IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 11, Issue 3, March 2023

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 3, March 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1103027 |

Correcting Image Anomalies Using Deep Learning Based On Image Inpainting

Prasanth P. Nair, Mohan S.

Lecturer in Computer Engineering, Government Polytechnic College, Neyyattinkara, Thiruvananthapuram, India

Lecturer in Computer Engineering, Government Polytechnic College, Nedumangad, Thiruvananthapuram, India

ABSTRACT: Image inpainting or completion is an active computer vision problem. Here the damaged, deteriorating or missing parts of the artwork are reconstructed. Image completion tasks typically have the aim to complete missing regions of an image in the most natural-looking way and correcting image anomalies can be done using image completion. This method uses deep learning to recreate the corrupted part of an image. The system is based on Generative Adversarial Networks. It contains a discriminator and an adversarial network which works together to generate the required inpainted image. The generator creates data to fill up the corrupted part or missing part of an image and the discriminator evaluates the generated data. Then calculating the loss from the discriminator classification, weights of the generator are updated. The result is newly generated inpainted images.

KEYWORDS: Information security, Deep learning, Image processing,Generative Adversarial Networks, steganalysis.

I. INTRODUCTION

Image is the one thing that is inevitable in the current digital world and image anomalies are part of a corrupted image. To recreate the original image given the corrupted part, image inpainting is used. An example of image inpainting is shown below in figure 1. Image inpainting is a largely studied problem in image processing and various tools have been brought to serve it over the years. The basic task of image completion[1] or inpainting is to successfully fill the damaged area. There are different existing algorithms used for image completion such as Bertalmio's Algorithm, Oliveira's Algorithm, Criminisi's Algorithm[8] etc. In recent years the study of machine learning and improvement in the research of deep neural network has introduced new ways to solve a lot of problems in computer science such as object identification, image generation, classification of images etc. Here deep learning[3]can be used for image inpainting.

Artists and digital photographic industry uses image inpainting to recreate a missing part of an image or remove a certain part of an image. Without the use of a computer, inpainting is a largely time-consuming process. When the machines are used without deep learning or using a certain straight forward algorithm to fill in the missing region, it is not efficient as the artist does. The reason behind this is existing methods for image inpainting either fill the missing regions by borrowing information from surrounding areas or generating semantically coherent content from region context. To solve this, a neural network is used here, which works like a human mind and learns from experience, thus providing the required inpainting.

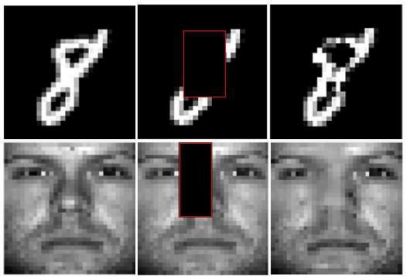

Fig. 1. Examples of image Inpainting

Existing works on image inpainting uses the traditional methods which use the corrupted images masked region to reproduce the missing data. The traditional methods consist of patch-based method which works on filling the image using the surrounding area of the image. These methods don't yield an efficient image completion. Some libraries like

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 3, March 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1103027 |

OpenCV gives techniques to solve image inpainting which is based on fast marching [6] and fluid dynamics. These methods are also not good enough to fill in a large missing part of a complex image. Therefore, here we can use deep learning to solve these issues found in traditional based methods. GAN(Generative Adversarial Networks) is the main component which we use here to create this system. Which uses two deep neural networks to jointly work together to create required output

II. PROBLEM STATEMENT

In the current scenario there aren't many robust algorithms that reproduce the missing part of an image with effectiveness. Therefore, the end goal of this project is to solve this problem by implementing an effective and robust image inpainting system which can inpaint in complex images such as a face image or an asymmetrical image. The system should produce output images such that a casual observer perceives the generated image as a completely plausible image. The input used here is a set of hand written digit images varying from 0 to 9, which is masked in the central part. The output will be images with generated missing part.

Solving the image inpainting problem with forward marching, copying method or by traditional methods are not effective and may not give the required output. For removing all these shortcomings of the existing techniques, a new model is proposed which is based on Deep Convolutional Generative Adversarial Network.

III. ARCHITECTURE OF THE SYSTEM

An architecture of the system gives an outline of the system. It gives the complete requirements of the system.

The general architecture of GAN is shown below.

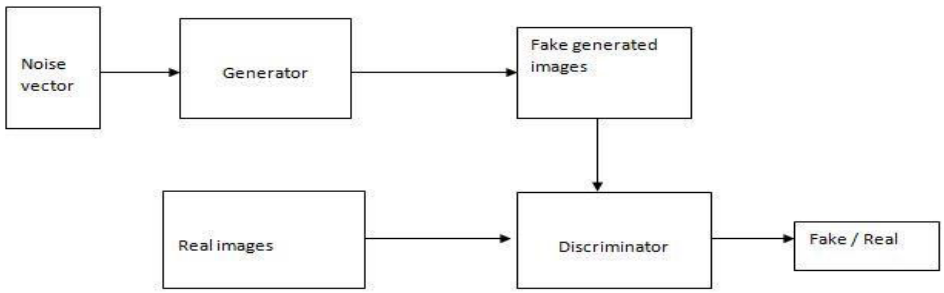

Fig. 2. GAN Architecture.

The architecture has a generator and a discriminator network. The noise vector is given to the generator to initialize the system. The generator is responsible for generating the missing part of the image. The discriminator will evaluate the generated image and real image to classify these images. According to this classification loss, generator will try to fool the discriminator by creating more realistic images.

A. Dataset

MNIST handwritten dataset is used to inpaint in the corrupted region. The dataset consist of different hand written digit images from 0 to 9. It is downloaded directly using Keras library. The images are of 28x28 resolution and each image having an only single channel. There are 60000 training images and 10000 testing images of different handwritten digits. These images are in a Numpy array format. By cutting out the central part of the image, noised training and testing dataset are created. Using these datasets, the networks are trained jointly.

B. Generative Adversarial Networks

It consists of two networks, a generator and a discriminator [2]. The generator is an autoencoder and it has an encoding part and a decoding part. Discriminator on the other hand act as image classifier CNN which points out whether the input image is real or fake. The generator's function is to fool the discriminator.

1) Generator: The generator is an autoencoder. Which takes input as random noise and then turns it into an output which here is the image. In general, the generator creates new dataset with the help of discriminator based on the input dataset. In the encoding part, the generator applies max-pooling which reduces the size of the block by each layer. In the decoding part, the whole thing is reversed using upsampling. The design details are given below. The first three layers are encoding part and then the two layers are decoding part.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 3, March 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1103027 |

- The first block which is a convolutional 2D layer consist of 32 filters of size 3x3 and pooling is average pooling with 2x2 size. Relu activation is used.
- In the second block, there is 64 filters of the same size and using the same pooling and activation.
- Third block, it is also the final layer of the encoder. Consist of 128 filters with the same activation and pooling.
- Each part of the encoder blocks uses batch normalization.
- Decoder has 128 filters in the first layer and with 3x3 size. Unpooling [5] is done using Upsampling2D.
- Second layer has 64 filters which use Upsampling2D to unpool.
- Last layer has 1 filter with 3x3 size and Tanh activation.

2) Discriminator:It is an image classifier with only one single output, which is identifying the given image is from the original dataset or it is from the generated set. The generator will always try to fool the discriminator thus improving the generated image quality. Here the samples from generator input and the original dataset are taken as input to the network and it will classify the input image as real or fake. Using the discriminator loss, the generator's weights are updated. The details of the network are shown below.

- First layer uses 64 filters with size 3x3. It uses LeakyRelu as the activation and a dropout of 20 percent as the regularization technique.
- Second and third layer consist of 128 and 256 filters with the same activation and regularizer.
- Final layer uses sigmoid activation and it is also a dense layer.

Here, we take a closer look at how the system was implemented. The whole system was developed using python language.

IV. DATASET

The MNIST Handwritten digits dataset is loaded usingKeras dataset module, which already comes as vectoriseddata. Then the dataset is divided into a training data set (60000images) and testing dataset (10000 images). Each image is aNumPy array. To simulate the Noised dataset, centre part ofthe original dataset is removed.

A. Pre-processing Data

As above mentioned the data is loaded as training and testing data from Keras library. From these training data, the noised training data is created. Each pixel consists of a valueranging in between 0 and 255. First, the centring of the data is done by subtracting each image (Keras will load the image as a NumPy array) with 127.5 then for scaling each pixel value, it is divided by 127.5. In this way, the dataset values will range from -1 to 1, which helps to improve the stability and performance of the network. To plot the generated image the processed data should rescale the values in the range of 0 to 255, for this purpose a rescale function is also created.

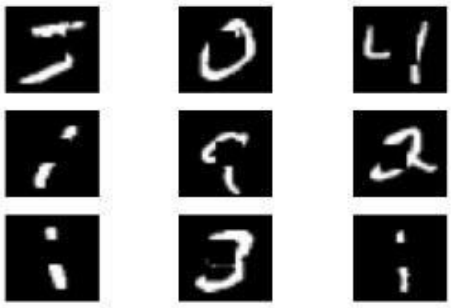

Fig. 3. Noised images

V. DCGAN IMPLEMENTATION

A DCGAN(Deep Convolutional Generative Adversarial Network) [4] is a popular design for creating GAN, which is used in this project. It takes in both generator and discriminator and makes the discriminator non-trainable while

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 3, March 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1103027 |

training the generator. This model pipes the input generated from the generator to discriminator and that forms DCGAN architecture.

A. Generator and Discriminator Network

The generator will get a random noise vector (1-dimensional vector) as input in the first place then it will generate an entirely new output image accordingly. The output from this generator can be called as a fake image. This set of fake images and real images (images from the original training dataset) are given input to the discriminator. The discriminator will try to classify the images as real or fake. Thus the discriminator classification loss will be calculated and backpropagation works through both discriminator and generator to create gradients. Then the gradients are used to update only the weights of the generator. Therefore, helping the generator to create more realistic output.

V. TRAINING

After implementing the generator and discriminator as above-mentioned ways these two networks will be loaded by calling each function. After that, both generator and discriminator will be fed through the DCGAN which will be compiled using binary cross-entropy loss and using adam optimizer. The 128 original images and corresponding corrupted images are created. Then the corrupted image batch which will be feed through the generator network to get new fake image set. After that the original set and generated fake set will be concatenated so that the first set is original and the second set is all fake. The generated images will be labelled as 0 and the original image will be labelled as 0.9, it is given 0.9 for label smoothing which makes the adversarial network resilient. Discriminator will get the 256 images as the input and will try to classify which is real or fake. Generator and discriminator should not be trained at the same time. To implement this, when the generator is training, a boolean data type is used to make sure that the discriminator is not training. Both generator and discriminator uses Adam optimizer because it gives more efficient and less time-consuming results.

VI. TESTING

To test the model, we use the noised testing data with a set of 10000 images each having size 28x28. Here data will be called upon the generator to create a new set of images which is not in the original set. Then the set of new data generated will go through an image classifier which is a normal classifier created for predicting the 10 digits. Then the prediction on the generated data is taken to find the accuracy of the generated data. To implement this an image classifier is made using three convolutions with 32,64,128 filter and the two dense layers. Each layer uses Relu activation and final layer is a softmax activation to give probability distribution of the digits.

VII. INPAINTING MNIST HANDWRITTEN DIGITS

In this section the digits are inpainted using the layered convolutional network which is the generator. Below figure 4 shows the output that got in the sixth epoch of training.

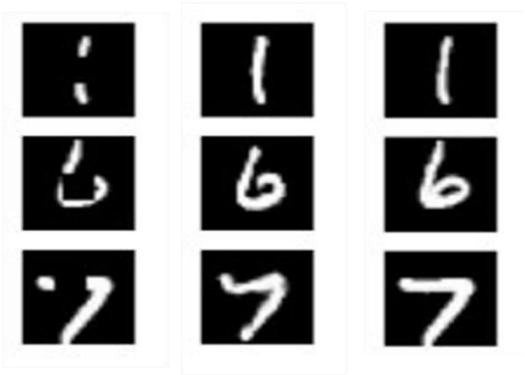

Fig. 4. Output

The first column shows the noised images, second column shows the generated images and the third column shows the original images. The generated images will pass through an image classifier network to check the accuracy of the generated data. Here in this system the accuracy was 87.7 percentage. Comparing the output generated by FMM(Fast

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 3, March 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1103027 |

Marching Method) [6] and a GAN shows that FMM creates blurry and noised images which implies that it cannot createdetail because it works only on the information from the surrounding pixels of the mask.

Fig. 5. Result plotted

VIII. DISCRIMINATOR AND ADVERSARIAL LOSS

The losses plotted using matplotlib is shown below. Here we can see that the discriminator loss on real and fake images after 1500 iterations are in between 0.5 and 0.7 which shows the stability of the GAN (According to observation in a stable GAN training, the discriminator loss is always around 0.5). Even though the network is stable there is still images generated by this which looks like random noise. The given graph shows the adversarial loss at peak in the first few iteration, which shows that the system creates unrealistic outputs or random noise images in the first few iterations. The loss of generator (adversarial loss) is reduced as iteration is increasing, thus shows the improvement in the output inpainted image.

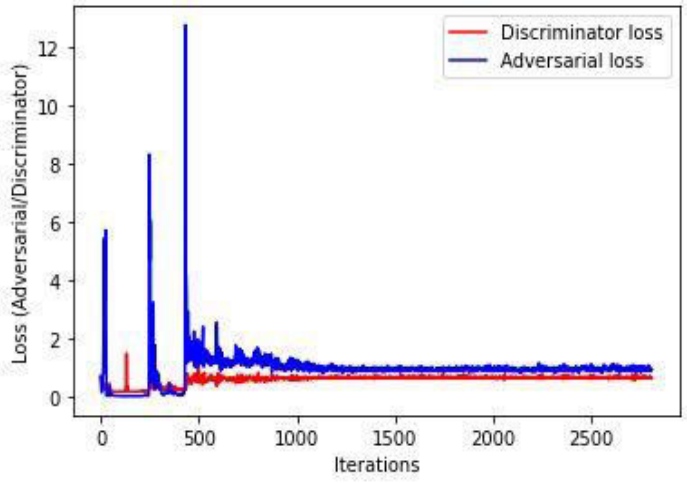

Fig. 6. Discriminator and adversarial loss during training

IX. FAILURE CASE

This model has some realistic digit output, but there are limitations to this model. Sometimes it lacks to generate relevant data in the missing parts. The figure 7 below shows a failure case were the digit 8 is not successfully inpainted. However 10000 images of handwritten digits are used to test and 87.7 percentage of testing images were predictedsuccessfully.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 3, March 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1103027 |

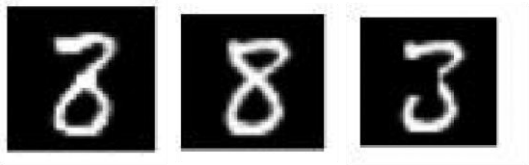

Fig. 7. Generated, Original, Masked image

X. LIMITATIONS

- Vanishing gradient problem: If the discriminator is optimal at classifying real and fake images it cannot provide enough data to improve the generator.
- Controlling GAN is difficult if the network is not tweaked in the right way.

XI. CONCLUSION

Recent developments in digital media steganalysis clearly indicate the immense importance of accurate models that arerelevant for steganalysis. This method gives a new way to solve this steganalysis problem by taking an assumption thattwo regions of a normal image might possess similarities in the noise ratio. Thus the relationships between two image subregion can be employed in an effort to improve steganographic feature distinction. The project proposes a new hybrid network which can work with more information by incorporating TLU activation in the first layer of Siamese network. By using this method performance improvement is achieved. For future work, the arbitrary sized image steganalysis can be improved by stretching the research on noise dense areas of the image.

REFERENCES

1. Haselmann, Matthias, Dieter P. Gruber, and Paul Tabatabai, “Anomaly detection using deep learning based image completion",2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2018.
2. Jiang, Y., Xu, J., Yang, B., Xu, J. and Zhu, J., “Image inpainting based on generative adversarial networks ". IEEE Access, 8, pp.22884-22892,2020.
3. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., & Huang, T. S. “Free-form image inpainting with gated convolution”, Proceedings of the IEEE International Conference on Computer Vision. 2019.
4. Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks.",arXiv preprint arXiv:1511.06434, 2015
5. V. Turchenko, E. Chalmers, and A. Luczak, “A deep convolutional autoendoer with pooling - unpooling layers in caffee,” 2017.
6. A. Telea, “An image inpainting technique based on the fast marching method,” J. Graph. Tools, vol. 9, no. 1, pp. 23–34, Jan. 2004.
7. M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein GAN," 2017, arXiv:1701.07875., https://arxiv.org/abs/1701.07875.
8. Vreja, Raluca & Brad, Remus. “Image Inpainting Methods Evaluation and Improvement. TheScientificWorldJournal". 014. 937845. 10.1155/2014/937845.

INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.165

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details