Image inpainting using deep learning technique DC-GAN

Sharth Boya(16EC109) ,K Jagadish(16EC223), V Narendra(16EC225), V Reethu(16EC250)
Deapartment of ECE, National Institute of Technology Karnataka,
Surathkal, Mangalore, India - 575 025.

***Abstract*—Image Inpainting is an ancient art and is a research topic in image processing. It fills the corrupted and lost parts of an image in an undetectable manner. Image Inpainting is an ancient method of restoring the images which are damaged due to scratches or are old. Image inpainting finds numerous applications like red eye correction, restoration of old damaged films, compression etc. There are several algorithms proposed for Image Inpainting. In this paper, we discuss about our approach of image inpainting using deep learning. Finding the dataset intially was difficult and since the dataset was huge it took a lot of time to run. We had to try different loss functions to check for the one which gives optimal result. We compare our generated images with the original images using image similarity metrics to check the efficiency of reconstruction of the corrupted image .**

***Index Terms*—General Adversarail Network, Adversarial loss, Masked Image, Generator, Discriminator**

I. Introduction

The purpose of image inpainting is to remove the unwanted sections from an image. The unwanted objects and scratches cannot be just erased. This work can be done manually but this an exhausting job and also creates an odd patch in the image. So the necessity to eliminate this odd patch such that there is no trace of this odd patch brings us to develop image inpainting. Image Inpainting uses the information from the rest of the image which is not corrupted to produce a patch which looks like it is a part of the original image.

Image Inpainting fills the regions that are missing or damaged using the spatial information available in its surrounding region. Image Inpainting is an important research subclass of image processing. It plays an important role in computer graphics, in preserving the historical heritage and eliminating the unwanted objects. Image Inpainting refers to manipulation of an image in an undetectable manner.

By the help of Image Inpainting we can fix the void or patch in the original image making the observer feel that there was never a patch in the new image produced by Inpainting. Image Inpainting techniques simply fills the patch with the information which is obtained from the remaining parts of the original image. These methods does not re-construct the original image, but instead fills the void or the gap in the image. Marcelo Bertalmio was the first person to introduce the world to the notion of Image Inpainting. The proposed Image Inpainting algorithms follow the following steps

- Restoration of the image is the main purpose of the Image Inpainting and the way to fill the hole is determined by the information that is received from its surrounding pixels.
- The hole is packed by continuing the structure of the region surrounding the gap into the hole. Prolongation of the contour lines arriving at the boundary of the hole is done.
- The contour lines define different regions in inpainting domain and are filled with the colors the match the boundary of the hole.
- Finally, the texture is added by filling the small regions.

Image inpainting algorithms find the widespread applications which include:

Object removal: The objects indicated by the user can be removed by using inpainting techniques in a visually plausible way.

Scratch removal: Old images damaged by scratches can be recovered by applying image inpainting algorithms at the part containing the scratch.

Correction of corrupted images during transmission: Image gets often corrupted in wireless transmission. By considering the lost part as inpainting domain the original image can be restored.

Generating visually stunning effects: In an image we can produce the required stunning effects by using inpainting technique.

Text removal: Inpainting algorithms can be used to eliminate the unwanted texts on the image.

II. Literature Review

There are many algorithms proposed earlier and some of them yielded good results while some did not.Each of those implementations have their own advantages along with some limitations.

Diffusion Method: An algorithm was developed to repair the damaged region by using the information from the pixels surrounding the damaged region. This algorithm was proposed by Bertalmio [1]. The filling of the damaged region was done in such a way that isophote lines arriving at the region's boundry are completed towards inside. This algorithm produced good results for structured images. The result of this algorithm is an inpainted image without colour artifacts and is sharp. But this algorithm was not contrast variant and another drawback of this algorithm was that it could not reproduce

images with big textured regions. This was caused by the use of Total Variation denoising model. So as an improvement, the Variational Image denoising method was used. This new method preferred isotopes of edge curves with shortest length and straight lines. So this method also had a drawback of producing images with bad continuity.

Then the inpainting problem was considered as a particular case of image interpolation by A.C.Kokaram [9]. Expressing the level lines in terms of local neighborhoods by using a Taylor expansion a third-order PDE that performs inpainting is derived. This PDE is optimal and the most accurate third-order PDE which can ensure continuation of level lines. The drawback of all the above mentioned methods is that when these are applied to images of large patches of corrupted region produce an image which is blur or over smooth. So these methods are good only for small missing regions.

Texture Synthesis Method: In this method the damaged region was filled using the patches. These patches were synthesised by texture synthesis. This method derives a parametric model by analysing the input and synthesising the output texture. This method proposed by Yamauchi [2]. The above method produced brilliant results for stochastic textures but it had huge computational time and also failed to reconstruct the global features of the structural texture. In order to maintain the global features of texture, patch-based texture synthesis algorithms have been proposed. But this approach, however, is still not suitable for the general image because of the patch boundary problem. In general, the texture based algorithms fail to recover the edges.

Improvements were made to the basic approaches by separating the image into cartoon and texture layers, and sparsely represented these two layers by two incoherent over-complete transforms. This approach can effectively fill in the region with composite textures and structures, especially in the application of missing block completion. However, similar to the diffusion-based approach, it may fail to recover structure or introduce smooth effect when filling large missing region.

Our proposed algorithm contains deep learning.Generative Adversarial Networks are used to reconstruct the lost part of the image. Since deep learning involves in creating,evaluating and recreating,better results can be obtained which will overcome above limitations.By selecting appropriate loss function the results will be even better.

A. Database Collection

For our DCGAN based image inpainting algorithm, we used CelebA dataset which comprises of some thousands of images of human faces collected from the web. The CelebA dataset is mainly used for training, which contains in total of 202,599 human face images,.and similar images are used for validation and testing.

B. Preprocessing

Before we feed our raw images of celebA dataset into our model for training, preprocessing is to be done to it using openface library to

III. Proposed Methodology

For our DCGAN approach, the general idea is to generate an image that looks close to the target image and then take the corresponding area to ll in the target image. Fig.1. depicts the basic flowchart of such implementation.

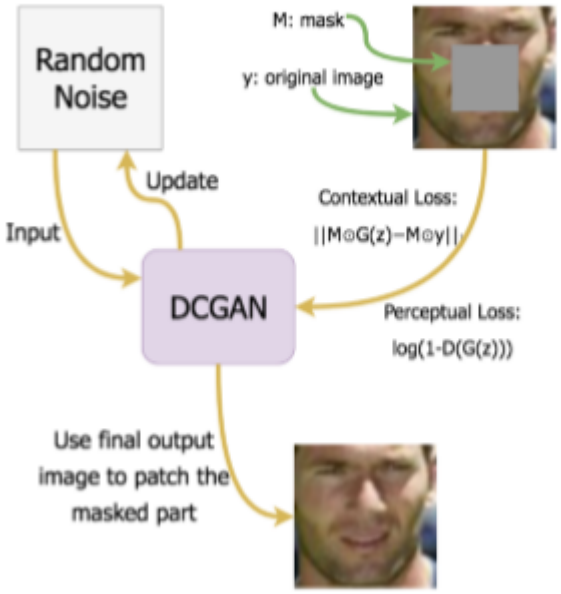

Fig. 1. Control-Flow of our method

First, using a dataset of images as input, we train a DCGAN model following the algorithm proposed in Goodfellow et al.[5]. We can select any desired database to work on.

1. crop the images to 64 x 64 pixels size
2. remove other Noises except the face section

From then we advance through further preprocessing. In this step, the pixel values of input images are divided with value 127.5 (average between 0 and 255), and then subtract by one. By doing so, we are converting our pixel values from range of [0 , 255] to [-1 , 1]. This results in normalization of data., i.e the stability of DCGAN increases.

Some sample images we used from the above two datasets and for our algorithm are presented in Fig. 2

Fig. 2. sample images of celebA dataset

C. DCGAN

In this section we discuss about the methodology of our implementation.This includes the implementation of DCGAN(Deep Convolutional Generative Adversarial Networks).This consists of 2 networks namely Generator and Discriminator which work in adversarial(opposite) manner.

Discriminator is a network involving a set of convolutional layers with a fully connected network on the top as shown below in table, which makes it a classifier neural network. whereas the generator is exact opposite.A generator produces an image by taking a 1 dimenasional vector as an input.It generally consists of a linear layer with a reshape to convert a 1 dimensional noise vector into desired dimensions which are then converted to a 3 X 64 X 64 image using 4 other CONV2D layers with appropriate activation layers(relu is preferred).

Layer (type)	Output Shape
input(noise)	100
reshape	4 X 4 X 1024
conv1	8 X 8 X 512
conv2	16 X 16 X 256
conv3	32 X 32 X 128
conv4	64 X 64 X 3

generator summary

Layer (type)	Output Shape
input(image)	64 X 64 X 3
conv 5 X 5	32 X 32 X 64
conv 5 X 5	16 X 16 X 128
conv 5 X 5	8 X 8 X 256
conv 5 X 5	4 X 4 X 512
full(dense)	1

discriminator summary

The compile function for this model uses Adam optimizer with standard parameters to ensure reliable training updates.The discriminator then after being trained over the dataset its **trainable** flag is disabled.so that the parameters are set to differentiate between real and fake images. This whole process works through gradient decent method.so the images generated by the generator are discriminated using discriminator network.Then the generator tries to generate more realistic image with respect to given dataset by using the feedback from the discriminator.This feedback is given in the form of loss which is weighted sum of contextual and perceptual loss of the image generated.

Fig 3. gives the pictorial representation of the working of above adverasial network.

Contextual loss captures the intuition that if the generated image G(z) have similar values at places where the original image is not missing, then the part of G(z) where we use to complete the original image should t reasonably well.

A mask M is defined which is a matrix of size same as the image contains 0's and 1's. The value of 0 represents the part of the image we need to restore.so by hadamard multiplication of the mask M over our image y results masked image $M \otimes y$

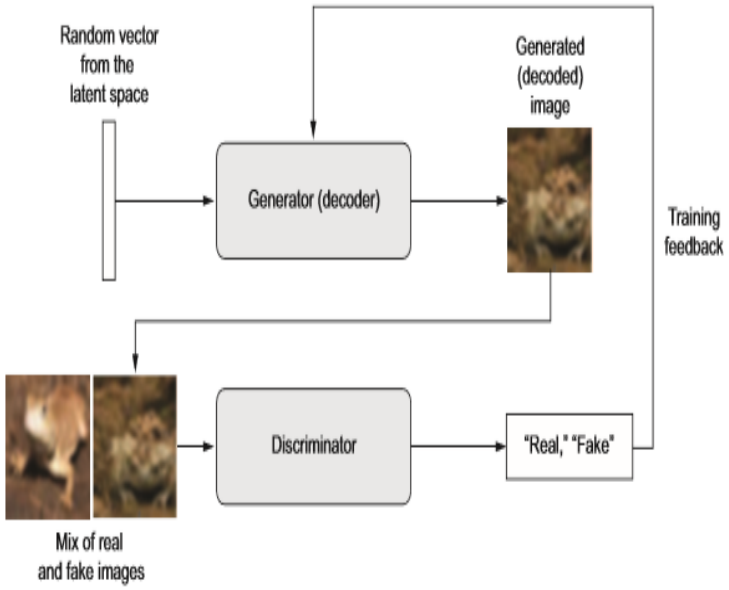

Fig. 3. Generator - Discriminator

We then derive the contextual loss as the L1 distance between the masked original image $M \otimes y$ and the masked generated image $[M \otimes G(z)]$. This ensures that the image we generate is as similar as possible to the target image in unmasked parts. Whereas the perceptual loss is used to measure the realism of the image generated.Naturally, we can pass a generated image G(z) through discriminator D and evaluate the loss as below. After many iterations of gradient descent, we obtain a realistic image that looks like our target image.

Lcontextual(z) = $||M \otimes G(z) - M \otimes y||$

Lperceptual(z) = $\log(1 - D(G(z)))$

xreconstructed = $M \otimes y + (1 - M) \otimes G(z)$

So to improve the quality of the image to be generated, we have to use the suitable loss function that helps the generator network to be trained in such a manner that it can generate images very close to reality.

IV. Result and Observations

In this section,we discuss about our evaluations on DCGAN implementation.

After training our DCGAN with 20 epochs. Many types of loss functions have been used and observed the results.we observed that **sigmoid_cross_entropy_loss_function** gave better results when compared to other loss functions like **mean_squared_error** etc.The results were obtained by comparing the similarity of original and inpainted image.

This is because Cross-entropy is preferred for classification, while mean_squared_error is one of the best choices for regression. This comes directly from the statement of the problems itself - in classification you work with very particular set of possible output values thus MSE is badly defined (as it does not have this kind of knowledge thus penalizes errors in incompatible way).The result(inpainted image) is as follows when we used **sigmoid cross entropy loss function**:

Fig. 4. original and inpainted img

The metrics we used for calculating the similarity between images mentioned below. These are available in python library named scikit-image.

A. *Mean Square Error(MES)*

MSE (Mean Square Error) is calculated by

$$MSE = \frac{1}{NM} \sum_{n=1}^{N-1} \sum_{m=0}^{M-1} (R(n,m) - P(n,m))^2 \quad (1)$$

where N x M is the size of the image, R(n,m) is the real image and P(n,m) is the predicted image.

Table below depicts the two measures computed on the image inpainting results obtained by using different inbuilt loss functions.

Loss function used	MSE
Mean_Square_Error	441.68
Sigmoid_cross_entropy	105.76

Quantitative Evaluation

V. Conclusion and Futurework

We tackle the challenging image inpainting task by providing a novel GAN-based solution that maintains the original spatial information of the training distribution and generates plausible content encompassing both the local continuity and global consistency. By using an effective loss function better results can be obtained. In future, we plan to make the model capable of handling different size, shapes, and positions of masking. Also we would try to implement User defined loss function inorder to make it work more efficiently.

VI. REFERENCES

[1] Bertalmo, Marcelo Sapiro, Guillermo Caselles, Vicent Ballester, C. (2000). Image inpainting. Proceedings of the ACM SIGGRAPH Conference on Computer Graphics. 417-424.

[2]H. Yamauchi, J. Haber and H. -. Seidel, Image restoration using multiresolution texture synthesis and image inpainting, Proceedings Computer Graphics International 2003, Tokyo, Japan, 2003.

[3] Lixin Yin and Chen Chang, An effective exemplar-based image inpainting method, 2012 IEEE 14th International Conference on Communication Technology, Chengdu, 2012.

[4] Bin Shen, Wei Hu, Y. Zhang and Y. Zhang, Image inpainting via sparse representation, 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, 2009.

[5] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adverasial Networks, arXiv preprint arXiv:1406.2661, 2014.

[6] Xiao M, Li G, Xie L, Peng L, Chen Q (2018) Exemplar-based image completion using image depth information. PLoS ONE 13(9): e0200404

[7] C. Ramya, and S.Subaha Rani. A Novel in painting Algorithm based on Sparse Representation. International Journal of Computer Applications (0975 8887) Volume 74 No.6, July 2013

[8] X. Y. Liu, C.-H. Lai, K. A. Pericleous, and M. Q. Wang.On a Modified Diffusion Model for Noise Removal.Journal of Algorithms Computational Technology Vol. 6 No. 1