to compare the enhanced images to the original images as well as the low-resolution images to the original images. The measures used were:

1) MS-SSIM: Multi Scale Structural Similarity (MS-SSIM) index is used to predict the perceived quality of images and videos [15]. The values of MS-SSIM range from $(-1$ to $1)$ with 1 only possible in-case of identical images. Higher values indicate better similarity and hence less error.

2) MSE: Mean Squared Error(MSE) is used to mathematically differentiate two images by comparing the value in each pixel [12]. The closer the values are to 0, the more identical the compared images are, with 0 being identical images.

3) PSNR: Peak Signal to Noise Ratio (PSNR) can also be used to compare two images. In general it is described as the ratio between the maximum possible power of a signal and the power of corrupting noise [16]. Higher values of PSNR indicate higher fidelity and hence less error.

4) L1: L1 also know as Least Absolute Deviations (LAD) or Least Absolute Errors (LAE) [17], [18], is a regularization technique that can be used to compare images. The closer the values are to 0, the more identical the compared images are, with 0 being identical images.

E. Creating the Web Application

The web application was created using Python and was deployed using Docker. The web application allows the user to pick a low-resolution image and enhances it to an image of resolution $(512 * 512$ pixels). The images 2 and 3 shows the user interface for the web application.

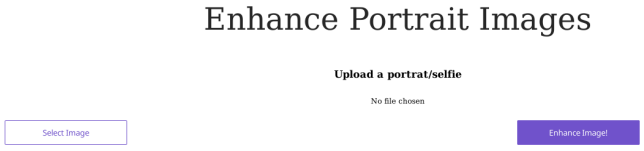

Fig. 2. Web application UI (no image chosen)

IV. Experimentation

The ML model was trained in Google Colab. The specification of the hardware is as follows:

- RAM: 25.51 GB
- VRAM (GPU Memory): 12 GB
- Disk Space: 358.27 GB

The approximate training times of various models are given below:

- Generator model (individual): 13 minutes
- Discriminator model (individual): 18 minutes

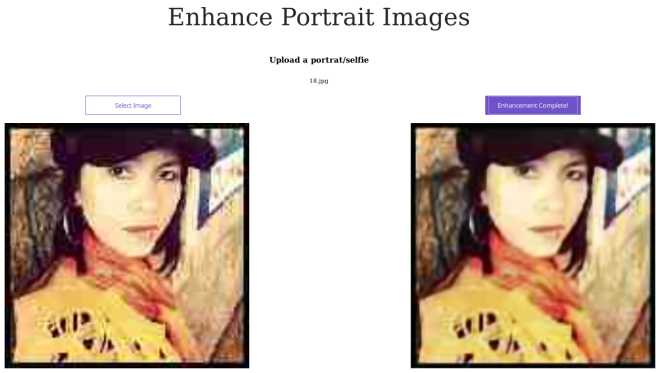

Fig. 3. Web application UI (after enhancing image)

- GAN: 48 minutes
- Up-sampler: 135 minutes

After the ML models were trained, they were saved so that they could be used to make prediction on test dataset or other unseen data.

V. Results

Two results were obtained during the study. One was the comparison between the low resolution images and the images obtained by the pre-trained generator (before applying GAN model), with the original images, the results of which are given in table I. The second was a comparison between the low resolution images (directly resized) and the enhanced images, with the original images, the results of which are given in table II.

TABLE I
Comparison between low-res and initially generated images

	Image type	average SSIM	average MSE	average PSNR	average L1
1	Low-res Image	0.127138	25652.102085	4.295813	3.603803e+06
2	Generated Image	0.135904	25757.227375	4.304846	3.611744e+06

TABLE II
Comparison between low-res and enhanced images

	Image type	average SSIM	average MSE	average PSNR	average L1
1	Noisy Image	0.308175	25258.535317	4.370232	5.720727e+07
2	Enhanced Image	0.340402	24976.292169	4.456091	5.675491e+07

It can seen seen from tables I and II that both the generated images and the enhanced images have less error than the low resolution images. The enhanced images have much greater similarity to the original images than the low resolution images (that were directly resized).

Some of the enhanced images can be seen in the figures 4 and 5.

Fig. 4. Results of image enhancement (vgg)

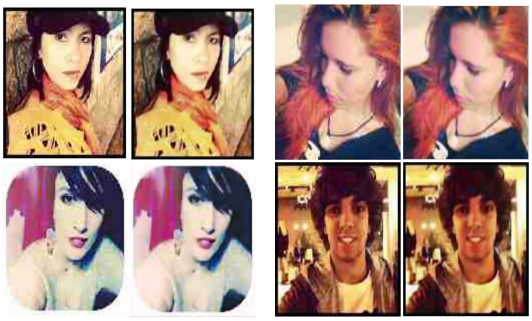

Fig. 5. Results of image enhancement (web-app) (zoom in to view difference)

VI. Conclusion

Artificial Intelligence (AI) and Machine Learning(ML) have made a lot of progress and in many areas AI and ML models have even managed to outperform humans. With active research in this field, the performance of models is bound to improve further.

The independent study undertaken manages to use such developments in ML to apply it the problem of enhancing low resolution images. The model was trained on dataset containing images of people and was able to perform reasonably well in the low resolution images it was tested with. With improvements to the model, it can also find commercial application which can be run in smartphones. A lot of users have their images stored in the cloud. The service providers tend to use compression to save space, sacrificing some quality in this process, so when the user downloads her/his images locally, super-resolution algorithms can be run to convert them back to high quality images.

The study concluded with the creation of ML models that can be imported by other applications to directly perform image enhancement or can be used by other models through transfer learning. A basic web application was also created that can be used to perform image enhancements by a user.

VII. Drawbacks

Some of the drawbacks of the current implementation include:

- The algorithm current can output to a maximum resolution of $512 * 512$ pixels. It is due to lack of computational resources during training.
- The algorithm only works for low resolution images (preferably below $384 * 384$ pixels). Giving an image of larger resolution $> (512 * 512)$ will result in the output image having lower quality.
- The web application is not able to use the up-sampler based on VGG architecture. Hence it is not as accurate and can lead to over-softening of images which look less sharp.
- Since the model was trained on a relatively small dataset (due to hardware resources), it may not output good quality output for images of certain scenes (e.g. overexposed scenes).

VIII. Future Score

The current implementation may be improved in the following way:

- The output resolution of images may be increased on more powerful hardware.
- The training loop can be extended to more cycles that may result in better output images.
- The size of the training dataset can be increased for better learning by the model.
- The underlying architectures (e.g. VGG16, ResNet34) can be replaced with more extensive architectures (e.g. VGG19, ResNet101, DenseNet161, etc.). This may result in better output with sharper images. However these architectures would require more powerful hardware to train.
- The web application that was created has been deployed locally. For better reachability, it can be hosted in Cloud Services like Amazon Web Services (AWS) and made public for anyone to use on the internet.

Apart from modifications to the existing model, other models can also be explored that may or may not use GANs. Some models that provide better results include:

- Using Perceptual Loss Functions as described in the paper by Christian Ledig et al. [19].
- Use Automated Texture Analysis as described by the paper by Mehdi S. M. Sajjadi et al. [20].
- fastai super resolution implementation using VGG16 and having multiple training loops, each time with images of higher resolution, i.e. first training with images of size $(64 * 64)$, followed by images of size $(128 * 128)$,

followed by images of size $(256 * 256)$, and so on. An implementation of the same can be found here.

ACKNOWLEDGMENT

I express my gratitude to Dr. Pawan Kumar for providing with the opportunity to take PG-Independent Study under his guidance. During the course of the study, Dr. Pawan provided ample freedom to explore the field of study apply my own approach. Dr. Pawan also provided necessary guidance when required. The independent study helped me learn more about Generative Adversarial Networks, how they can be implemented and how to use ML models with more traditional programs like web applications.

REFERENCES

[1] Goodfellow, Ian J. and Pouget-Abadie, Jean and Mirza, Mehdi and Xu, Bing and Warde-Farley, David and Ozair, Sherjil and Courville, Aaron and Bengio, Yoshua, "Generative Adversarial Nets", arXiv e-prints, https://arxiv.org/pdf/1406.2661.pdf

[2] "Generative Adversarial Network", Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Generative_adversarial_network

[3] Alec Radford, Luke Metz, Soumith Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks", arXiv e-prints, https://arxiv.org/pdf/1511.06434.pdf

[4] Martin Arjovsky, Soumith Chintala, Léon Bottou, "Wasserstein GAN", arXiv e-prints, https://arxiv.org/pdf/1701.07875.pdf

[5] Min Lin, "Softmax GAN", arXiv e-prints, https://arxiv.org/pdf/1704.06191.pdf

[6] Kevin Su, "Introduction to Image Super-resolution", http://www.cs.utsa.edu/~qitian/seminar/Fall04/superresolution/SR_slides_xsu.pdf

[7] Jeremy Howard et al, "fastai", https://docs.fast.ai/, https://www.fast.ai/

[8] NVlabs, "Flickr-Faces-HQ Dataset (FFHQ)", https://github.com/NVlabs/ffhq-dataset

[9] Center for Research in Computer Vision, "Selfie Data Set", University of Central Florida https://www.crcv.ucf.edu/data/Selfie/

[10] Olaf Ronneberger, Philipp Fischer, Thomas Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", arXiv e-prints, https://arxiv.org/pdf/1505.04597.pdf

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun , "Deep Residual Learning for Image Recognition", arXiv e-prints, https://arxiv.org/pdf/1512.03385.pdf

[12] "Mean squared error", Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Mean_squared_error

[13] "Cross entropy", Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Cross_entropy

[14] Neurohive, "VGG16 – Convolutional Network for Classification and Detection", https://neurohive.io/en/popular-networks/vgg16/

[15] "Structural similarity", Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Structural_similarity#Multi-Scale_SSIM

[16] "Peak signal-to-noise ratio", Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio

[17] "Least absolute deviations", Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Least_absolute_deviations

[18] "L1 norm", Wikipedia, the free encyclopedia, https://en.wikipedia.org/w/index.php?title=L1_norm&redirect=no

[19] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network", arXiv e-prints, https://arxiv.org/pdf/1609.04802.pdf

[20] Mehdi S. M. Sajjadi, Bernhard Schölkopf, Michael Hirsch, "EnhanceNet: Single Image Super-Resolution Through Automated Texture Synthesis", arXiv e-prints, https://arxiv.org/pdf/1612.07919.pdf