TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

HIMALAYA COLLEGE OF ENGINEERING

[CT-707]

A

FINAL YEAR MAJOR PROJECT PROGRESS REPORT

ON

IMAGE INPAINTING USING GAN

BY:

Ajay Maharjan (HCE075BCT006)

Ashish Rai (HCE075BCT042)

Nibendra Bajracharya (HCE075BCT015)

Sujan Maharjan (HCE075BCT037)

A PROJECT SUBMITTED TO DEPARTMENT OF
ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL
FULFILLMENT OF THE REQUIREMENT FOR BACHELOR'S
DEGREE OF COMPUTER ENGINEERING

HIMALAYA COLLEGE OF ENGINEERING

LALITPUR,NEPAL

December, 2022

IMAGE INPAINTING USING GAN

BY:

Ajay Maharjan (HCE075BCT006)

Ashish Rai (HCE075BCT042)

Nibendra Bajracharya (HCE075BCT015)

Sujan Maharjan (HCE075BCT037)

A report submitted for partial fulfillment of the requirements for the degree of Bachelor in Computer Engineering.

Department of Electronics and Computer Engineering

HIMALAYA COLLEGE OF ENGINEERING

Tribhuvan University

Lalitpur, Nepal

December, 2022

ACKNOWLEDGEMENT

We take this occasion to thank our parents for their consistent support and encouragement. We are immensely thankful to our college, **Himalaya College of Engineering**, for including the project in the syllabus. We are also very grateful for the college for providing us with this opportunity. Furthermore, we extend our sincere and heartfelt thanks to our **Head of Department, Er. Ashok GM,** for providing us with the right guidance and for showing us the right way. We would like to express our deep gratitude towards **Er. Narayan Adhikari Chhetri** as well as other faculty members for their proper guidance and inspiration. Finally, we would like to give special thanks to acknowledge all the people who have helped us towards the development of this proposal.

Group Members

Ajay Maharjan (HCE075BCT006)

Ashish Rai (HCE075BCT042)

Nibendra Bajracharya (HCE075BCT015)

Sujan Maharjan (HCE075BCT037)

ABSTRACT

In the era, where colors and style fascinate everyone, more emphasis is given on aesthetics and beauty. This research paper proposes a deep learning method based on Generative Adversarial Network (GAN) as well as Convolutional Neural Network (CNN) to develop an application for converting images into artistic style, colorization of the image, and inpainting of image. Image inpainting aims to fill missing regions of a damaged image with plausibly synthesized content. Existing methods for image inpainting either fill the missing regions by borrowing information from surrounding areas or generating semantically coherent content from region context. They often produce ambiguous or semantically incoherent content when the missing region is large or with complex structures. The proposed method combines the two applications into a single web-based application. Here, colorization is performed with a GAN which employs a U-Net as its generator model. Similarly, image inpainting is carried out using Deep Convolution Generative Adversarial Network (DCGAN).

Keywords*: CNN, Deep Learning, Image Colorization, Image Inpainting, GAN.*

TABLE OF CONTENTS

Contents

LIST OF FIGURES

LIST OF ABBREVIATIONS

DB	Database
JS	JavaScript
GAN	Generative Adversarial Network
CNN	Convolutional Neural Network
DCGAN	Deep Convolution Generative Adversarial Network
DFD	Data Flow Diagram
ReLU	Rectified Linear Unit
SDLC	Software Development Lifecycle

CHAPTER 1. INTRODUCTION

1.1. Introduction

Currently, image processing using a deep learning technique is an emerging area and is gaining greater popularity especially in improving the quality of digital images. Transferring Style from one image to the other is one of the major concerns in texture transfer. In texture transfer, the main motive is to combine a texture feature from an input image to the expected image. This is done by preserving the semantics of the target image. They synthesize the real like natural textures by resampling the pixels of a designated source texture. Inpainting is a procedure in which is used to recover the lost fragments of an image and to recreate them. Image inpainting is applied for restoring old images, damaged films, and to edit an image in order to eliminate undesired image content. Currently, deep learning and neural networks have obtained a lot of recognition among researchers in the area of image processing. CNNs, and GANs have proved to be a successful method in image recognition, color recognition, image sharpening and restoration, pattern recognition, and image generation.

1.2. Problem Statement

As time passes by, some part of the image may suffer from corrosion and it is difficult for humans to restore the real part of images. Thus, we propose a system that can restore the damaged part of an image in such a way that the inpainted region cannot be detected by a casual observer and looks realistic.

1.3. Objectives

- To reconstruct damaged parts or missing parts of image using GANs.

1.4. Scope and Application

Image inpainting can be immensely useful for museums that might not have the budget to hire a skilled artist to restore deteriorated paintings. Image inpainting can also be extended to videos (videos are a series of image frames after all). Due to over-compression, it is very likely that certain parts of the video can get corrupted sometimes. Modern image inpainting techniques are capable of handling this gracefully as well. Producing images where the missing parts have been filled with both visually and semantically plausible appeal is the main objective of an artificial image inpainter.

CHAPTER 2. LITERATURE REVIEW

Colorization basically involves assigning realistic colors to grey-scale image. Convolutional neural networks are specifically designed to deal with image data. Many authors have done promising work on this idea. Domonkos Varga [1] proposed the idea of automatic coloring of cartoon images, since they are very different from natural images, they pose a difficulty as their colors depend on artist to artist. So, the data-set was specifically trained for cartoon images, about 100000 images, 70% of which were used in training and rest for validation. But unfortunately, the color uncertainty in cartoons is much higher than in natural images and evaluation is subjective and slow. Shweta Salve [2] proposed another similar approach, employing the use of Google's image classifier, Inception ResNet V2. The system model is divided into 4 parts, Encoder, Feature extractor, Fusion layer and Decoder. The system is able to produce acceptable outputs, given enough resources, CPU, Memory, and large data-set. This is mainly proof of concept implementation. V.K. Putri [3] proposed a method to convert plain sketches into colorful images. It uses sketch inversion model and color prediction in CIELab color space. This approach is able to handle hand-drawn sketches including various geometric transformations. The limitation found was that, data-set is very limited but it works well for uncontrolled conditions. Richard Zhang [4] has proposed a optimized solution by using huge data-set and single feed-forward pass in CNN. Their main focus lies on training part. They used human subjects to test the results and were able to fool 32% of them. can have various number of neurons.

Inpainting is the process of completing or recovering the missing region in the image or removing some objects added to it. To handle this, many methods have been proposed including sequential algorithms or deep learning techniques. For that, we categorize the existing methods for images inpainting into three categories: sequential-based approaches, CNN-based approaches, and GAN-based Approaches. Recently, the strong potential of deep convolutional networks (CNNs) is being exhibited in all computer vision tasks, especially in image inpainting. CNNs are used specifically in order to improve the expected results in this field using large-scale training data. The sequential-based methods succeed in some parts of image inpainting like filling texture details with promising results, yet the problem of capturing the global structure is still a challenging task. Several methods have been proposed for image inpainting using convolutional neural networks (CNNs) or encoder-decoder network based on CNN. Shift-Net based on U-Net architecture is one of these methods that recover the missing block with good accuracy in terms of structure and fine-detailed texture.

The much-used technique nowadays, was introduced for image generation in 2014 in [5]. Generative adversarial networks (GANs) are a framework which contains two feed-forward networks, a generator G and a discriminator D. The generative network, G, is trained to create a new image which is indistinguishable from real images, whereas a discriminative network, D is trained to differentiate between real and generated images. This relation can be considered as a two-player min-max game in which G and D compete. To this end, the G (D) tries to minimize (maximize) the loss function, i.e. adversarial loss, as follows:

$$\min_{G} \max_{D} E_{x\sim P_{data}(x)}\left[\log D(x)\right] + E_{z\sim P_z(z)}\left[\log(1 - D(G(z)))\right]$$

where z and x denote a random noise vector and a real image sampled from the noise Pz(z) and real data distribution Pdata(x), respectively. Recently, the GAN has been applied to several semantic inpainting techniques in order to complete the whole region naturally.

GANs are a framework that contains two feed-forward networks, a generator G and a discriminator D. The generator takes random noise z as input and generates some fake samples similar to real ones; while the discriminator has to learn to determine whether samples are real or fake. At present, Generative Adversarial Network (GAN) becomes the most used technique in all computer vision applications. GAN-based approaches use a coarse-to-fine network and contextual attention module gives good performance and is proven to be helpful for inpainting. Existing image inpainting methods based on

GAN are generally a few. Out of these, we find that in [6], Chen and Hu proposed a GAN-based semantic image inpainting method, named progressive inpainting, where a pyramid strategy from a low-resolution image to a higher one is performed for repairing the image.

For handwritten images, Li et al. [7] proposed a method for inpainting and recognition of occluded characters. The methods use improved GoogLeNet and deep convolutional generative adversarial network (DCGAN). In an image inpainting method named PEPSI [8] the authors unify the two-stage cascade network of the coarse-to-fine network into a single-stage encoder-decoder network. Where PEPSI++ is the extended version of PEPSI [9]. In [10] the authors used Encoder-decoder network and multi-scale GAN for image inpainting. The same combination is used in [11] for image inpainting and image-to-image transformation purposes. On the RBG-D images, Dhamo et al. [12] used CNN and GAN model to generate the background of a scene by removing the object in the foreground image as performed by many methods of motion detection using background subtraction [13] [14]. In order to complete the missing regions in the image, Vitoria et al. [15] proposed an improved version of the Wasserstein GAN with the incorporation of Discriminator and Generator architecture. In the same context, but on sea surface temperature (SST) images, the Dong et al. [16] proposed a deep convolutional generative adversarial network (DCGAN) for filing the missing parts of the images. Also, Lou et al. [17] exploit a modifier GAN architecture for image inpainting whereas, Salem et al. [18] proposed a semantic image inpainting method using adversarial loss and self-learning encoder-decoder model. A good image restoration method requires preserving structural consistency and texture clarity. For this reason, Liu et al. [19] proposed a GAN-based method for image inpainting on face images. FiNet [20] is another approach found in the literature for fashion image inpainting that consists of completing the missing parts in fashion images.

Recently, several approaches are proposed by combining some additional techniques (GAN, CNN...) for inpainting the images. Jiao et al. [21] combined an encoder-decoder, multi-layer convolutions layers and GAN for restoring the images. The authors in [22] proposed a two-stage adversarial model named EdgeConnect by providing a generator for edge followed by an image inpainting model. The first model attempt to provide an edge completion component and the second one, inpaint the RGB image. According to the fact that GAN-based image inpainting models do not care out to the consistency of the structural and textural values between the inpainted region and their neighboring, the authors in [23] attempts to handle this limitation by providing a GAN model for learning the alignment between the block around the restored region and the original region. For the same reason as [23], taking into consideration the semantic consistency between restored images and original images, Li et al. [24] provided a boosted GAN model comprising an inpainting network and a discriminative network. When the inpainting network discovers the segmentation information of the input images, the discriminative network discovers the regularizations of the overall realness and segmentation consistency with the original images. In the same context and using GAN-based models for images inpainting, each work provides some prior processing on GAN networks to get the best inpainting results for different types of images including medical images [25], face images [26] or scenes images [27].

The GAN-based methods give a good addition to the performance of image inpainting algorithms, but the speed of training is lower and needs very good performance machines, and this is due to computational resources requirements including network parameters and convolution operations.

CHAPTER 3. REQUIREMENT ANALYSIS

3.1. Functional Requirements

The functionalities that the system should provide in order to satisfy the needs and requirements of the users are as listed below:

Use Case Diagram

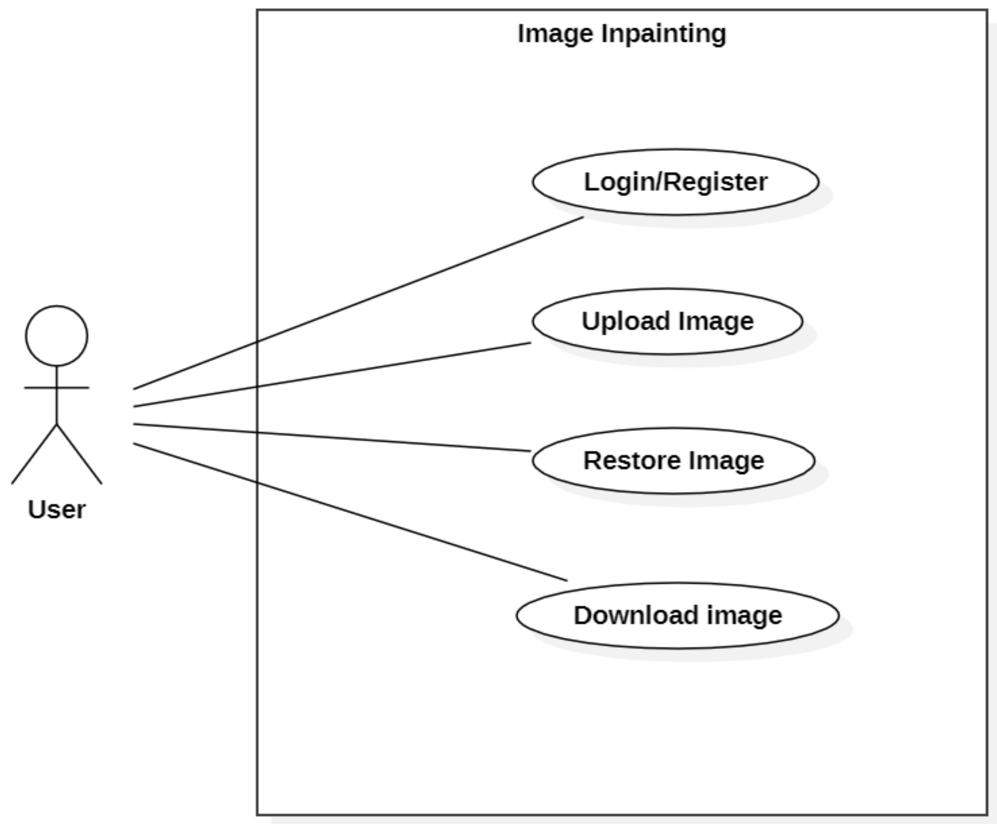

Figure 1: Use case diagram

3.1.1. Login/Register

The user will be able to register and login to the system, which will provide him/her with additional features such as unlimited processing of images.

3.1.2. Add image

The users can upload their images in the website and use the tools provided to modify the uploaded image.

3.1.3. Image restoration

The image provided, which has damaged or missing parts, can be restored into another image which approximately identical to the original image (ground truth).

3.1.4. Download image

After processing of the image, users can download the newly generated image.

3.2. Non-Functional Requirements

3.2.1. Reliability

The system has to be reliable by properly handling unwanted actions or exceptions.

3.2.2. Availability

The system should have uptime to the maximum level.

3.2.3. Performance

The User Interface should be interactive by responding to the actions fast.

3.2.4. Scalability

The system should be capable of supporting the growth and address the concurrent actions.

3.2.5. Maintainability

The system should be maintainable after the deployment.

3.2.6. Security

The system should store the users' credentials securely.

3.2.7. Usability

The user interface should be simple and easily adaptable for the users to operate the system with ease.

3.3. Feasibility Analysis

3.3.1. Technical Feasibility

The application uses software technologies and tools which are freely available. The technical skills required can be easily manageable. There are many research papers for analysis. The hardware technology required for operation is easy to obtain since the application can run on any computer with a web browser and an internet connection. So, the hardware and software technicalities are within accessible boundaries.

3.3.2. Operational Feasibility

Since, the application is interactive, the user can easily be familiarized with the software system. This system highly focuses on parameters like reliability, maintainability, supportability, usability, sustainability, etc. that fits into the operating functions of the project. As the system is accessible with a web browser, it can be easily operated the desired functionalities, both by the user and the administrator.

3.3.3. Economic Feasibility

Economic feasibility attempts to weigh the costs of developing and implementing a new system, against the benefits that would increase from having the new system in place. This feasibility study gives the top management the economic justification for the new system. There could be various types of intangible benefits on account of automation. The objectives may be achieved with a little investment and some periodic maintenance of the system which will prove beneficial to the organization in the long run.

CHAPTER 4. SYSTEM DESIGN

4.1. Software Development Approach

The project will implement the **Incremental Software Model** in its SDLC. It will be developed in multiple increments. In each successive increment, certain portion of the system will be developed. After completion of each increments, testing will be performed to ensure quality of the system.

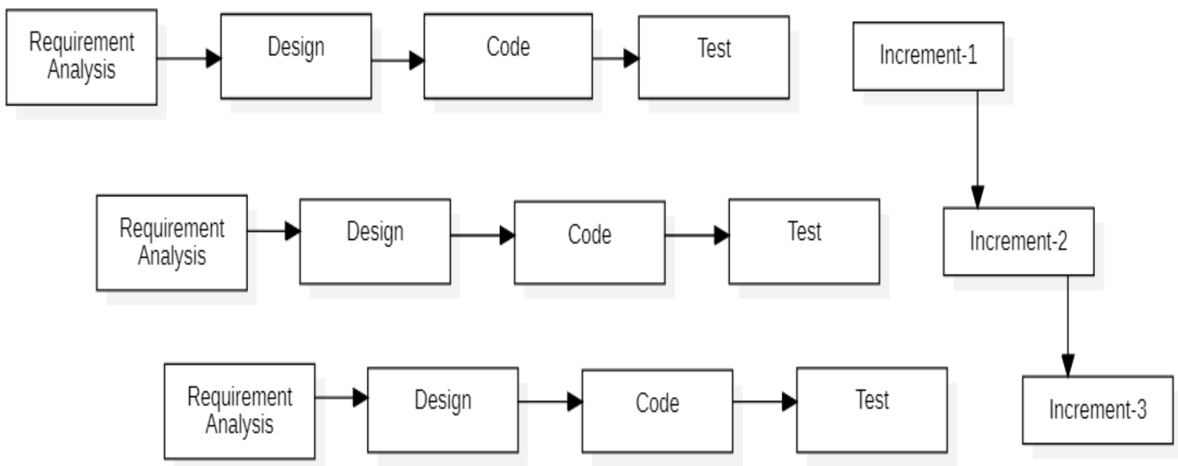

Figure 2: Representation of incremental model

4.2. DFD Diagram

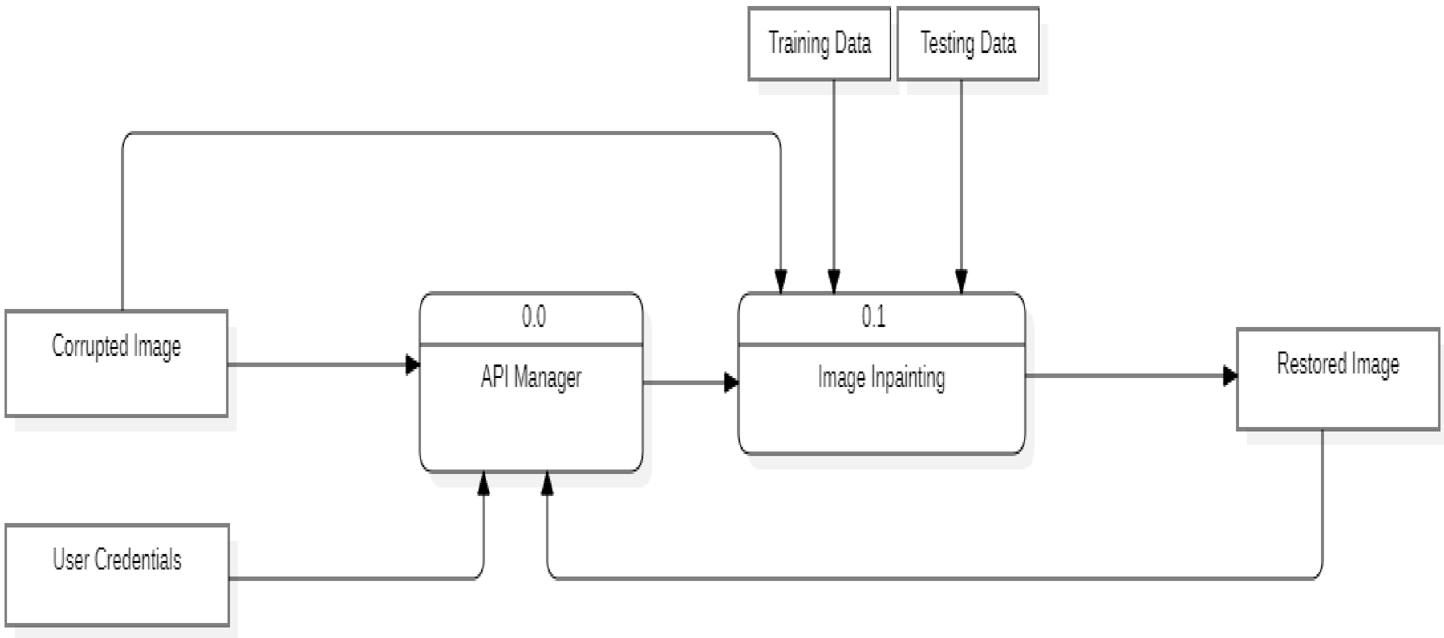

Figure 3: DFD Level 0

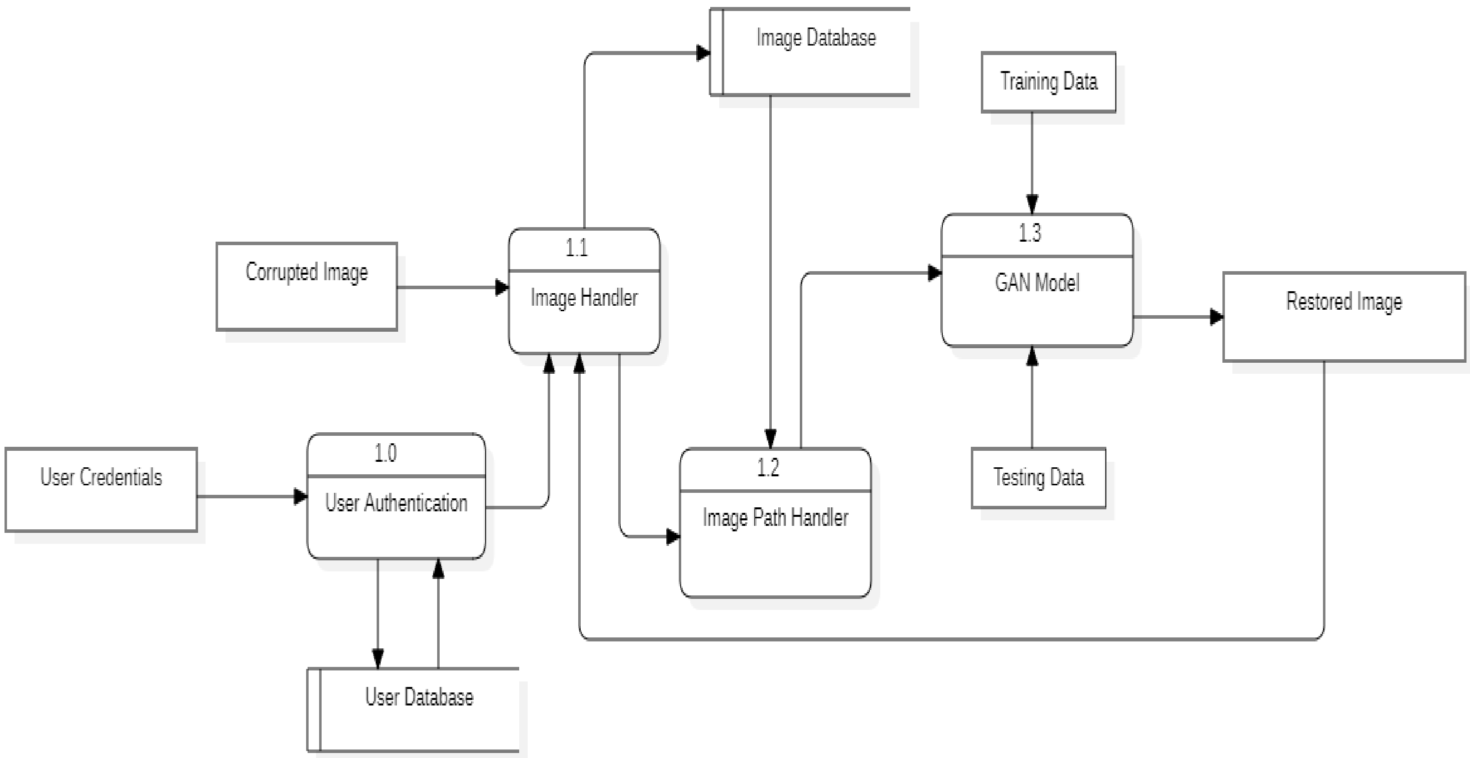

Figure 4: DFD Level 1

4.3. ER Diagram

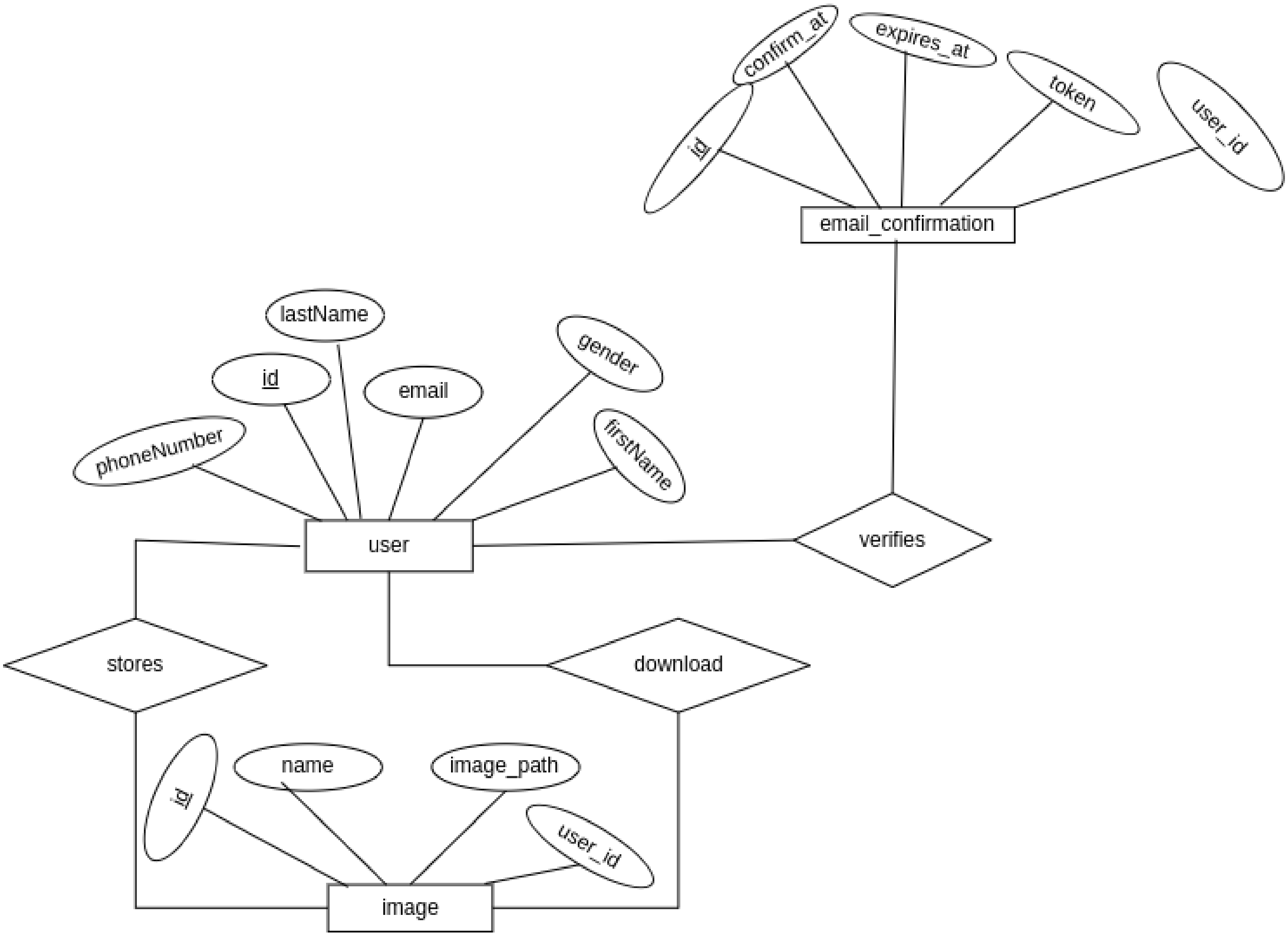

Figure 5: ER Diagram

CHAPTER 5. METHODOLOGY

5.1. Implementation

5.1.1. Image inpainting

5.1.1.1. Conditional GANs

Generative Adversarial Nets (GAN) contains two competing neural network models, including a generator and a discriminator, which are two players in a game and trying to beat each other. Generally speaking, the discriminator tries to tell the fake images generated by the generator from real images, while the generator tries to generate good-looking images to fool the discriminator. Specifically, for our model, the generator takes in the input cropped images and generates fake recovered images. The discriminator takes in both generated images from the generator and the ground truth real images, along with the cropped images, and tries to discriminates real images from fake generated images. During the training process, the generator and the discriminator are playing a continuous game. At each iteration, the generator is trained to produce more realistic images, while the discriminator is getting better at distinguishing fake images. Both models are trained together in a minimax fashion and the goal is to train a generator to be indistinguishable from real data.

5.1.1.2. Network Architecture

The architecture of the generator network we built is based on the U-Net, which incorporates convolution layers in an encoder-decoder fashion to generate recovered images from cropped images. The generator consists of an encoder, which is a contracting network, and a decoder, which is an expanding network. The input to the encoder is the input tensor of cropped images. The encoder shrinks the size of the tensor layer by layer, with several convolutional layers whose strides are larger than one. However, the depth of the tensor, e.g. the last dimension of the tensor is increased layer by layer with an increasing number of filters used each layer. The output of the decoder is a small tensor ($4 \times 4 \times 512$) which is the encoded embeddings in the latent space, containing the context information of the original images. The output of the encoder is the input to the decoder, which expands the tensor layer by layer with conv2d transpose and construct the recover images that are of the original image size. Using this model can lead to much more compact feature learning in the middle of the layers without consuming large memory.

The encoder and the decoder are basically symmetrical: there are 6 layers of encoding and 6 layers of decoding. The number of filters in the encoder increases layer by layer, while the number of filters in the decoder decreases number of filters layer by layer. Each encoding layer consists of a 2D-convolution for down-sampling, batch normalization, and leakly relu activations; each decoding layer consists of a transpose convolution for up-sampling, batch normalization and relu activations. To allow the network to skip layers, we concatenate the mirroring layer from encoder at each decoding layer. With skipped layers, the model can learn weights to ignore deeper layers. This can help model to retain components from original input easier in deep CNN, which is particularly useful in segmentation task.

The discriminator is a simple decoder classifier network, and its architecture is shown in figure3. The input of the discriminator is the concatenation of the cropped image with either the ground truth images or the recovered images generated by the generator. The discriminator is consisted of 5 layers of encoder, which is similar to the encoder of the generator: each encoding layer is consisted of a convolution operation with stride greater than 1, a batch normalization, and a leaky relu activation. The last layer then goes through a sigmoid activation to return a number from 0 to 1 that can be interpreted as the probability of the input being real or fake.

5.1.1.3. Loss Functions and Objective

With conditional GAN, both generator and discriminator are conditioning on the input x. Let the generator be parameterized by θg and discriminator be parameterized by θd.

The objective function consists two parts. The first part represents the loss coming from the discriminator. If the discriminator did well in distinguishing the generated picture and target, this gives the generative model a high loss, vice versa. while second part represents the L1 loss the difference between the generated picture and the target. Note that we did not introduce noise in our generator because we do not find it working better. With GAN, if the discriminator considers the pair of images generated by the generator to be fake (not well recovered), the loss will be back-propagated through discriminator and through generator. Therefore, the generator can learn how to recover the image to make it look real.

5.2. Project Tools

The following tools are going to be utilized for the development of this system.

- ReactJS
- NodeJS
- Python

CHAPTER 6. EPILOGUE

6.1. Tasks Completed

S.N.	***Tasks completed***	***Description***
1.	Static client-side application	Static user interface has been developed which allows the users to interact with the system.
2.	Research on GAN models	Reviewed some research papers on GAN models used for image inpainting.
3.	Server-side backend development	Building database, creating APIs, authentication and authorization have been completed.
4.	Building the GAN model	A GAN model has been built as per the official research paper.
5.	Creating the dataset	The dataset has been built from Google streetview images and processed as required by the model.
6.	Training the model	The model has been trained with the generated dataset.

6.2. Tasks Remaining

S.N.	***Tasks remaining***	***Description***
1.	Dynamic application	The dynamic properties of a web application such as responsiveness, event handling, routing, etc. need to be added.
2.	Evaluating the trained model	The outputs generated and accuracy of model need to be reviewed. As per the observations, the model may be modified and retrained.
3.	Integration	The python module needs to be integrated with server-side application.
4.	System testing	After complete integration, the system as a whole, needs to be tested to ensure that it meets the functional and non-functional requirements.

REFERENCES

[1] C. A. S. D. V. a. T. Szirfffdfffdnyi, "Automatic cartoon colorization based on convolutional neuralnetwork," 2017. [Online]. Available: https://core.ac.uk/download/pdf/94310076.pdf. [Accessed 13 06 2022].

[2] S. Salve, T. Shah, V. Ranjane, and S. Sadhukhan, "Automatization of coloring grayscale images using convolutional neural network," 2018.

[3] V. K. Putri and M. I. Fanany, "Sketch plus colorization deep convolutional neural networks for photos generation from sketches," pp. 1-6, 2017.

[4] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," *ECCV,* 2016.

[5] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y, "Generative adversarial nets," *Advances in neural information processing systems,* p. 2672–2680, 2014.

[6] H. H. Chen Y, "An improved method for semantic image inpainting with gans: progressive inpainting.," *Neural Process Lett,* p. 1355–1367, 2019.

[7] Li J, Song G, Zhang M, "Occluded offline handwritten Chinese character recognition using deepconvolutional generative adversarial network and improved GoogLeNet.," *Neural Comput Appl,* p. 1–15, 2008.

[8] Sagong M-c, Shin Y-g, Kim S-w, Park S, Ko S-j, "Pepsi: fast image inpainting with paralleldecoding network," *Proceedings of the IEEE conference on computer vision and pattern recognition,* p. 11360–11368, 2019.

[9] Shin YG, Sagong MC, Yeo YJ, Kim SW, Ko SJ, "Pepsi++: fast and lightweight network for imageinpainting," 2019.

[10] Wang H, Jiao L, Wu H, Bie R, "New inpainting algorithm based on simplified context encodersand multi-scale adversarial network," *Procedia Comput Sci,* p. 254–263, 2019.

[11] Wang C, Xu C, Wang C, Tao D, "Perceptual adversarial networks for image-to-image transformation," *IEEE Trans Image Process,* p. 4066–4079, 2018.

[12] Dhamo H, Tateno K, Laina I, Navab N, Tombari F , "Peeking behind objects: layered depth pre-diction from a single image," *Pattern Recognit Lett*, p. 333–340, 2019.

[13] Elharrouss O, Al-Maadeed N, Al-Maadeed S, "Video summarization based on motion detection forsurveillance systems," in *15th international wireless communications & mobile computing conference(IWCMC)*, 2019.

[14] Almaadeed N, Elharrouss O, Al-Maadeed S, Bouridane A, Beghdadi A, "A novel approach forrobust multi human action detection and recognition based on 3-dimentional convolutional neural networks," 2019.

[15] Vitoria P, Sintes J, Ballester C, "Semantic image inpainting through improved Wasserstein generative adversarial networks," 2018.

[16] Dong J, Yin R, Sun X, Li Q, Yang Y, Qin X, "Inpainting of remote sensing sst images with deepconvolutional generative adversarial network," *Geosci Remote Sens Lett*, p. 173–177.

[17] Lou S, Fan Q, Chen F, Wang C, Li J, "Preliminary investigation on single remote sensing imageinpainting through a modified gan," in *2018 10th IAPR workshop on pattern recognition in remotesensing (PRRS)*, 2018.

[18] Salem N.M, Mahdi HM, Abbas H, "Semantic image inpainting vsing self-learning encoder-decoderand adversarial loss," in *2018 13th international conference on computer engineering and systems(ICCES)*, 2018.

[19] Liu H, Lu G, Bi X, Yan J, Wang W, "Image inpainting based on generative adversarial networks," in *2018 14th international conference on natural computation, fuzzy systems and knowledge discovery(ICNC-FSKD)*, 2018.

[20] Han X, Wu Z, Huang W, Scott MR, Davis LS, "Compatible and diverse fashion image inpainting," 2019.

[21] Jiao L, Wu H, Wang H, Bie R, "Multi-scale semantic image inpainting with residual learning andGAN," *Neurocomputing,* p. 199–212.

[22] Nazeri K, Ng E, Joseph T, Qureshi F, Ebrahimi M, "Edgeconnect: generative image inpainting with adversarial edge learning," 2019.

[23] Li A, Qi J, Zhang R, Ma X, Ramamohanarao K , "Generative image inpainting with sub-manifold alignment," 2019.

[24] Li A, Qi J, Zhang R, Kotagiri R, "Boosted gan with semantically interpretable information forimage inpainting," in *2019 international joint conference on neural networks (IJCNN)*, 2019.

[25] Armanious K, Mecky Y, Gatidis S, Yang B, "Adversarial inpainting of medical image modalities," in *ICASSP 2019–2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2019.

[26] Yeh R A, Chen C, Yian Lim T, Schwing AG, Hasegawa-Johnson M, Do MN, "Semantic imageinpainting with deep generative models," in *Proceedings of the IEEE conference on computer vision andpattern recognition*, 2017.

[27] Yuan L, Ruan C, Hu H, Chen D, "Image inpainting based on patch-GANs," 2019.