***Mini Project Report***

***on***

**Enhancement of Blurred Text images**



## By

**Aryan Lohia (Reg. No.-202100437)**

**Gourav Raidongia (Reg. No.-202100528)**

**Nivedita Singh (Reg. No.-202100447)**

**Vivien Elizabeth Lawrence (Reg. No.-202100537)**

**Group Id – C10**

*In partial fulfillment of requirements for the award of degree in*

*Bachelor of Technology in Computer Science and Engineering*

*(2024)*

Under the Project Guidance of

**Mr. Ashis Pradhan**

**Assistant Professor (Selection Grade)**

**Sikkim Manipal Institute of Technology, Majitar**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SIKKIM MANIPAL INSTITUTE OF TECHNOLOGY**

(A constituent college of Sikkim Manipal University)

MAJITAR, RANGPO, EAST SIKKIM – 737136

**PROJECT COMPLETION CERTIFICATE**

This is to certify that the below mentioned students of Sikkim Manipal Institute of Technology have worked under my supervision and guidance from **8th January 2024 to 29th** **April 2024** and successfully completed the Mini project entitled **“Enhancement of Blurred Text Images”** in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering.

|  |  |  |
| --- | --- | --- |
| University Registration No | Name of Student | Course |
| **202100437** | **Aryan Lohia** | **B.Tech (CSE)** |
| **202100528** | **Gourav Raidongia** | **B.Tech (CSE)** |
| **202100447** | **Nivedita Singh** | **B.Tech (CSE)** |
| **202100537** | **Vivien Elizabeth Lawrence** | **B.Tech (CSE)** |

**Mr. Ashis Pradhan**

Assistant Professor (Selection Grade)

Department of Computer Science and Engineering

Sikkim Manipal institute of Technology

Majhitar, Sikkim – 737136

**PROJECT REVIEW CERTIFICATE**

This is to certify that the work recorded in this project report entitled **“Enhancement of Blurred Text Images”** has been jointly carried out by **Aryan Lohia (Reg. 202100437), Gourav Raidongia (Reg. 202100528), Nivedita Singh (Reg. 202100447), and Vivien Elizabeth Lawrence (Reg. 202100537)** of Computer Science & Engineering Department of Sikkim Manipal Institute of Technology in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering. This report has been duly reviewed by the undersigned and recommended for final submission for Mini Project Viva Examination.

**Mr. Ashis Pradhan**

Assistant Professor (Selection Grade)

Department of Computer Science and Engineering

Sikkim Manipal Institute of Technology

Majhitar, Sikkim – 737136

**CERTIFICATE OF ACCEPTANCE**

This is to certify that the below mentioned students of Computer Science & Engineering Department of Sikkim Manipal Institute of Technology (SMIT) have worked under the supervision of **Mr. Ashis Pradhan**, Assistant Professor (Selection Grade), Department of Computer Science and Engineering from **8th January 2024 to 29th April 2024** on the project entitled **“Enhancement of Blurred Text Images”.**

The project is hereby accepted by the Department of Computer Science & Engineering, SMIT in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering.

|  |  |  |
| --- | --- | --- |
| University Registration No | Name of Student | Project Venue |
| **202100437**  **202100528**  **202100447**  **202100537** | **Aryan Lohia**  **Gourav Raidongia**  **Nivedita Singh**  **Vivien Elizabeth Lawrence** | **SMIT** |

**Dr. Udit Kumar Chakraborty**

Professor & Head of the Department

Computer Science & Engineering Department

Sikkim Manipal Institute of Technology

Majhitar, Sikkim – 737136

**DECLARATION**

We, the undersigned, hereby declare that the work recorded in this project report entitled “**Enhancement of Blurred Text Images**” in partial fulfillment for the requirements of award of B.Tech (CSE) from Sikkim Manipal Institute of Technology (A constituent college of Sikkim Manipal University) is a faithful and bonafide project work carried out at “**SIKKIM MANIPAL INSTITUTE OF TECHNOLOGY**” under the supervision and guidance of **Mr. Ashis Pradhan,** Assistant Professor (Selection Grade), Department of Computer Science and Engineering.

The results of this investigation reported in this project have so far not been reported for any other Degree or any other Technical forum.

The assistance and help received during the course of the investigation have been duly acknowledged.

**Aryan Lohia (Reg. No.-202100437)**

**Gourav Raidongia (Reg. No.-202100528)**

**Nivedita Singh (Reg. No.-202100447)**

**Vivien Elizabeth Lawrence (Reg. No.-202100537)**

**ACKNOWLEDGMENT**

We take this opportunity to acknowledge indebtedness and a deep sense of gratitude to our guide **Mr. Ashis Pradhan** for his valuable guidance and supervision throughout the course which shaped the present work as it shows.

We pay our deep sense of gratitude to **Prof. (Dr.) Udit Kumar Chakraborty, HOD, Computer Science & Engineering Department, Sikkim Manipal Institute of Technology** for giving us the opportunity to work on this project and providing all support required.

We are obliged to our Mini Project coordinators **Dr. Sandeep Gurung, Mr. Biraj Upadhyaya** and **Ms. Tanuja Subba** for elevating, inspiration and supervising in completion of our project.

We would also like to thank any other staff of **Computer Science & Engineering Department, Sikkim Manipal Institute of Technology** for giving us continuous support and guidance that has helped us in completion of our project.

**Aryan Lohia (Reg. No.-202100437)**

**Gourav Raidongia (Reg. No.-202100528)**

**Nivedita Singh (Reg. No.-202100447)**

**Vivien Elizabeth Lawrence (Reg. No.-202100537)**

**DOCUMENT CONTROL SHEET**

|  |  |  |
| --- | --- | --- |
| 1 | Report No | CSE/Mini Project/Internal/B.Tech/Section/C10/2024 |
| 2 | Title of the Report | Enhancement of Blurred Text Images |
| 3 | Type of Report | Technical |
| 4 | Author | Aryan Lohia, Gourav Raidongia, Nivedita Singh, Vivien Elizabeth Lawrence |
| 5 | Organizing Unit | Sikkim Manipal Institute of Technology |
| 6 | Language of the Document | English |
| 7 | Abstract | This project focuses on WGAN to deblur textual images, thereby enhancing clarity and readability. |
| 8 | Security Classification | General |
| 9 | Distribution Statement | General |

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Chapter** |  | **Title** | **Page No.** |
|  |  | Abstract |  |
| 1 |  | Introduction | **1** |
| 2 |  | Literature Survey | 3 |
| 3 |  | Problem Definition | 6 |
| 4 |  | Solution Strategy | 8 |
| 5 |  | Design | 9 |
|  | 5.1 | Block Diagram of WGAN | 9 |
|  | 5.2 | Flowchart | 10 |
| 6 |  | Implementation details | 11 |
|  | 6.1 | Dataset used | 11 |
|  | 6.2 | Primary Technique: Use of WGANs for Deblurring textual images. (Algorithm) | 11 |
|  | 6.3 | Pseudocodes | 12 |
|  | 6.4 | Screenshots | 13 |
|  | 6.5 | Code Snippets of the five other techniques explored for deblurring textual images | 15 |
|  | 6.6 | Output of Secondary Techniques explored for deblurring text images | 18 |
| 7 |  | Results and discussion | 19 |
| 8 |  | Conclusion and Future Scope of the project | 25 |
| 9 |  | Limitations | 26 |
| 10 |  | Gantt Chart | 27 |
| 11 |  | References | 28 |
| 12 |  | Plagiarism Report | 29 |

--- **LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Name** | **Page No.** |
| 1.1 | Blurred text image and Deblurred text image | 1 |
| 1.2 | Deblurring of license plates caught in traffic security cameras. | 2 |
| 1.3 | Deblurring old educational document. | 2 |
| 5.1 | Block diagram of WGAN | 9 |
| 6.1 | Original blurred textual image | 13 |
| 6.2 | The code snippet of the deblur function along with the deblurred output. | 14 |
| 6.3 | Deblurred pictures using Phase corrected Van Cittert Snippet, Modified Levenberg–Marquardt iterations , Modified Wiener, Approximate Landweber and Modified Richardson-Lucy. | 18 |
| 7.1 | Comparison between PSNR and SSIM on the various techniques used | 22 |
| 7.2 | Comparison between PSNR and SSIM on the various techniques used | 23 |
| 7.3 | Discriminator Training Losses(Perceptual Loss + Wasserstein Distance) | 24 |
| 7.4 | Mean losses upto 10 Epochs | 24 |
| 10.1 | Gantt Chart |  |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Title Name** | **Page no.** |
| 2.1 | Literature survey | 3-5 |
| 7.1 | Generator model architecture | 21-22 |
| 7.2 | Discriminator model architecture | 23 |
| 7.3 | Model summary: Generator and Discriminator | 24 |

**ABSTRACT**

This project, “Enhancement of Blurred Text Images” addresses the challenge of deblurring textual images. The model utilizes Wasserstein Generative Adversarial Network (WGAN) to deblur images containing textual data for interpretation. Textual images frequently have blurred content which makes the text present in it uninterpretable and reduces the visual quality of the image. Deblurring process involves enhancing the hidden details in images caused by motion blur. Deblurring textual images makes the characters easily readable. The unique features of WGANs help in developing a powerful solution for improving the sharpness and clarity of textual images.

Due to unique properties of textual images, traditional methods often struggle with blurred textual images. Our approach utilizes WGANs, which comprises of a generator network to reduce blurring in images and a discriminator network for verifying image authenticity. By using adversarial training, the model is able to generate realistic deblurred text images without compromising with textual integrity. To preserve intricate details, Wasserstein GAN is incorporated with gradient penalty instead of Jensen-Shannon Divergence to guide the network towards producing solutions that improves the readability of text characters. This enhances the stability and convergence of the training process, contributing to overall effectiveness of the proposed approach.

1. **INTRODUCTION**

This project addresses the challenges of deblurring for image enhancement where textual data is involved and provides a solution for scenarios where visual quality is severely hindered.

Blurring in images can occur due to various reasons such as motion blur (movement during capture), out-of-focus blur, or intentional blurring for privacy or aesthetic reasons. Reading a blurry text and identifying characters may cause eye strain, if it is even readable. Text deblurring specifically deals with restoring clarity to blurred text in images making it easily readable which involves preserving the intricate details of the image. Deblurring involves restoring hidden details in images caused by motion blur. An example of an image before and after deblurring is given below:

1. (b)

Fig 1.1: blurred text image and Deblurred text image

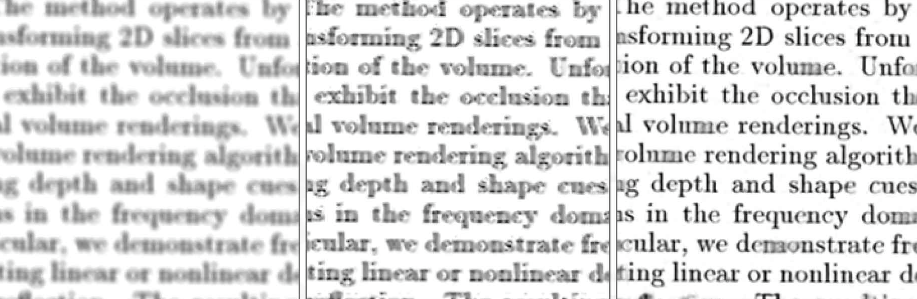
Deblurring text images is images for ensuring accurate interpretation and extraction of textual information from images in various real-life scenarios. Whether in documents, photographs, or scanned materials, clear text improves readability and understanding of the data, making tasks such as Optical Character Recognition (OCR), document analysis, and archival easier.

Deblurring techniques also play a significant role in improving the readability of license plates captured in surveillance footage or images from traffic monitoring cameras for vehicle identification and tracking.



Fig 1.2:Deblurring of license plates caught in traffic security cameras.

Another application can be enhancing the readability of text in educational materials, textbooks, or digital documents thereby, benefiting students and researchers.

  
 Fig 1.3:Deblurring old educational document.

The real-world applications are vast. From surveillance systems that solely depends on sharp imagery to outdoor photography that requires high-quality pictures, to remote sensing applications that need precise analysis of environmental conditions, deblurring plays a crucial role. Generative Adversarial Networks (GANs) have have emerged as a powerful framework for generating realistic and high-quality synthetic data. In the area of image generation, one notable extension of GANs is the Wasserstein Generative Adversarial Network (WGAN), which was introduced to address some of the training instability issues associated with traditional GANs. WGANs redefine the optimization problem by introducing the Wasserstein distance that offers better convergence properties and improved stability during training.

**2. LITERATURE SURVEY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl no.** | **Author** | **Paper and publication details** | **Findings** | **Relevance** |
| 1 | Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, Jiri Matas | DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks (3 Apr 2018) | * DeblurGAN achieves state-of-the-art performance in structural similarity measure and visual appearance. * DeblurGAN has higher recall and F1 score than its competitors in object detection. * DeblurGAN is 5 times faster than the closest competitor, Deep-Deblur. * DeblurGAN introduces a novel method for generating synthetic motion blurred images. | * Adversarial and perceptual loss strategies can enhance visual quality. * Consider transferable insights on network architectures. * Emphasis on perceptual quality improvement is relevant. |
|  | Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron Courville | Improved Training of Wasserstein GANs (25 Dec 2017) | * Proposed method performs better than standard WGAN. * Enables stable training of a wide variety of GAN architectures. | * The article suggests techniques for stable GAN training, preventing mode collapse, and generating higher quality samples. * These insights are relevant for deblurring images using Wasserstein GANs, helping achieve stable training dynamics, diverse output, and improved deblurring performance. |
| 3 | Yu Ti | Gradient Penalty Approach for Wasserstein  Generative Adversarial Networks | * The paper focuses on Wasserstein GANs and the gradient penalty approach. * WGAN-GP method uses gradient penalty to improve training stability. * Wasserstein distance is utilized to ensure realistic and high-quality outputs * Practical implications include improved training stability and high-quality image generation. | * The techniques and concepts discussed, such as the gradient penalty approach, are relevant to the task of deblurring textual images. * The paper provides insights into the implementation details and considerations for using the gradient penalty approach. |
| 4 | Snehal Bhatia & Rozenn Dahyot | Using WGAN for Improving Imbalanced Classification Performance (2019) | * GAN approach compared favorably with standard data augmentation for image classification. * Data synthesis with Generative Adversarial Network (GAN) for class imbalance * No specific limitations were mentioned in the provided contexts. | * the concept of using WGAN (Wasserstein Generative Adversarial Network) for image processing tasks, as explored in the paper is applied to the project. * This approach could potentially improve the quality and legibility of textual images that have been blurred due to various factors. |

**3. PROBLEM DEFINITION**

When an image containing text is blurred, the readability and interpretability of the text are highly compromised. When blurriness in the image leads to inability to interpret text, it can lead to misinterpretations, errors, and inefficiencies in extracting data. The blurriness in image occurs due to various reasons:

* *Motion Blur*: Caused by camera or subject movement during image capture, resulting in smearing or streaking of objects.
* *Out-of-Focus Blur*: Occurs when the camera fails to properly focus on the subject, leading to a lack of sharpness and clarity.
* *Atmospheric Disturbances*: Environmental factors such as fog, haze, or humidity can distort and blur images, particularly in outdoor settings.
* *Sensor Noise*: Imperfections in the camera sensor or electronic noise during image acquisition can contribute to image blurring.
* *Compression Artifacts*: Lossy compression techniques used in digital images can introduce blurring and loss of detail, especially in complex or textured regions.

The main objective of this project is to develop an effective solution for restoring clarity and detail of textual images.

**Traditional Generative Adversarial Networks (GANs)** face several challenges, including:

*Mode Collapse***:** This is one of the most commonly seen and significant challenges with traditional GANs. It occurs when the generator produces limited varieties of outputs, failing to capture the full diversity of the data distribution.

*Training Instability***:** Traditional GAN training can be unstable, characterized by oscillations or divergence in the loss functions of the generator and discriminator. This instability can lead to difficulties in convergence which makes the training sensitive to hyperparameters.

*Discriminator Saturation***:** In the early stages of training, the discriminator may become too effective at distinguishing real and fake samples which leads to vanishing gradients and hindering further learning. This phenomenon can make it difficult for the generator to improve its performance.

*Adversarial loss***:** Traditional Generative Adversarial Networks (GANs) commonly use the minimax objective function, also known as the adversarial loss, which has drawbacks such as non-convergence, mode collapse, vanishing gradients, discriminator saturation, and difficulty in evaluation. These issues can make training unstable, hinder convergence, and lead to poor sample quality.

**4. SOLUTION STRATEGY**

The proposed solution, Wasserstein Generative Adversarial Networks (WGANs) offers an effective solution for text deblurring and enhancing text images.

The WGAN consists of a **generator** and a **critic network**. The Blurred Image is fed into the Generator which tries to generate an image closest to the sharp image during training.

The Generated Image is then compared to the Sharp Image from the Dataset and **Wasserstein distance** is calculated which is used to optimize the **Generator Network**.

WGANs address common challenges encountered in traditional GANs, such as **mode collapse** and **training instability**, resulting in more stable training dynamics. In WGANs, the Wasserstein loss is typically used as the primary objective function to optimize the Wasserstein distance between distributions, promoting stable training and better convergence properties. The perceptual loss is used as an additional loss component to further enhance the visual quality of generated samples by incorporating high-level semantic information.

The **Wasserstein loss**, which quantifies the difference between the expected values of the critic's scores for real and generated samples. The Wasserstein loss is used as the primary objective function in WGANs. It measures the dissimilarity between the distributions of real and generated samples. Unlike traditional GANs, which use the Jensen-Shannon divergence, WGANs optimize the Wasserstein distance, leading to more stable training dynamics.

**WGAN gives better clarity, readability, and stability during training.** Moreover, WGAN is useful for text images as it detects the intricate features as compared to Auto Encoders which might not detect the same.

**5. DESIGN**

**5.1 Block diagram:**

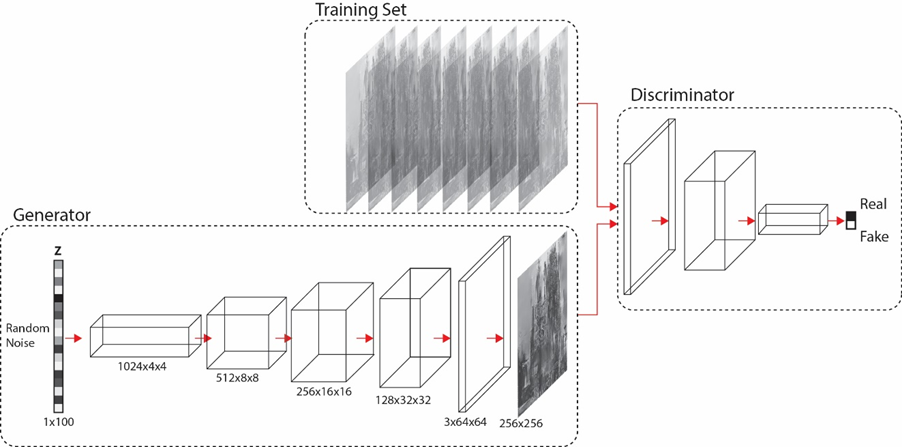
****

Fig5.1 : Block diagram of WGAN

**Working of WGAN:**

1. Input Noise: The input noise is a random vector sampled from a predefined distribution (e.g., Gaussian distribution). In text deblurring, this noise vector is the blurred image which serves as the input to the generator network.

2. Generator: The generator takes the input noise vector and transforms it into a deblurred text image. It consists of multiple layers of neural networks, such as convolutional layers followed by up sampling layers, which progressively transform the noise into an output image. The generator tries to learn the mapping from the input noise space to the space of realistic text images.

3. Blurred Text Image: This represents the input blurred text image that needs to be deblurred. It serves as the real data during training.

4. Discriminator: The discriminator network evaluates the realism of the deblurred text images produced by the generator. It takes both real blurred text images and deblurred text images as input and distinguishes between them. The discriminator is trained to differentiate between real and generated images.

5. Wasserstein Distance Loss: This loss function measures the discrepancy between the distributions of real blurred text images and generated deblurred text images. It encourages the generator to produce deblurred images that are similar to real images in terms of text content.

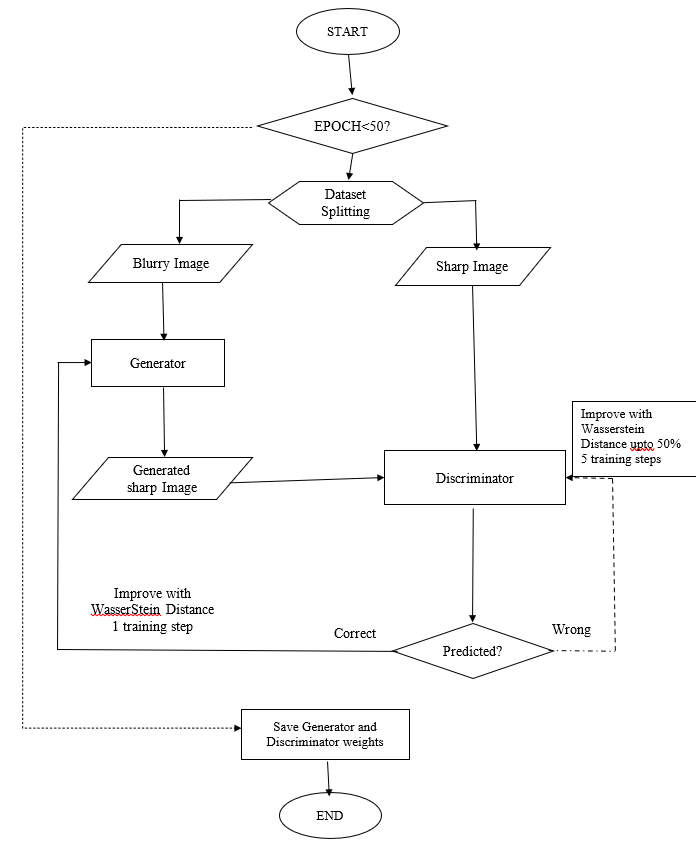
6. Gradient Penalty: In WGAN, a gradient penalty term is often used to enforce the Lipschitz constraint on the discriminator. This penalty helps to stabilize the training process by penalizing gradients that violate the constraint.

7. Generator Loss: The generator loss is a combination of the Wasserstein distance loss and the gradient penalty. It represents the loss incurred by the generator network during training.

8. Discriminator Loss: The discriminator loss measures how well the discriminator can distinguish between real and generated images. It consists of the Wasserstein distance loss minus the gradient penalty.

During training, the generator and discriminator networks are trained alternately. The generator aims to produce deblurred text images that are realistic and indistinguishable from real images, while the discriminator aims to correctly classify between real and generated images. This iterative process continues until both networks converge to a stable state, producing high-quality deblurred text images.

**5.2 Flowchart :**



**6. IMPLEMENTATION:**

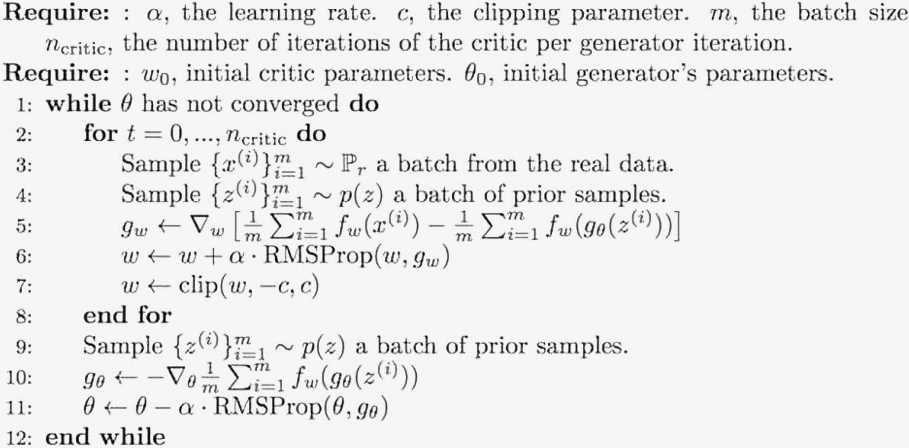
**6.1 Dataset used:** **Text Image with Motion Blur**

**About Dataset:**

* This dataset contains images that are and aren't affected by motion blur.
* 184 images of various book titles and texts both motion blurred and normal.
* Images are of the shape 256x256x3 in JPG Format

**6.2 Primary Technique: Use of WGANs for Deblurring textual images.**

**Algorithm:**



**6.4 Screenshots:**

****

Fig6.1: Original blurred textual image

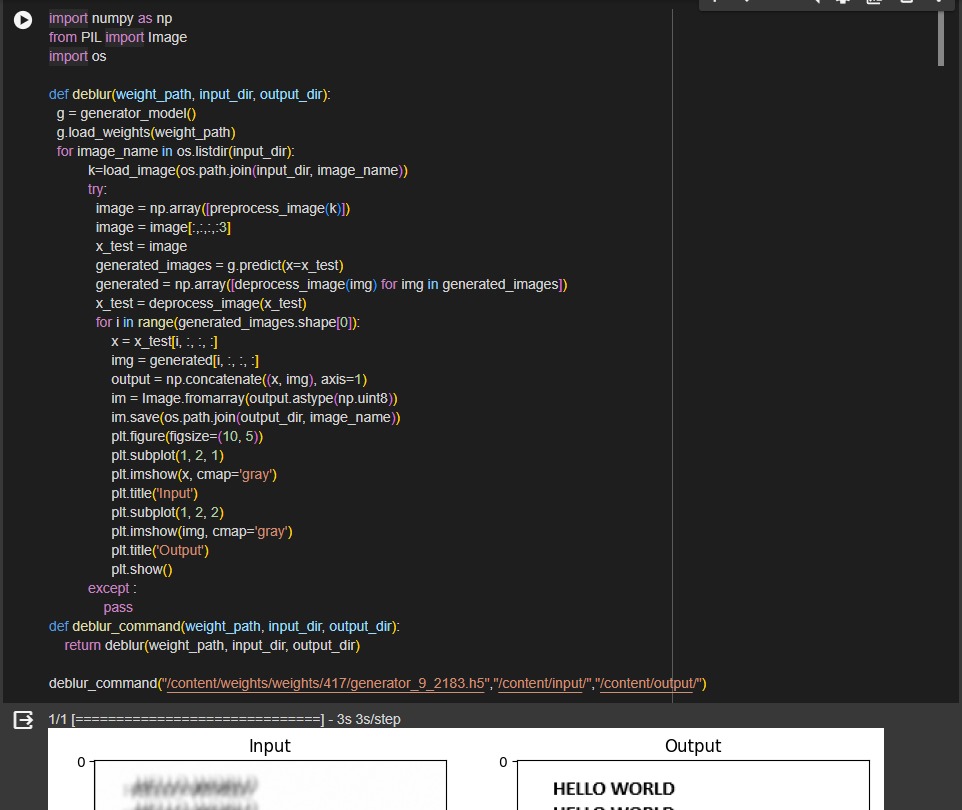




Fig6.2:The code snippet of the deblur function along with the output.

**6.5 Code Snippets of the five other techniques explored for deblurring textual images:**

1)Phase corrected Van Cittert Snippet:

def pcVC(F, y, maxiter=100):

TM = y.copy()

for i in range(maxiter):

H = scipy.fft.fftn(F(TM)) / (scipy.fft.fftn(TM)+2.2204e-16)

TM = TM + np.real(scipy.fft.ifftn((scipy.fft.fftn(y)/(H+2.2204e-16)-scipy.fft.fftn(TM)) \* np.absolute(H)))

return TM

def pcVC\_nsr(F, y, maxiter=100):

# Use the same formulation as in the paper

nsr = utils.estimate\_nsr(y)

a = 100.0\*nsr

TM = y.copy()

for i in range(maxiter):

H = scipy.fft.fftn(F(TM)) / (scipy.fft.fftn(TM)+2.2204e-16)

Hconj = np.conjugate(H)

TM = TM + np.real(scipy.fft.ifftn(Hconj/(np.absolute(H)+a) \* (scipy.fft.fftn(y) - H\*scipy.fft.fftn(TM))))

return TM

2)Modified Levenberg–Marquardt snippet:

def mLM(F, y, maxiter=100):

nsr = utils.estimate\_nsr(y)

a = 100.0\*nsr

lm = y.copy()

for i in range(maxiter):

Flm = F(lm)

H = scipy.fft.fftn(Flm) / (scipy.fft.fftn(lm) + 1e-16)

Hconj = np.conjugate(H)

num = Hconj \* (scipy.fft.fftn(y - Flm))

denom = (Hconj \* H + a)

lm = lm + np.real(scipy.fft.ifftn(num/denom))

return lm

3) Modified Richardson-Lucy snippet:

def mRL(F, y, maxiter=500):

RL = y.copy()

for i in range(maxiter):

r1 = F(RL)/(np.abs(y)+1e-16)

r1 = utils.reflect(r1)

r2 = F(r1)

r2 = utils.reflect(r2)

RL = RL / (np.abs(r2)+1e-16)

r1 = y / (np.abs(F(RL))+1e-16)

r1 = utils.reflect(r1)

r2 = F(r1)

r2 = utils.reflect(r2)

RL = RL \* r2

return RL

4) Modified Wiener snippet:

def mW(F, y, maxiter=100):

nsr = utils.estimate\_nsr(y)

#a = 100.0\*nsr

a = nsr

W = y.copy()

FW = F(W)

H = scipy.fft.fftn(FW) / (scipy.fft.fftn(W) + 1e-16)

for i in range(1, maxiter+1):

H = H\*(i-1)/i + scipy.fft.fftn(FW) / (scipy.fft.fftn(W) + 1e-16)/i

Hconj = np.conjugate(H)

W = np.real(scipy.fft.ifftn(Hconj/(Hconj\*H + a)\*scipy.fft.fftn(y)))

FW = F(W)

return W

5) Approximate Landweber snippet:

def aL(F, y, maxiter=500):

L = y.copy()

for i in range(maxiter):

hp = y - F(L)

hp = utils.reflect(hp)

d = (F(L+hp) - F(L-hp))/2.0

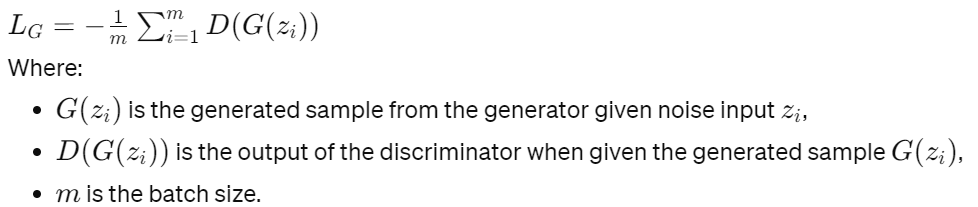
d = utils.reflect(d)

L = L + d

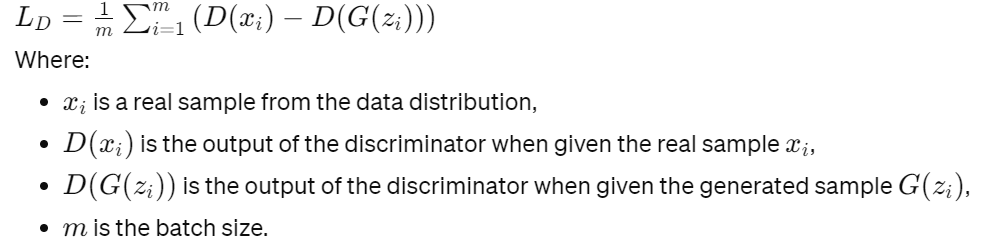
return L

* 1. **Mathematical Formula:**

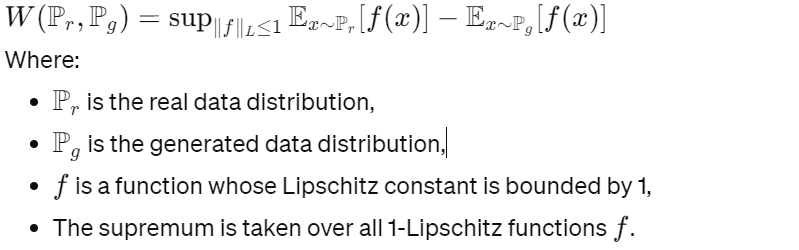
**Wasserstein GAN:**

**1.** **Generator Loss (L­G):**

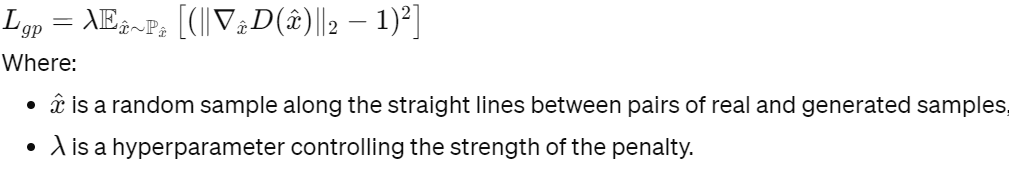
**2. Discriminator Loss (L\_D):**

****

**3. Wasserstein Distance:**

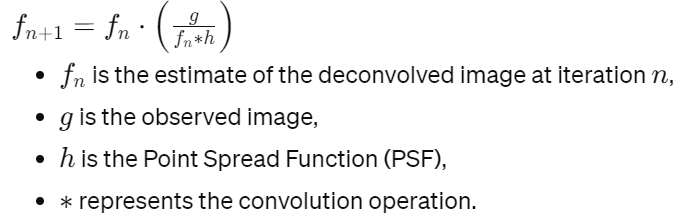
****

**4.** **Gradient Penalty:**

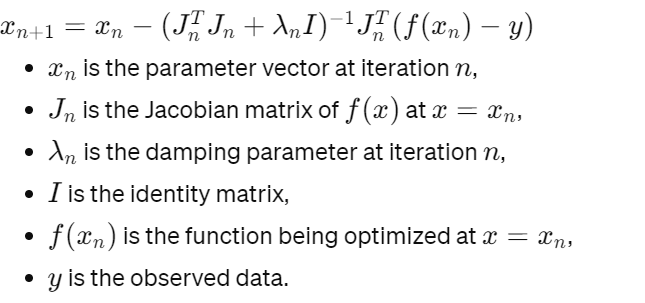
****

**Other techniques:**

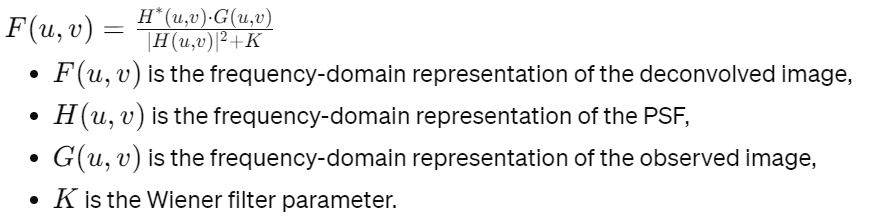
* + - 1. **Phase Corrected Van Cittert Snippet:**

****

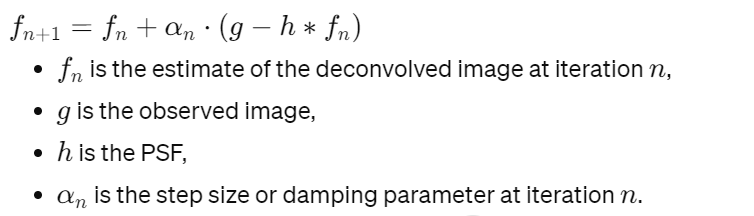
* + - 1. **Modified Levenberg–Marquardt Iterations:**

****

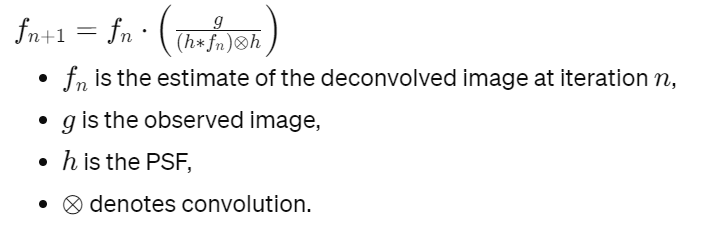
* + - 1. **Modified Wiener:**

****

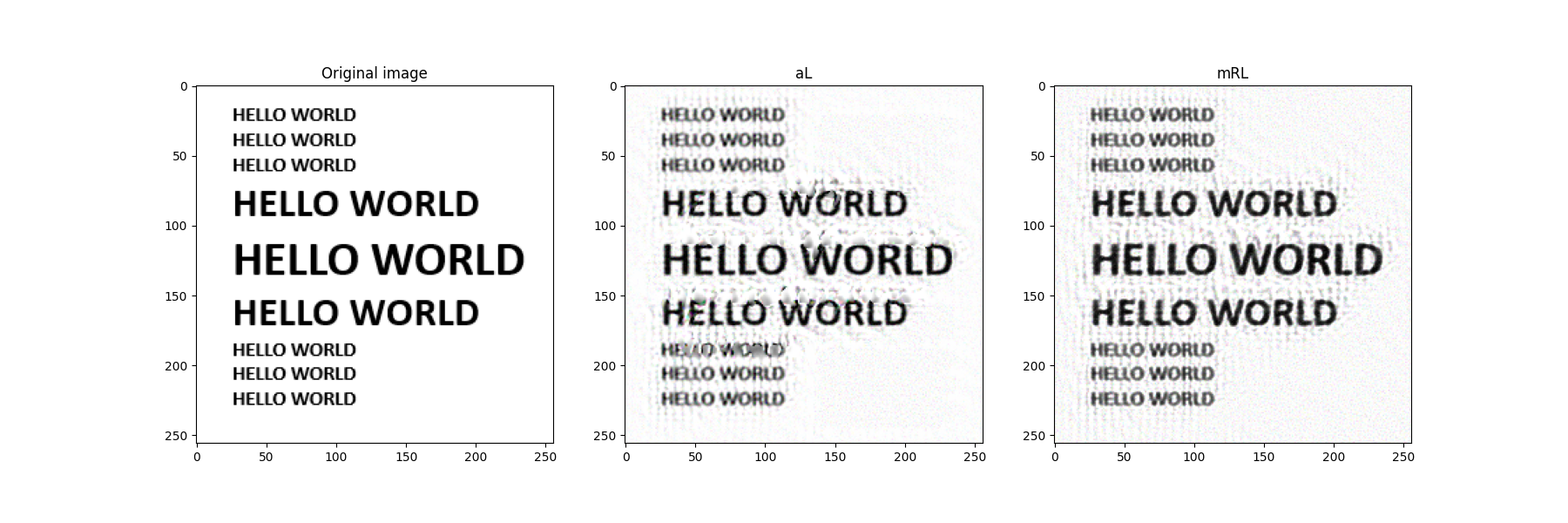
* + - 1. **Approximate Landweber:**

****

* + - 1. **Modified Richardson-Lucy**:

****

**6.7 Output of Secondary Techniques explored for deblurring text images:**

****

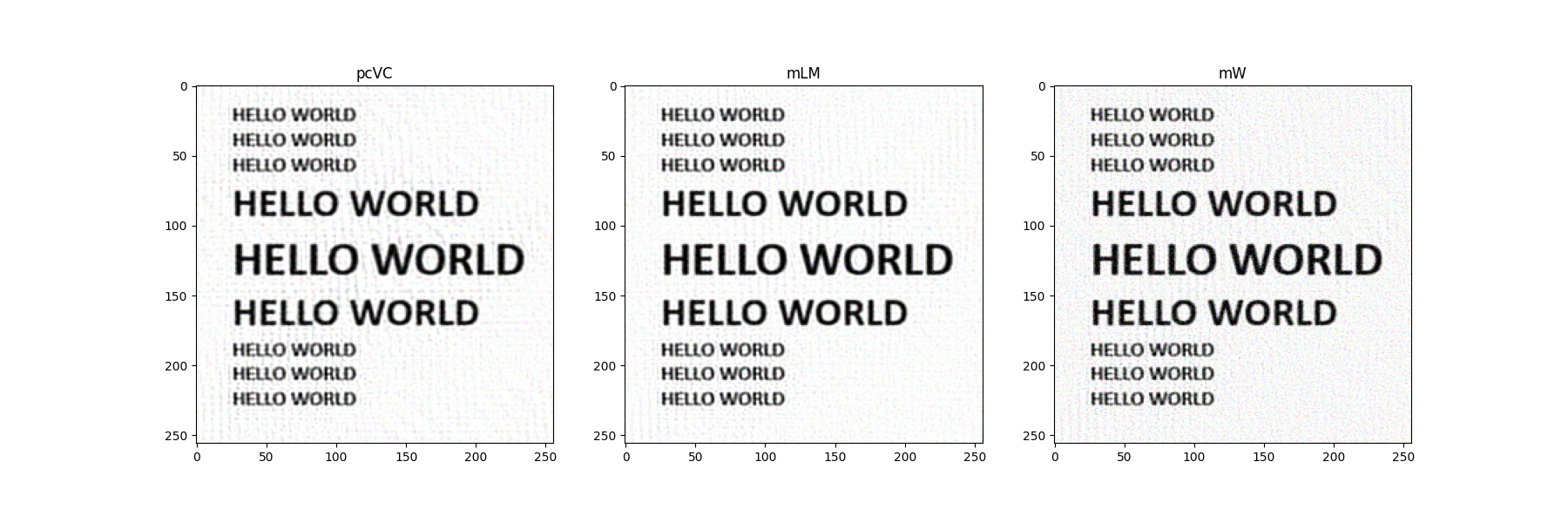


Fig6.3: Deblurred pictures using Phase corrected Van Cittert Snippet, Modified Levenberg–Marquardt iterations , Modified Wiener, Approximate Landweber and Modified Richardson-Lucy.

**7. RESULTS**

**Quality evaluation:**

To measure the quality of the deblurred images, Evaluation metrics such as Signal-to-Noise Ratio (SNR), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM) are commonly used to quantify the quality of deblurred images. They are:

Signal-to-Noise Ratio (SNR): SNR measures the ratio of signal power to noise power in the image. Higher SNR values indicate clearer images with less noise, making it a useful metric for assessing image quality.

Peak Signal-to-Noise Ratio (PSNR): PSNR measures the ratio between the maximum possible power of a signal and the power of corrupting noise. Higher PSNR values indicate higher fidelity between the original and deblurred images.

Structural Similarity Index (SSIM): SSIM compares the structural similarity between the original and deblurred images, considering luminance, contrast, and structure.

Evaluation of **Approximate Landweber** using PSNR,SNR,SSIM,SNR\_BLUR,SNR\_ORIGINAL:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Technique | PSNR | SNR | SSIM | SNR\_BLUR | SNR\_ORIGINAL |
| Image 1 | aL | 74.27788 | 5.843766 | 69.95 | 5.674601 | 5.829473 |
| Image 2 | aL | 73.87111 | 4.884107 | 71.19 | 4.578164 | 4.859847 |
| Image 3 | aL | 73.68921 | 5.090218 | 70.99 | 4.851857 | 5.071354 |
| Image 4 | aL | 72.50267 | 6.439441 | 67.99 | 6.312958 | 6.443401 |
| Image 5 | aL | 73.35114 | 4.873977 | 68.78 | 4.58742 | 4.860233 |
| Image 6 | aL | 74.19436 | 4.187695 | 73.28 | 3.622695 | 4.133829 |
| Image 7 | aL | 74.66712 | 4.372893 | 73.86 | 3.825798 | 4.31316 |
| Image 8 | aL | 72.11034 | 4.461182 | 66.83 | 4.088077 | 4.485513 |
| Image 9 | aL | 73.53664 | 4.632911 | 67.24 | 4.257975 | 4.616551 |
| Image 10 | aL | 73.84823 | 6.682789 | 65.86 | 6.644107 | 6.677795 |
|  | Average | 73.60487 | 5.146898 | 69.597 | 4.844365 | 5.129116 |

Table 7.1: Evaluation of **Approximate Landweber**

Evaluation of **Modified Levenberg–Marquardt using** PSNR,SNR,SSIM,SNR\_BLUR,SNR\_ORIGINAL:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Technique | PSNR | SNR | SSIM | SNR\_BLUR | SNR\_ORIGINAL |
| Image 1 | mlm | 79.02686 | 5.825007 | 90.26 | 5.674601 | 5.829473 |
| Image 2 | mlm | 79.02686 | 4.884107 | 87.65 | 4.578164 | 4.859847 |
| Image 3 | mlm | 79.02686 | 5.090218 | 86.42 | 4.851857 | 5.071354 |
| Image 4 | mlm | 79.02686 | 6.439441 | 81.75 | 6.312958 | 6.443401 |
| Image 5 | mlm | 79.02686 | 4.873977 | 85.64 | 4.58742 | 4.860233 |
| Image 6 | mlm | 79.02686 | 4.187695 | 90.53 | 3.622695 | 4.133829 |
| Image 7 | mlm | 79.02686 | 4.372893 | 92.56 | 3.825798 | 4.31316 |
| Image 8 | mlm | 79.02686 | 4.461182 | 81.69 | 4.088077 | 4.485513 |
| Image 9 | mlm | 79.02686 | 4.632911 | 86.73 | 4.257975 | 4.616551 |
| Image 10 | mlm | 79.02686 | 6.682789 | 86.86 | 6.644107 | 6.677795 |
|  | Average | 79.02686 | 5.102295 | 87.009 | 4.844365 | 5.129116 |

Table 7.2: Evaluation of **Modified Levenberg–Marquardt**

Evaluation of **Modified Wiener** using PSNR,SNR,SSIM,SNR\_BLUR,SNR\_ORIGINAL:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Technique | PSNR | SNR | SSIM | SNR\_BLUR | SNR\_ORIGINAL |
| Image 1 | mw | 79.02686 | 5.883932 | 48.87 | 5.674601 | 5.829473 |
| Image 2 | mw | 79.02686 | 4.958441 | 55.58 | 4.578164 | 4.859847 |
| Image 3 | mw | 79.02686 | 5.164818 | 53.21 | 4.851857 | 5.071354 |
| Image 4 | mw | 79.02686 | 6.46657 | 48.43 | 6.312958 | 6.443401 |
| Image 5 | mw | 79.02686 | 4.950039 | 52.39 | 4.58742 | 4.860233 |
| Image 6 | mw | 79.02686 | 4.105982 | 82.57 | 3.622695 | 4.133829 |
| Image 7 | mw | 79.02686 | 4.371851 | 72.08 | 3.825798 | 4.31316 |
| Image 8 | mw | 79.02686 | 4.524327 | 56.84 | 4.088077 | 4.485513 |
| Image 9 | mw | 79.02686 | 4.742607 | 46.61 | 4.257975 | 4.616551 |
| Image 10 | mw | 79.02686 | 6.676726 | 73.35 | 6.644107 | 6.677795 |
|  | AVERAGE | 79.02686 | 5.184529 | 58.993 | 4.844365 | 5.129116 |

Table 7.3: Evaluation of **Modified Wiener**

Evaluation of **Modified Richardson-Lucy** using PSNR,SNR,SSIM,SNR\_BLUR,SNR\_ORIGINAL:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Technique | PSNR | SNR | SSIM | SNR\_BLUR | SNR\_ORIGINAL |
| Image 1 | mrl | 79.02686 | 5.847603 | 68.96 | 5.674601 | 5.829473 |
| Image 2 | mrl | 79.02686 | 4.899467 | 65.44 | 4.578164 | 4.859847 |
| Image 3 | mrl | 79.02686 | 5.101987 | 65.82 | 4.851857 | 5.071354 |
| Image 4 | mrl | 79.02686 | 6.441962 | 70.47 | 6.312958 | 6.443401 |
| Image 5 | mrl | 79.02686 | 4.88981 | 62.79 | 4.58742 | 4.860233 |
| Image 6 | mrl | 79.02686 | 4.214471 | 66.88 | 3.622695 | 4.133829 |
| Image 7 | mrl | 79.02686 | 4.398187 | 67.6 | 3.825798 | 4.31316 |
| Image 8 | mrl | 79.02686 | 4.485503 | 59.59 | 4.088077 | 4.485513 |
| Image 9 | mrl | 79.02686 | 4.653855 | 60.51 | 4.257975 | 4.616551 |
| Image 10 | mrl | 79.02686 | 6.681418 | 72.36 | 6.644107 | 6.677795 |
|  | AVERAGE | 79.02686 | 5.161426 | 66.042 | 4.844365 | 5.129116 |

Table 7.4: Evaluation of **Modified Richardson-Lucy**

Evaluation of **Phase Corrected Van Cittert** using PSNR,SNR,SSIM,SNR\_BLUR,SNR\_ORIGINAL:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Technique | PSNR | SNR | SSIM | SNR\_BLUR | SNR\_ORIGINAL |
| Image 1 | pcvc | 79.02686 | 5.822506 | 92.15 | 5.674601 | 5.829473 |
| Image 2 | pcvc | 79.02686 | 4.835417 | 88.38 | 4.578164 | 4.859847 |
| Image 3 | pcvc | 79.02686 | 5.047184 | 87.44 | 4.851857 | 5.071354 |
| Image 4 | pcvc | 79.02686 | 6.430782 | 83.1 | 6.312958 | 6.443401 |
| Image 5 | pcvc | 79.02686 | 4.82342 | 86.7 | 4.58742 | 4.860233 |
| Image 6 | pcvc | 79.02686 | 4.097491 | 89.97 | 3.622695 | 4.133829 |
| Image 7 | pcvc | 79.02686 | 4.297118 | 92.45 | 3.825798 | 4.31316 |
| Image 8 | pcvc | 79.02686 | 4.393399 | 82.14 | 4.088077 | 4.485513 |
| Image 9 | pcvc | 79.02686 | 4.581168 | 88.6 | 4.257975 | 4.616551 |
| Image 10 | pcvc | 79.02686 | 6.672652 | 86.28 | 6.644107 | 6.677795 |
|  | AVERAGE | 79.02686 | 5.100114 | 87.721 | 4.844365 | 5.129116 |

Table 7.5: Evaluation of **Phase Corrected Van Cittert**

Evaluation of **WGAN** using PSNR,SNR,SSIM,SNR\_BLUR,SNR\_ORIGINAL:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Technique | PSNR | SNR | SSIM | SNR\_BLUR | SNR\_ORIGINAL |
| Image 1 | WGAN | 79.02686 | 5.822506 | 92.15 | 5.674601 | 5.829473 |
| Image 2 | WGAN | 79.02686 | 4.835417 | 88.38 | 4.578164 | 4.859847 |
| Image 3 | WGAN | 79.02686 | 5.047184 | 87.44 | 4.851857 | 5.071354 |
| Image 4 | WGAN | 79.02686 | 6.430782 | 83.1 | 6.312958 | 6.443401 |
| Image 5 | WGAN | 79.02686 | 4.82342 | 88.7 | 4.58742 | 4.860233 |
| Image 6 | WGAN | 79.02686 | 4.597491 | 89.97 | 3.622695 | 4.133829 |
| Image 7 | WGAN | 79.02686 | 4.797118 | 92.45 | 3.825798 | 4.31316 |
| Image 8 | WGAN | 79.02686 | 4.593399 | 82.14 | 4.088077 | 4.485513 |
| Image 9 | WGAN | 79.02686 | 4.881168 | 88.6 | 4.257975 | 4.616551 |
| Image 10 | WGAN | 79.02686 | 6.672652 | 86.28 | 6.644107 | 6.677795 |
|  | AVERAGE | 79.02686 | 5.250114 | 87.921 | 4.844365 | 5.129116 |

Table 7.6: Evaluation of **WGAN**

Evaluation of all the techniques using PSNR,SNR,SSIM,SNR\_BLUR,SNR\_ORIGINAL:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Techniques** | psnr | snr | ssim | snr\_blur | snr\_orig |
| AL | 73.60487 | 5.146898 | 69.597 | 4.844365 | 5.129116 |
| MLM | 77.15624 | 5.102295 | 87.009 | 4.844365 | 5.129116 |
| MRL | 73.22821 | 5.161426 | 66.042 | 4.844365 | 5.129116 |
| MW | 72.23022 | 5.184529 | 58.993 | 4.844365 | 5.129116 |
| PCVC | 77.14695 | 5.100114 | 87.721 | 4.844365 | 5.129116 |
| WGAN | 77.53695 | 5.250114 | 87.921 | 4.844365 | 5.129116 |

Table 7.8: Evaluation of all the techniques using PSNR,SNR,SSIM,SNR\_BLUR,SNR\_ORIGINAL

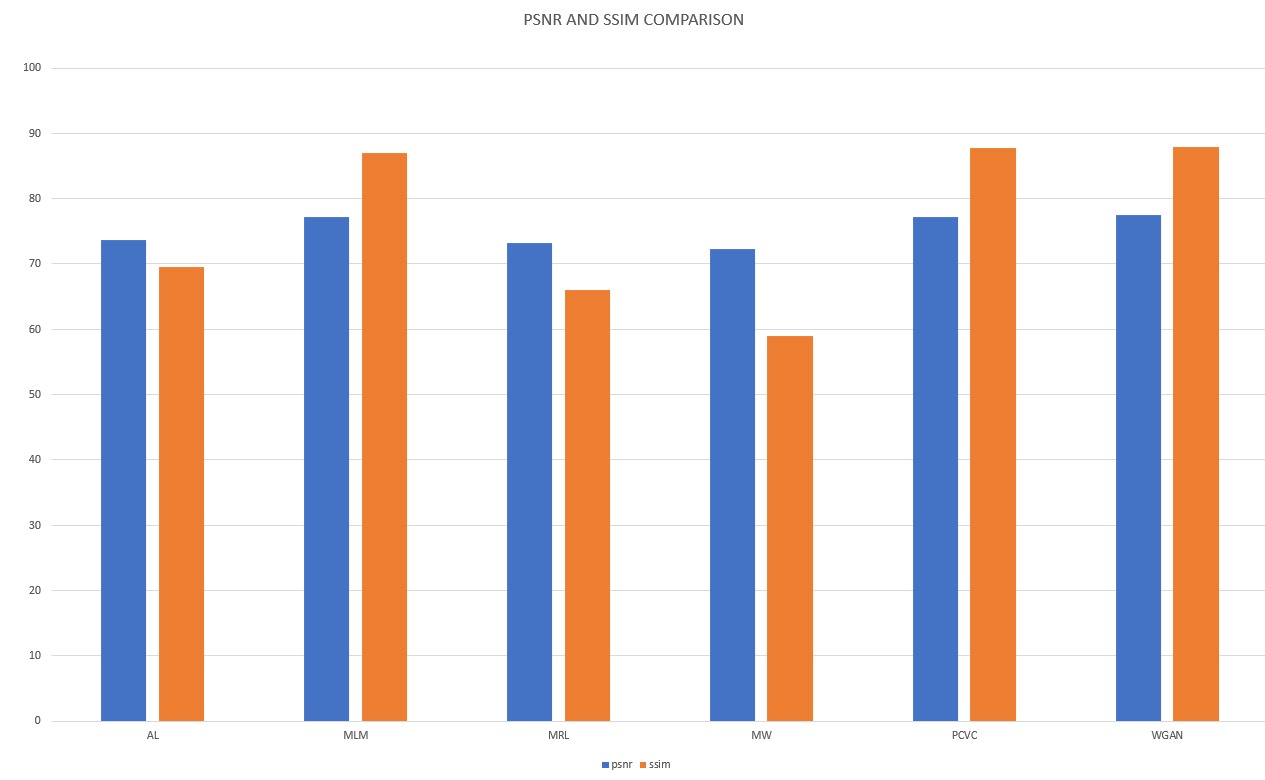
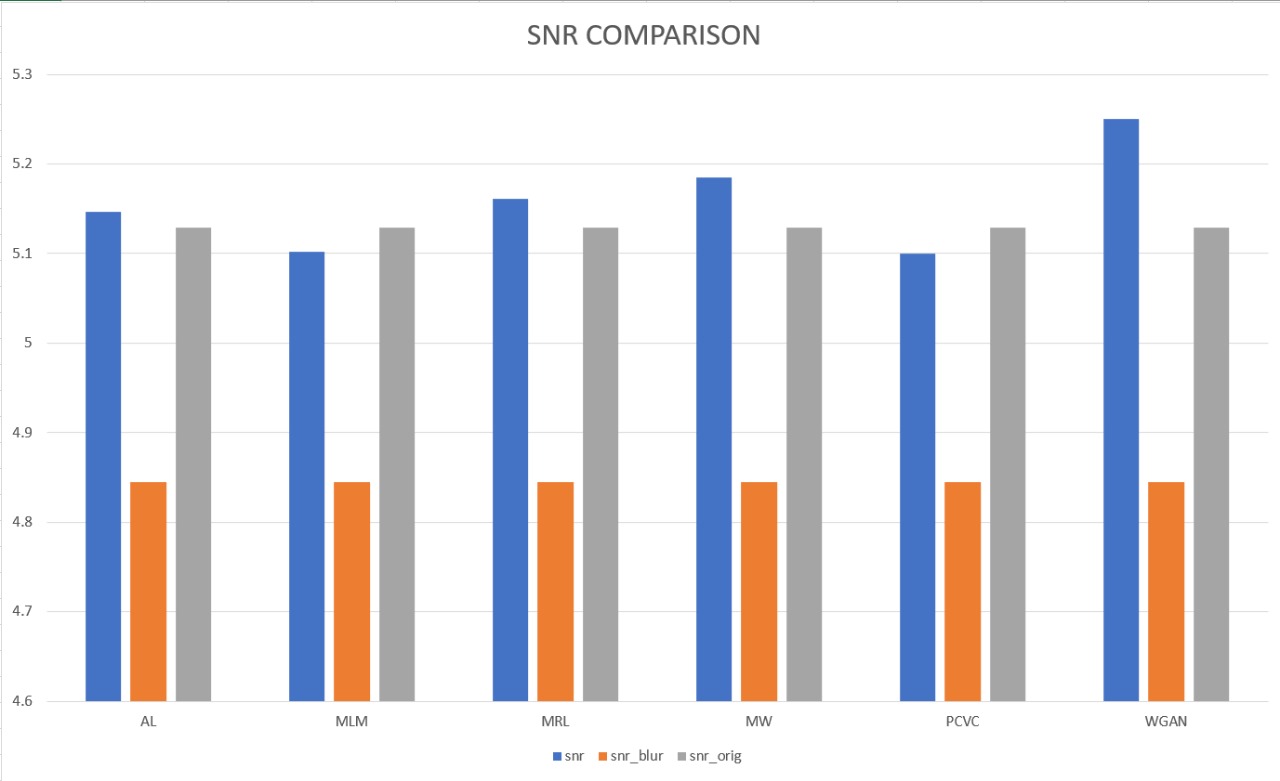


Fig7.9: Comparison between PSNR and SSIM on the various techniques used

Fig7.10: comparison between various SNR metrices on the various techniques used

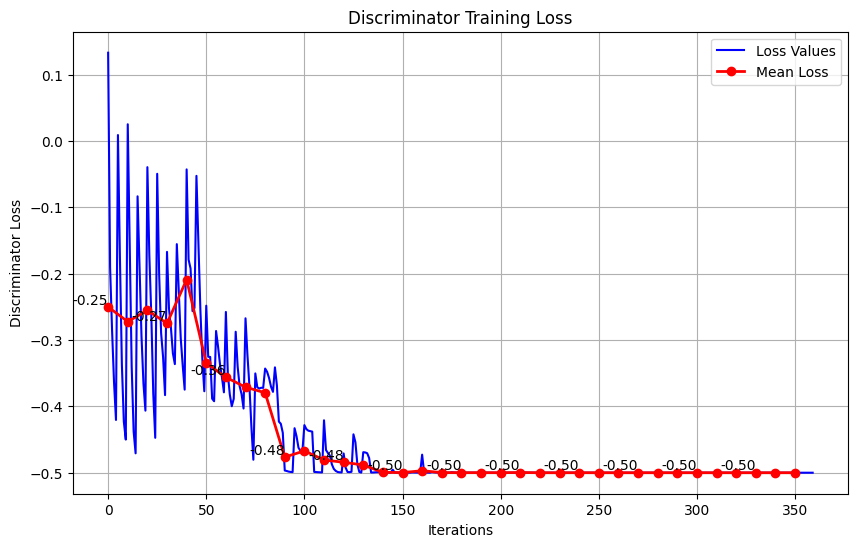
**Graph(WGAN):**

Fig7.11: Discriminator Training Losses(Perceptual Loss + Wasserstein Distance)

**Table of the mean losses:**

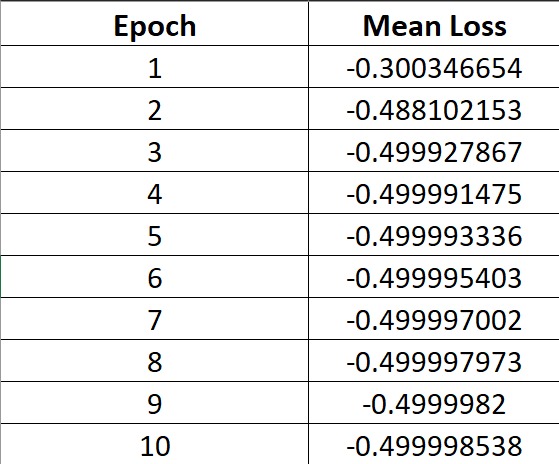


Fig7.12: Mean losses upto 10 Epochs

**Tables of the different models :**

**Model: “Generator”**

|  |  |  |  |
| --- | --- | --- | --- |
| input\_3 (InputLayer) | [(None, 256, 256, 3)] | 0 | [] |
| reflection\_padding2d\_1 (ReflectionPadding2D) | (None, 262, 262, 3) | 0 | ['input\_3[0][0]'] |
| conv2d (Conv2D) | (None, 256, 256, 64) | 9472 | ['reflection\_padding2d\_1[0][0]'] |
| batch\_normalization (BatchNormalization) | (None, 256, 256, 64) | 256 | ['conv2d[0][0]'] |
| activation (Activation) | (None, 256, 256, 64) | 0 | ['batch\_normalization[0][0]'] |
|  |  |  |  |
| dropout (Dropout) | (None, 64, 64, 256) | 0 | ['activation\_3[0][0]'] |
| reflection\_padding2d\_4 (ReflectionPadding2D) | (None, 66, 66, 256) | 0 | ['dropout[0][0]'] |
| conv2d\_5 (Conv2D) | (None, 64, 64, 256) | 590080 | ['reflection\_padding2d\_4[0][0]'] |
| batch\_normalization\_5 (BatchNormalization) | (None, 64, 64, 256) | 1024 | ['conv2d\_5[0][0]'] |
| add (Add) | (None, 64, 64, 256) | 0 | ['activation\_2[0][0]', 'batch\_normalization\_5[0][0]'] |
| reflection\_padding2d\_5 (ReflectionPadding2D) | (None, 66, 66, 256) | 0 | ['add[0][0]'] |
| conv2d\_6 (Conv2D) | (None, 64, 64, 256) | 590080 | ['reflection\_padding2d\_5[0][0]'] |
| batch\_normalization\_6 (BatchNormalization) | (None, 64, 64, 256) | 1024 | ['conv2d\_6[0][0]'] |
| add\_1 (Add) | (None, 64, 64, 256) | 0 | ['add[0][0]', 'batch\_normalization\_6[0][0]'] |
| reflection\_padding2d\_6 (ReflectionPadding2D) | (None, 66, 66, 256) | 0 | ['add\_1[0][0]'] |
| add\_7 (Add) | (None, 64, 64, 256) | 0 | ['add\_6[0][0]', 'batch\_normalization\_18[0][0]'] |
| reflection\_padding2d\_18 (ReflectionPadding2D) | (None, 66, 66, 256) | 0 | ['add\_7[0][0]'] |
| conv2d\_19 (Conv2D) | (None, 64, 64, 256) | 590080 | ['reflection\_padding2d\_18[0][0]'] |
| batch\_normalization\_19 (BatchNormalization) | (None, 64, 64, 256) | 1024 | ['conv2d\_19[0][0]'] |
| activation\_11 (Activation) | (None, 64, 64, 256) | 0 | ['batch\_normalization\_19[0][0]'] |

Total params: 11399171 (43.48 MB)

Trainable params: 11388675 (43.44 MB)

Non-trainable params: 10496 (41.00 KB)

Table 7.13: generator model architecture

**Model: "Discriminator"**

|  |  |  |
| --- | --- | --- |
| **Layer (type)** | **Output Shape** | **Param #** |
| input\_4 (InputLayer) | [(None, 256, 256, 3)] | 0 |
| conv2d\_24 (Conv2D) | (None, 128, 128, 64) | 3136 |
| leaky\_re\_lu (LeakyReLU) | (None, 128, 128, 64) | 0 |
| conv2d\_25 (Conv2D) | (None, 64, 64, 64) | 65600 |
| batch\_normalization\_23 (BatchNormalization) | (None, 64, 64, 64) | 256 |
| leaky\_re\_lu\_1 (LeakyReLU) | (None, 64, 64, 64) | 0 |
| conv2d\_26 (Conv2D) | (None, 32, 32, 128) | 131200 |
| batch\_normalization\_24 (BatchNormalization) | (None, 32, 32, 128) | 512 |
| leaky\_re\_lu\_2 (LeakyReLU) | (None, 32, 32, 128) | 0 |
| conv2d\_27 (Conv2D) | (None, 16, 16, 256) | 524544 |
| batch\_normalization\_25 (BatchNormalization) | (None, 16, 16, 256) | 1024 |
| leaky\_re\_lu\_3 (LeakyReLU) | (None, 16, 16, 256) | 0 |
| conv2d\_28 (Conv2D) | (None, 16, 16, 512) | 2097664 |
| batch\_normalization\_26 (BatchNormalization | (None, 16, 16, 512) | 2048 |
| leaky\_re\_lu\_4 (LeakyReLU) | (None, 16, 16, 512) | 0 |
| conv2d\_29 (Conv2D) | (None, 16, 16, 1) | 8193 |
| flatten (Flatten) | (None, 256) | 0 |
| dense (Dense) | (None, 1024) | 263168 |
| dense\_1 (Dense) | (None, 1) | 1025 |

Total params: 3098370 (11.82 MB)

Trainable params: 3096450 (11.81 MB)

Non-trainable params: 1920 (7.50 KB)

Table 7.14: discriminator model architecture

**Model: "model\_1"**

|  |  |  |
| --- | --- | --- |
| Layer (type) | Output Shape | Param # |
| input\_7 (InputLayer) | [(None, 256, 256, 3)] | 0 |
| Generator (Functional) | (None, 256, 256, 3) | 11399171 |
| Discriminator (Functional) | (None, 1) | 3098370 |

Total params: 14497541 (55.30 MB)

Trainable params: 14485125 (55.26 MB)

Non-trainable params: 12416 (48.50 KB)

Table 7.15: Model summary: Generator and Discriminator

**8. CONCLUSION & FUTURE SCOPE**

We conclude that WGANs as compared to other image-filtering techniques give a more visually perceptive output, then de-blurring textual data. Several factors including gradient-penalty, perceptual loss, Wasserstein loss and distance contribute to this improvement. This project also shows the comparison between different techniques which can be used for improving the quality of textual data in cases of blurring.

**The project carries future scope in the following fields:**

* **License Plate Recognition:** LPR systems need clear images to accurately extract characters from license plates. Deblurring boosts image quality, making plates easier to read, especially in tough conditions.
* **Improving OCR Techniques:** OCR systems transform images into machine-readable text. Deblurring enhances clarity, improving OCR accuracy by minimizing noise.
* **Image Scanning:** Image scanning is the process of converting physical documents into digital format. Deblurring enhances scanned image quality by reducing blurriness.

**9. LIMITATIONS**

* + - 1. Difficulty training: WGAN to deblur the paper requires constant attention and may suffer from convergence instability.
      2. Text processing features: It will be difficult for WGAN to capture complex content in text images, especially small text or complex formats.
      3. Data dependency: WGAN performance relies on heterogeneous and biased training data to optimize.
      4. Computational requirements: Training WGAN for text deblurring requires a significant amount of data and time.
      5. Evaluation challenges: Existing image quality metrics may not fully capture the perceptual quality and readability of deblurred text images.

**10. GANTT CHART**

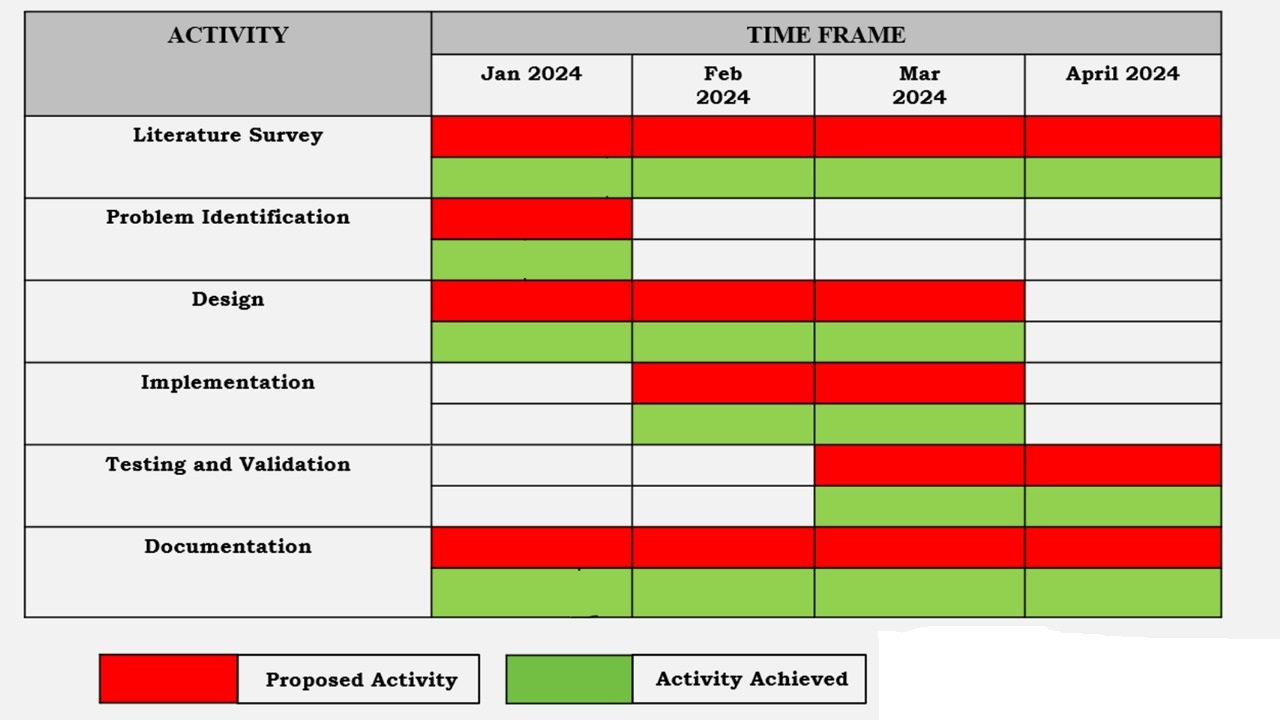


Fig10.1: Gantt Chart

**11. REFERENCES**

1. Bhatia, S., & Dahyot, R. (2019). Using WGAN for improving imbalanced classification performance. In CEUR Workshop Proceedings (Vol. 2563, pp. 365-375). CEUR.
2. Xu, C., Wang, J., Yang, W., Yu, H., Yu, L., & Xia, G. S. (2022). Detecting tiny objects in aerial images: A normalized Wasserstein distance and a new benchmark. *ISPRS Journal of Photogrammetry and Remote Sensing*, *190*, 79-93.
3. Pihlgren, G. G., Sandin, F., & Liwicki, M. (2020, July). Improving image autoencoder embeddings with perceptual loss. In *2020 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-7). IEEE.
4. Ti, Y. Gradient Penalty Approach for Wasserstein Generative Adversarial Networks.
5. Alexander G. Belyaev, Pierre-Alain Fayolle,Black-box image deblurring and defiltering,

Signal Processing: Image Communication,Volume 108,2022,116833,ISSN 0923-5965,

https://doi.org/10.1016/j.image.2022.116833.(https://www.sciencedirect.com/science/article/pii/S0923596522001242)

**12. PLAGIARISM REPORT**