Augmentation of Images through DCGANs

Himanshu Arora, Samyak Jain, Sanket Anand, Dharmveer Singh Rajpoot
Department of Computer Science and Engineering / Information Technology
Jaypee Institute of Information Technology, Noida
himanshua85@gmail.com, samyakjn20@gmail.com, sanketanand62@gmail.com, dharmveer.rajpoot@jiit.ac.in

***Abstract* — Now-a-days, extending images has become a challenging task to implement. Many of algorithms like convolution neural networks (CNN), Generative Adversarial Network (GAN) are used to fill out the image spaces. But the challenge arrives to guess or make appropriate assumption of image extended borders. We used Deep Convolution Generative Adversarial Network(DCGAN) over GAN and CNN to implement it.[1,2] GAN's are implemented to guess the space in image by generating fake examples using generator and decides using discriminator which determines how much the image is real. DCGAN is improved version of GAN which generates spatial correlations [5].**

Keywords – Image, Convolution neural network, Generative Adversarial Network, Augmentation etc.

I INTRODUCTION

Images may get damaged or some portions of the image may get lost or deteriorated. And if these images make up the valuable ones, it becomes very important to restore the damaged image in a form such that it looks like an original image.

Image inpainting or restoration is the process which helps to restore the corrupted parts of the image.[7,8,9]This process helps in filling up of missing regions or cracks , removing red eye effect, removing stamped date from images and may even remove an existing object from an image. GANs and CNNs are basically the two methods which make image inpainting possible.

A lot of work has been done in this field and a lot of algorithms have been proposed for the same. Another process which is Image Outpainting or ex- trapolation is very less explored. We fill in missing pixels in case of image inpainting but in case of outpainting, we have to generate thousands of new pixels which must look similar to original image. Therefore, they are completely different processes. In contrast to Image Inpainting, Image outpainting is the expansion of an image with no information available about its neighborhood. [11] It is quite difficult as the expanded output images must look realistic. This technique finds many uses such as creation of panaroma image, vertically filled expansion of videos and creation of different textures and patterns.

This paper proposes a deep learning approach to expand an image beyond the image boundaries. We have used GANs to implement image extrapolation. Broadly, we aim to use a DCGAN architecture as it allows outpainting of the image recursively up to some extent and in this way we are able to obtain larger realistic images.

II LITERATURE SURVEY

Image in painting or filling the spaces present in images has counter the good techniques and there several ones with high efficiency to provide optimal results. Throughout the years it has been implemented by using various algorithms. Convolution neural network and Generative adversial networks are one.[13,14] Firstly, CNN was used to fill up the space in image in one of research paper, although results were comparatively realistic but had a space for improving visual efficiency, but as for extension of image there is very less research done or existing algorithm model didn't match up the minimum efficiency that should back away the algorithms. The idea behind extending the image is to train our data efficiently and more realistic. Some of the researchers tried to implement it by graph matching but has a problem of color variations and looks virtually. Graph matching method include sex trapolation function, transforming optimizer, wrapping and stitching.[20] It works upon top, left and right direction. Extending the images cannot be always correct as there very less probability of introducing same results virtually in comparison of realistic, but as to counter it DCGAN works on several features that includes

1. Manipulating brightness using pure black as the degenerate image also Interpolation makes image darker whereas extrapolation brightens it. In both versions, bright pixels are more affected.
2. Contrast by taking a gray image which is constant throughout, altering saturation by taking black-&-white image as degenerate version.
3. Sharpening or blurring the image Combined operations- it includes various features such as luminance, tint, saturation.

DCGAN is significant due to many reasons that are as follows:

It can generate representation from under-training images, unsupervised. It is proficient in generating the realistic images

As for the previous work not much research has been done has the topic of image extension has been considered more unrealistic as compare to image in painting. [17, 18]So as to no algorithm can achieve very

978-1-7281-3591-5/19/$31.00 ©2019 IEEE

realistic result but as compare to others DCGAN proved to be of great beneficial and provides closely realistic result.

III PROPOSED ALGORITHM

We have proposed an algorithm which is a blend of both contextual and perpetual information. Con- textual information gives knowledge about the pixel with the help of surrounding pixels. And, perpetual information is based on information of other images or real life images. [12]We have used machine learning to implement this algorithm. Using probability distribution function (PDF) for completing images that is using the image with maximum probability is not feasible for compound distributions. We have made a Generative Adversarial Network (GAN) which side by side trains two neural networks. First is Discriminator which differentiates an image whether it is real or fake. An image is given as an input to the discriminator and it tells whether it is real or not based upon a scalar value. A value closer to 0 shows that image is real while a positive value differentiates as fake. Second is Generator which generates images to make the discriminator learn properly to differentiate images correctly with precision. During training, a real image is shown to discriminator which differentiates it as a value closer to 0. Then an image produced by the generator is given as input to the discriminator. Upon receiving this image, discriminator adjusts itself to a higher output. But at this point of time, generator deceives discriminator making it to think that image produced by generator is not fake.

Our dataset is places 365 standard dataset and consist of 36000 images(256 x 256) which are down sampled to 128 x 128. Given an image Ig, we have to generate an image Ig' such that Ig lies in middle of Ig' and the complete image looks like an actual image.

We have used deep neural network to define a function F(Z) which takes in a vector as input and outputs a picture. The training of DCGAN model is done.

```
def F(Z):
...
Return pictureSampleZ = np.
   random.uniform (-2, 2,
   10) pictureSample = F(Z)
```

Code Snippet 1: Code snippet for our function F(Z)

We have defined some parameters:
pd- distribution of our image dataset
pZ- distribution over Z
pG- distribution from where generator produce images

Our discriminator (image differentiator) return a value greater than 0 and close to 1 if the image comes from pd and value approximated around 0 in case there is a fake image (image produced from pG). The main goal of training is to augment value produced by discriminator if the image produced belongs to pd and to miniaturize value if the image is not from our dataset distribution. Also, we need to train our generator in order to serve its purpose of fooling our discriminator, D. Our generator outputs an image which is then fed to discriminator, D.

$$minGmax\ DEZ \sim pdatalogD(Z) + Ez \sim pz[log(1 - D(G(z)))]$$

Training part of discriminator and generator is done using the gradients of the above expression. For quick computation, we divided our respected outcomes into small batches of size M and inner augmentation done only 1 time for training part.

Steps for algorithm used are:

```
FOR n iteration:
  FOR I steps:
1. Make small batches of (M) samples prior P(Z).
2. Make small batches of (M) samples from probability distribution Pd.
3. Update discriminator by updating its stochastic gradient.
  END FOR
4. Sample small batches M from noise prior Pg.'
5. Downgrade the discriminator by decreasing its stochastic gradient.
END FOR
```

The flow chart for the algorithm is shown in Figure 1:

IV EXPERIMENTAL RESULTS

In order to get proper results, we augmented 3500 images to 10500 images. We have used batch size of 16. Augmentation of images is necessary to get better results.

Some of the snapshots of our images during the process of training are shown in Figure 2. There are three parts in this figure: First part is the original image, second is the input image and third image which you can see is the predicted image from our model. After successfully training our images, the predicted image (third part) is almost similar to the original image (first part). The results are shown in Figure 3 and Figure 4.

We also implemented recursive painting, which is shown in Figure 5.In this, we recursively extend our image by giving the predicted image to the network again and again resulting in its expansion up to a width of 3.5. In Figure 5, you can see how recursive painting is done during training [10].

Some of the snapshots of the expanded image after training are shown in Figure 6 and Figure 7.

We then tried on other beach images. The results were pretty good and our model was successfully able to expand images up to a width of 3.5.In Figure 6 and 7, One can see how accurately it is predicting and augmenting images. This can be used in various applications like interior designing, exploring missing facial expressions. Also it can be used to expand clues found during crime scene.

V CONCLUSION

With the help of deep learning approach , we were successfully able to augment the images with a realistic output. In order to provide stability to the images , three phase training was implemented with the GAN algorithm. Further to improve quality of an image , we were able to apply receptive field using the dilated convolutions and it made the augmented image look more realistic. At the end ,we tried to apply recursive outpainting in order to extend our image . Although we still got a realistic image but noise was compounded in the image.

Image outpainting though less explored has infinite possibilities in the future. Not only the image but even a video can be outpainted. For this, sequenced models need to be explored. Stability of an outpainted image is the main aim and therefore, we will try to implement Wasserstein GAN along with the three phase training. One can also try to implement partial convolutions , an approach used for image inpainting, to our model in the future. Further , we might experiment perceptual and style loss incorporation into our training model which might improve the overall performance of the model.

REFERENCES

[1] Konstantinos Gavriil , Georg Muntingh, and Oliver J. D. Barrowclough, “Void Filling of Digital Elevation Models with Deep Generative Models” in arXiv preprint arXiv:1811.12693, 2018 .

[2] Kamyar Nazeri , Eric Ng , Tony Joseph , Faisal Z. Qureshi, and Mehran Ebrahimi , “EdgeConnect: Generative Image Inpainting with Adversarial Edge Learning” in arXiv preprint arXiv:1901.00212 .

[3] Qiqin Dai , Henry Chopp , Emeline Pouyet , Oliver Cossairt , Marc Walton , and Aggelos K. Katsaggelos, “Adaptive Image Sampling using Deep Learning and its Application on X-Ray Flu- orescence Image Reconstruction” in arXiv preprint arXiv: 1812.10836 .

[4] Yi Wang, Xin Tao , Xiaojuan Qi1 Xiaoyong Shen, and Jiaya Jia1, “Image Inpainting via Genera- tive Multi-column Convolutional Neural Networks” arXiv preprint arXiv: 1810.08771

[5] Bolei Zhou , Agata Lapedriza , Aditya Khosla , Aude Oliva , and Antonio Torallba , “Places: A 10 Million Image Database for Scene Recognition” in IEEE conference , 2011.

[6] Mark Sabini , and Gili RusakPainting, “Out- side the Box: Image Outpainting with GANs” in IEEE conference , 2018.

[7] Marcelo Bertalmio , Guillermo Sapiro ,Vicent Caselles , and Coloma Ballester , “Image Inpainting ” in IEEE conference , 2014.

[8] Kaiming He , and Jian SunImage , “ Com- pletion Approaches Using the Statistics of Similar Patches” in IEEE Conference , 2014 .

[10] Satoshi Iizuka, Edgar Simo-Seraa, and Hiroshi Ishikawa, “Globally and Locally Consistent Image Completion” in IEEE Conference , 2011.

[11] L. Lorenzi, F. Melgani, and G. Mercier, “Inpainting Strategies for Reconstruction of Missing Data in VHR Images” in IEEE Conference , 2011.

[12] Criminisi, Perez P., and Toyama, K., “Object Removal by Exempler-Based Inpainting” in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003.

[13] Homan Igehy, and Lucas Pereira , “Image Replacement through texture synthesis” in IEEE Conference, 1997.

[14] Hayit Greenspan, Charles H. Anderson , and Sofia Akber, “Image Enhancement By Nonlinear Extrapolation in Free Space ” in IEEE Transactions on Image Processing , 2000.

[15] Yinda Zhang , Jianxiong Xiao , James Hays , and Ping Tan, “Dramatic Image Extrapolation by Guided Shift Maps ” in IEEE conference, 2013.

[16] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, “Simultaneous structure and texture image inpainting” in IEEE Conference , 2003.

[17] J. Kopf, W. Kienzle, S. Drucker, and S. B. Kang , “Quality prediction for image completion” in ACM transactions on graphcis , 2012.

[18] S. Iizuka, E. Simo-Serra, and H. Ishikawa , “Globally and locally consistent image completion” in ACM Transactions on Graphics , 2017.

[19] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro. Image inpainting for irregular holes using partial convolutions. arXiv preprint arXiv:1804.07723, 2018.

[20] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher., “Simultaneous structure and texture image inpainting” in IEEE Transactions on Image Processing, 2003.

[21] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein GAN” in arXiv preprint arXiv:1701.07875, 2017.

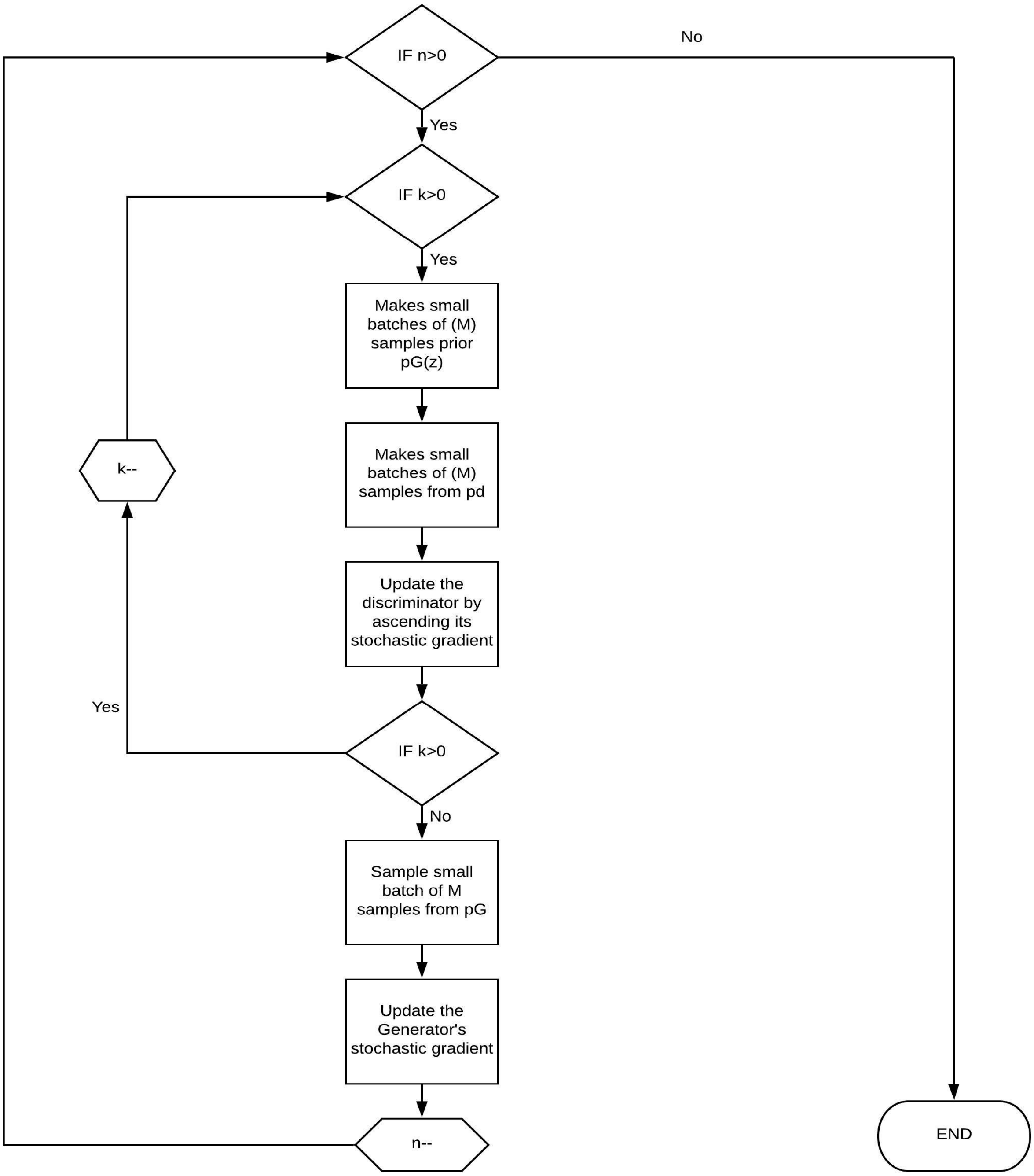

Figure 1: Flow chart for our model

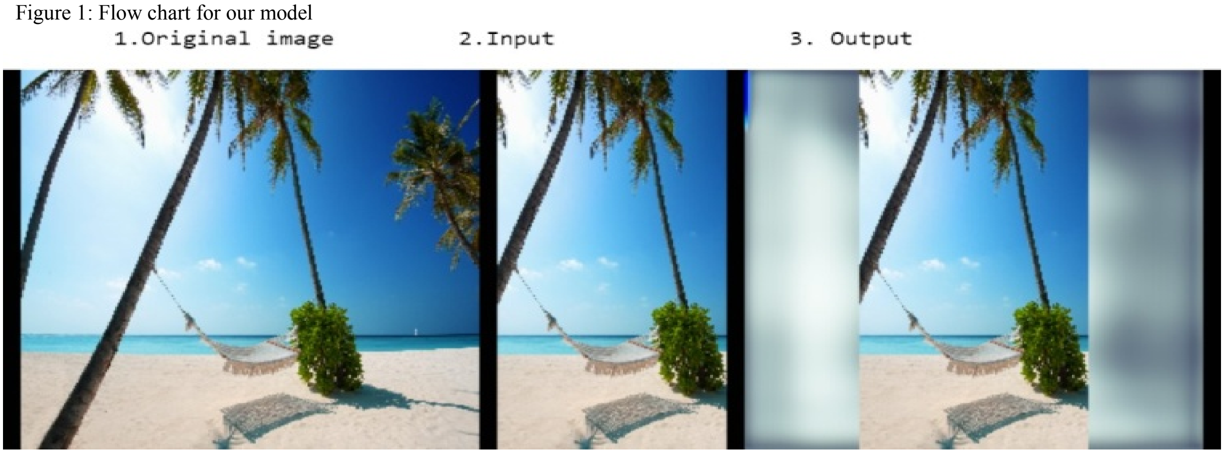

Figure 2: Snapshot of an image during training

Figure 3: Sample image after training

Figure 4: Sample Image example after training

Figure 5: Recursive painting during training

Figure 6: Recursive painting demonstration 1

Figure 7: Recursive painting demonstration