Unpaired Document Image Denoising for OCR using BiLSTM enhanced CycleGAN

by

Katyani Singh

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

University of Alberta

Abstract

The recognition performance of Optical Character Recognition (OCR) models can be sub-optimal when document images suffer from various degradations. Supervised learning-based methods for image enhancement can generate high-quality enhanced images. However, these methods require the availability of corresponding clean images or ground truth text for training. Moreover, the paired training data used for training these models is usually generated by adding different types of synthetic noise to clean images. Real-world noise is more challenging and complex in nature compared to synthetic noise. To effectively enhance real-world noisy images, the models must be trained using real noisy images. However, it is infeasible to have corresponding clean images for real-world noisy images, and creating ground truth text requires manual effort. Unsupervised methods have been explored in recent years, focusing on enhancing natural scene images. In the case of document images, preserving the readability of text in the enhanced images is of utmost importance for improved OCR performance. In this thesis, we explore the possibility of enhancing documents in an unsupervised setting using unpaired training samples. To this end, we propose a modified architecture for the standard CycleGAN model to improve its performance in enhancing document images with better text preservation. The results indicate that the proposed model leads to better preservation of text and improved OCR performance compared to the CycleGAN model and classical unsupervised image preprocessing techniques like Sauvola and Otsu.

Preface

This thesis is the compilation of the journal paper under submission to the International Journal on Document Analysis and Recognition (IJDAR), co-authored with Ganesh Tata, Eric Van Oeveren, and Nilanjan Ray. The work is the original contribution of Katyani Singh, while the co-authors contributed to the discussions.

To my Nanaji

For always inspiring me.

"I haven't failed, I've found 10,000 ways that don't work."

– Thomas Edison.

Acknowledgements

I would like to express my sincere thanks to my supervisor Nilanjan Ray for acknowledging the potential in me to pursue this work. I am grateful for his mentorship throughout the course of my research. His guidance has been instrumental in successful completion of this work. I would like to thank Ganesh Tata, fellow MSc. student and Eric Van Oeveren from Intuit Inc. for providing valuable feedback and ideas throughout the course of this research. Also, I express my gratitude to Intuit AI Research and NSERC for funding this research.

Finally, I would like to thank my grandparents, parents, Anmol, Aditi, Anjaneya, Kanishk, Kalpana, Kamalendu and all the other family members for always believing in me and encouraging me.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

With the advancements in AI, machine learning and deep learning models are widely being used by organizations in building data-driven AI solutions. When dealing with paper documents such as invoices, point-of-sale receipts, forms, and articles, digitization is necessary to extract the information from the images. Optical Character Recognition (OCR) technologies are used to convert the handwritten or printed text in these images into a computer-understandable electronic form. Over time, OCR engines have significantly improved in terms of recognition accuracy, multi-language support, and the ability to handle various handwriting styles [7], [53], [61], [80]. However, these OCR engines perform optimally when the input document images are free of noise. In real-world scenarios, noise can be present due to factors such as uneven illumination during image capture, faded text caused by low printer ink, or the presence of coffee, or ink stains [5], [16], [18], [115]. Such degradations in the image directly impact the OCR performance. Therefore, it is crucial to reduce the impact of these degradations on document images before performing OCR. By enhancing the document images, the OCR recognition accuracy can be improved, and the extracted information becomes more reliable. This thesis addresses the problem of document image enhancement as a preprocessing step to enhance the performance of OCR engines.

Previous works in deep learning have approached image enhancement as a supervised learning task [97], [98], [118], [120]. While these approaches have

been quite effective at reducing the presence of various noise artifacts from the images, these models require supervision in the form of clean ground truth images or ground truth text for training. If these models are trained on noisy-clean image pairs, the paired training data is usually generated by adding different types of synthetic noise to clean images. When these models are utilized for enhancing real-world noisy images, the performance is often sub-optimal [1], [44], [73], [108]. Real-world noise is more challenging and complex in nature compared to synthetic noise. To effectively enhance real-world noisy images, the models must be trained using real noisy images. However, it is infeasible to have corresponding clean images for every real-world noisy image to train these supervised models. On the other hand, creating ground truth text annotation for training is a tedious task requiring manual effort. These challenges motivated us to explore unsupervised methods for achieving this task.

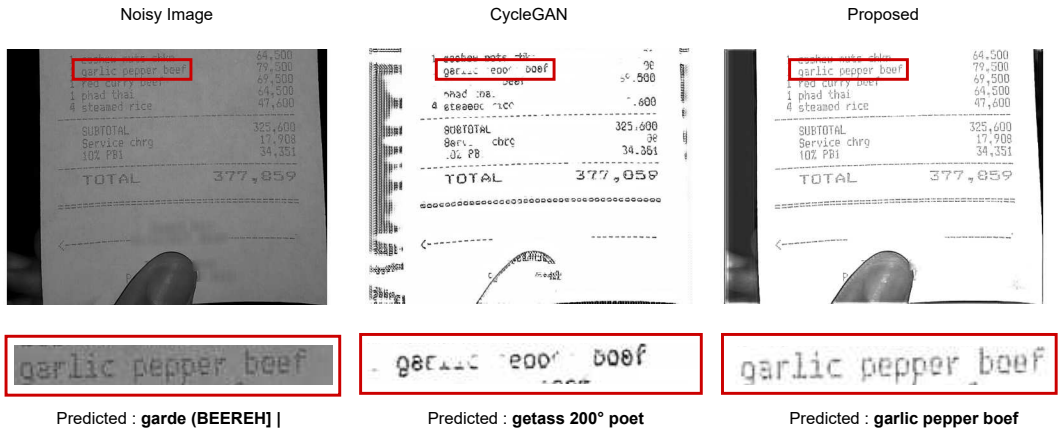


Figure 1.1: Noisy image cleaned by CycleGAN compared with the proposed model.

With the remarkable success of Generative Adversarial Networks (GANs) [21], various models have been proposed to achieve image-to-image translation in an unsupervised setting [24], [43], [70], [113], [121]. These methods have achieved impressive results in style transfer between natural scenes. The task of denoising images can also be formulated as an image-to-image translation task, where the objective is to learn a mapping that transforms an image in the noisy domain into an image in the clean domain [91]. Previously, models

for unpaired image translation have been utilized for unpaired image denoising tasks in natural scene images [31], [41], [90], [107], [112], [114]. However, the application of these unsupervised methods to enhancing document images has not been extensively explored. This encouraged us to explore this avenue.

It is important to understand that image-to-image translation for document images is different from natural scene images due to the presence of textual content in addition to the visual structure in the images [4], [19]. Effectively enhancing document images requires not only the elimination of noise but also preserving the textual content. Degradation of the text during enhancement would directly impact the OCR performance, even if the background noise is removed. Therefore, it becomes paramount to focus on preserving the textual content during the translation from one style to another.

One of the most popular GANs proposed for unpaired image-to-image translation is CycleGAN [121]. The CycleGAN model has achieved remarkable performance in various unpaired image style translation tasks, including denoising natural scene images. However, we observe that its direct application for unpaired document image denoising does not yield satisfactory results. We notice that while the model is capable of eliminating the degradation present in the document image, it fails to preserve the textual contents of the original image. As a result, the generated clean image, although free of degradations, often has distorted text, as shown in Figure 1.1. This distortion of the text leads to poor OCR performance. Therefore, to utilize the CycleGAN model effectively for this task, certain improvements are required.

The underlying framework of GANs consists of two networks: the generator and the discriminator. The goal of the generator is to generate new samples that match the underlying distribution of the real images. The discriminator, on the other hand, acts as a classifier to distinguish between real images and the samples generated by the generator.

Most GANs utilize a Convolutional Neural Network (CNN) in the discriminator network. While CNNs have strong capabilities for extracting meaningful features from images [34], [88], [95], we hypothesize that due to the presence of text in document images, it becomes important to extract stronger features

that also capture the text well. Image text recognition models [27], [38], [86], [87] widely adopt a combination of CNN and a sequential network to capture the local and contextual information present in the image. The local image features extracted by the CNN are enhanced by a Recurrent Neural Network (RNN) model that extracts sequential dependencies. This combination has shown superior results in extracting meaningful features from images for text recognition in document images. Exploring this combination, previous works, such as [106], have proposed the integration of a CNN-LSTM-CTC-based text recognition module alongside the discriminator network as a supervising signal for preserving the text characters.

1.2 Research Statement

In this thesis, we address the problem of unpaired document image enhancement for OCR using an enhanced CycleGAN model. Our hypothesis is that by incorporating a Bidirectional Long Short Term Memory network (BiLSTM) with robust sequential modeling capabilities into the discriminator network of the standard CycleGAN model, we can enhance the preservation of textual content during the translation process from noisy to clean document images.

1.3 Contributions

The main contributions of this thesis are as follows:

- We present a framework that is capable of enhancing real-world noisy documents in an unsupervised setting without the use of noisy/clean image pairs, ground-truth text, or metadata such as noise type.
- We demonstrate the effectiveness of our proposed discriminator architecture in better preserving the textual content during the enhancement and achieving superior performance across three different OCR engines compared to the standard CycleGAN model and classical unsupervised image pre-processing techniques like Sauvola and Otsu.

1.4 Thesis Outline

This thesis is structured into 8 chapters including the introduction. Following this outline of this thesis, in Chapter 2, we introduce the readers to the required deep learning background for understanding the work presented in this thesis. In Chapter 3, we familiarize the readers with some of the previous works related to document image preprocessing including conventional techniques, deep learning-based approaches as well as works utilizing GAN. In Chapter 4, we describe the details of our proposed model, including the network architectures, objective functions, and training procedure. Following this, in Chapter 5, we familiarize the readers with the procedure used in this work for preparing the data for training and evaluation of our proposed model. In Chapter 6, we begin by providing details of our experimental setup including OCR engines, evaluation metrics as well as hyperparameters and training details. Next, we present the results and analysis of the performance of our proposed model, along with comparisons with other baselines. We also highlight some additional ideas we explored. Finally, in Chapter 7, we conclude this thesis with a discussion of the limitations of the proposed work, future directions for this work, and some closing thoughts.

Chapter 2

Background

This chapter familiarizes readers with the background required to understand the work presented in this thesis. In Section 2.1, we begin by introducing Generative Adversarial Networks to provide readers with an overview of these networks and the key ideas involved. We elaborate on the architecture and the overall objective function of these networks. Additionally, we also discuss some challenges involved in training these networks. Next, in Section 2.2, we provide a detailed explanation of the CycleGAN model. Since our proposed model builds upon the CycleGAN framework, this section lays the groundwork for understanding our modifications. Finally, in Section 2.3, we introduce readers to Bidirectional Long Short Term Memory networks (BiLSTM) and their utilization in processing sequential data. We use BiLSTM networks as a key component in our proposed model.

2.1 Generative Adversarial Networks

2.1.1 Overview

Generative Adversarial Networks (GAN) is a type of generative model [21]. The goal of generative models is to generate new samples belonging to a certain data distribution. GANs have shown a remarkable ability to generate realistic data for various generative tasks across different modalities such as images [23], [42], [117], videos [54], [99], [101], audio [9], [48], [56], and text [17], [111], [119].

The key theory behind GANs is adversarial learning. In an adversarial

learning setting, two players compete against each other in a battle, where each one is learning to be better than the other, and in this process, eventually both these players become better. GANs employ a generator and a discriminator model, each being a neural network in an adversarial setting. The goal of the generator is to generate samples closely matching the underlying data distribution of the real samples while the discriminator attempts to distinguish between real and generated samples. Through this two-player game, eventually, the generator learns to generate samples that closely resemble the real samples.

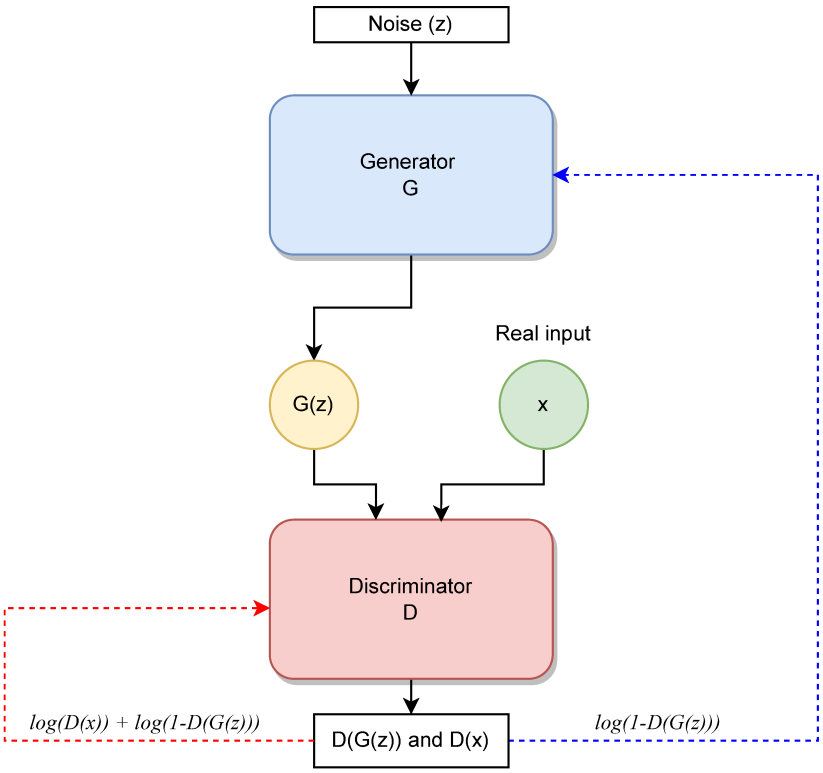


Figure 2.1: GAN architecture.

2.1.2 Architecture

Figure 2.1 shows the architecture of the GAN framework. Both G and D are neural networks. The generator G takes in a random noise input z following

the Gaussian distribution and generates an output $G(z)$. The discriminator D takes in the real input x and the output of the generator $G(z)$, tries to distinguish real data from the generated data, and outputs the probability that the input was real. $D(x)$ represents the probability of D predicting that x was real and $D(G(z))$ represents the probability of D predicting that $G(z)$ was real.

2.1.3 Objective function

Equation 2.1 shows the objective function of the GAN framework.

$$\min_G \max_D L(D,G) = \mathbb{E}_{x\sim p_r(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \quad (2.1)$$

The objective is a minimax function. The first term $\mathbb{E}_{x\sim p_r(x)}[\log D(x)]$ represents the log probability of D predicting the real data x as real. The second term $\mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]$ represents the log probability of D predicting the generated data $G(z)$ as not real. During training, D should learn to accurately predict the real data as real and the generated data as not real. Hence, the training of D involves maximizing the objective function. On the other hand, G has to attempt to fool D into predicting generated data $G(z)$ as real. Therefore, it has to minimize $[1 - D(G(z))]$. G is therefore trained to minimize the objective function.

The above equation can be rewritten as:

$$\min_G \max_D L(D,G) = \mathbb{E}_{x\sim p_r(x)}[\log D(x)] + \mathbb{E}_{x\sim p_g(x)}[\log(1 - D(x)] \quad (2.2)$$

where p_r and p_g represent the real data and the generated data distributions respectively. From this equation, for getting the optimal value of D, we need to maximize $L(D,G)$.

$$L(G,D) = \int_x (p_r(x)\log(D(x)) + p_g(x)\log(1 - D(x)))\,dx \quad (2.3)$$

To maximize $L(D,G)$, we need the best value for $D(x)$ [1].

[1] Referred from [105]

$$f(D(x)) = p_r(x)\log(D(x)) + p_g(x)\log(1 - D(x)) \tag{2.4}$$

Calculating $\frac{d(f(D(x))}{d(D(x))}$) and equating it to 0, we get $D^* = \frac{p_r(x)}{p_r(x)+p_g(x)}$.

Considering optimal G^*, $p_r(x) = p_g(x)$, therefore, $D^* = \frac{1}{2}$.

Putting these optimal values, we can get the global optimal value for the objective function as:

$$\begin{aligned} L\left(G^*, D^*\right) &= \int_x \left(p_r(x)\log\left(D^*(x)\right) + p_g(x)\log\left(1 - D^*(x)\right)\right) dx \\ &= \log\frac{1}{2}\int_x p_r(x)dx + \log\frac{1}{2}\int_x p_g(x)dx \\ &= -2\log 2 \end{aligned} \tag{2.5}$$

If D is optimal, the loss function represents minimizing the Jensen-Shannon (JS) divergence [55] between the real data distribution and generated data distribution. To understand this, it is important to understand how generative models generate data. Generative networks learn by minimizing the difference between the real data and the generated data distributions. There are different ways to measure the similarity or difference between the two distributions. One such measure is Kullback–Leibler (KL) Divergence [47] which calculates the divergence of one probability distribution from another reference probability distribution. For two probability distributions p and q, KL divergence of p from q and q from p is given by:

$$D_{KL}(p\|q) = \int_x p(x)\log\frac{p(x)}{q(x)}dx \tag{2.6}$$

$$D_{KL}(q\|p) = \int_x q(x)\log\frac{q(x)}{p(x)}dx \tag{2.7}$$

Derived from KL divergence, JS divergence is another measure of similarity between two probability distributions. For probability distributions p and q, JS divergence is given by:

$$D_{JS}(p\|q) = \frac{1}{2}D_{KL}\left(p\|M\right) + \frac{1}{2}D_{KL}\left(q\|M\right) \tag{2.8}$$

where,

$$M = \frac{p+q}{2}$$

For p_r and p_g representing the real data and the generated data distributions, using Equation 2.5, it can be shown below that for an optimal D, the training of GANs is equivalent to minimizing the JS divergence between p_r and p_g.

$$\begin{aligned} D_{JS}\left(p_r \| p_g\right) =& \frac{1}{2} D_{KL}\left(p_r \| \frac{p_r + p_g}{2}\right) + \frac{1}{2} D_{KL}\left(p_g \| \frac{p_r + p_g}{2}\right) \\ =& \frac{1}{2}\left(\log 2 + \int_x p_r(x) \log \frac{p_r(x)}{p_r + p_g(x)} dx\right) + \\ & \frac{1}{2}\left(\log 2 + \int_x p_g(x) \log \frac{p_g(x)}{p_r + p_g(x)} dx\right) \\ =& \frac{1}{2}\left(\log 4 + L\left(G^*, D^*\right)\right) \end{aligned}$$

$$L\left(G^*, D^*\right) = 2 D_{JS}\left(p_r \| p_g\right) - 2 \log 2 \tag{2.9}$$

2.1.4 Pitfalls of GAN

There are several challenges to training a GAN, where two neural networks are being trained simultaneously with one trying to maximize the objective function and the other minimizing it. In many cases, the training can be highly unstable. Instability also arises from vanishing gradients. For a perfect D, $D(x) = 1, \forall x \in p_r$ and $D(x) = 0, \forall x \in p_g$. This leads to overall loss becoming 0 in Equation 2.2. Ideally, we want D to be perfect but with loss 0, the gradients vanish over time and G cannot be updated well. On the other hand, if D is not perfect, G would not receive accurate feedback to update itself. Another issue with training GANs is *Mode Collapse* [82]. Mode Collapse happens when the generator tries to map all the inputs to a small space of outputs in the target domain for which it can fool the discriminator. With this, although it accomplishes fooling the discriminator, the generator fails to learn the data distribution effectively.

2.2 CycleGAN

2.2.1 Overview

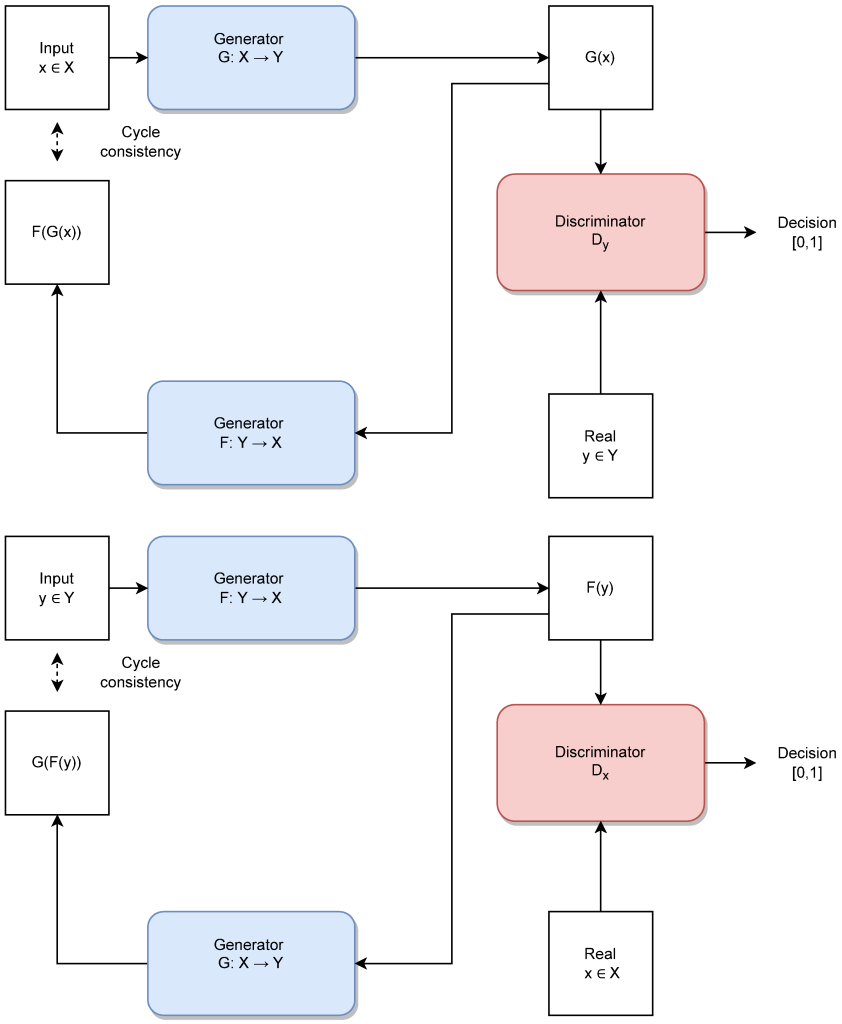


Figure 2.2: CycleGAN architecture.

The CycleGAN framework [121] is a specialized type of GAN proposed for unpaired image-to-image translation tasks. In an image-to-image translation task, the goal is to transform an input image in the source domain to an output image in the target domain using paired data samples for training. In certain use cases, it is challenging to obtain paired data to achieve the translation. To alleviate this problem, CycleGAN attempts to learn meaningful mappings between unpaired source and target images by utilizing the transitivity property in the form of a *cycle consistency* loss.

2.2.2 Cycle consistency

Consider a translation function $F_1 : X \to Y$ that maps an input x in domain X to an output y in domain Y and another function $F_2 : Y \to X$ that maps an input y in domain Y to an output x in domain X, cycle consistency enforces that $F_2(F_1(x)) \approx x$ and $F_1(F_2(y)) \approx y$. This means that if an image is transformed from one domain to another and then reverse-transformed, the generated samples should be close to the source domain. In the absence of cycle consistency, the generator network can learn to transform the set of input images to any random set of images in the target domain for which the output matches the target distribution.

2.2.3 Model architecture

Consider Figure 2.2, there are two generators $G : X \to Y$ and $F : Y \to X$. In the top diagram, the generator G maps an input image from the source domain X to an output image in the target domain Y. Generator F takes in the generated image and performs the inverse transformation from the target domain Y back to the source domain X. The discriminator D_y aims to distinguish between the real y and the generated image $G(x)$. Cycle consistency is ensured between x and $F(G(x))$.

Similarly, in the bottom diagram, the generator F maps an input image from the target domain Y to an output image in the source domain X. Generator G takes in the generated image and performs the inverse transformation from the source domain X back to the target domain Y. The discriminator D_x aims to distinguish between the real x and the generated image $F(y)$. Cycle consistency is ensured between y and $G(F(y))$. Therefore, the bi-directional conversion for image-to-image translation in CycleGAN is achieved by the use of two generators and two discriminators.

In the CycleGAN model, the generator network comprises two convolutional layers of stride 2, followed by a few residual blocks, and finally, two layers of transposed convolutions with stride 1. The discriminator network is a 70×70 CNN-based network called PatchGAN [36], [50], [52] that classifies

the 70×70 overlapping patches of images as real or fake. PatchGAN provides the output in the form of an array in which each number signifies whether its corresponding patch is real or fake. The discriminator output is taken as the average of the prediction for each patch.

2.2.4 Objective function

The goal is to learn a mapping function between a source domain X to a target domain Y given the training samples: $\{x_i\}_n^{i=1}$, $x_i \in X$ and $\{y_j\}_m^{j=1}$, $y_j \in Y$, with distributions $x \sim P_X(x)$ and $y \sim P_Y(y)$. The overall objective function in CycleGAN comprises two losses - the GAN loss that enforces the mapping of the image style from one domain to another and the cycle consistency loss that ensures that the contents of the original image remain preserved during the style transfer. Equations 2.10 and 2.11 shows the two adversarial losses in CycleGAN.

$$\mathcal{L}_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_Y(y)}[log(D_Y(y))] + \mathbb{E}_{x \sim p_X(x)}[log(1 - D_Y(G(x))] \tag{2.10}$$

$$\mathcal{L}_{GAN}(F, D_X, X, Y) = \mathbb{E}_{x \sim p_X(x)}[log(D_X(x))] + \mathbb{E}_{y \sim p_Y(y)}[log(1 - D_X(F(y))] \tag{2.11}$$

The cycle consistency loss is defined as :

$$\mathcal{L}_{cyc}(G, F) = \mathbb{E}_{x \sim p_X(x)}[||F(G(x)) - x||_1] + \mathbb{E}_{y \sim p_Y(y)}[||G(F(y)) - y||_1] \tag{2.12}$$

Adding the adversarial and cycle consistency loss, the overall objective function is defined as:

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, Y, X) + \lambda_{cyc}\mathcal{L}_{cyc}(G, F) \tag{2.13}$$

Here, λ_{cyc} controls the weight of cycle consistency loss.

The parameters G, F, D_X, D_Y are learned through optimization of the overall objective function as:

$$G^*, F^* = \arg\min_{G,F} \max_{D_X, D_Y} \mathcal{L}(G, F, D_X, D_Y) \tag{2.14}$$

2.3 Bidirectional Long Short Term Memory (BiLSTM)

One of the limitations of CNN is the inability to preserve the sequential order of information during processing. For data that is sequential in nature such as text, and speech, Recurrent Neural Networks (RNN) are used due to their capability for sequential modeling [15], [81]. RNNs make use of information from previous states for processing current state output. This is done with the use of a hidden state that stores outputs from the previous states. However, for longer sequences, simple RNNs suffer from the problem of *Vanishing gradients* [6]. Long Short-Term Memory Networks (LSTM) overcome this problem and are more suitable for handling long-range sequential dependencies [29].

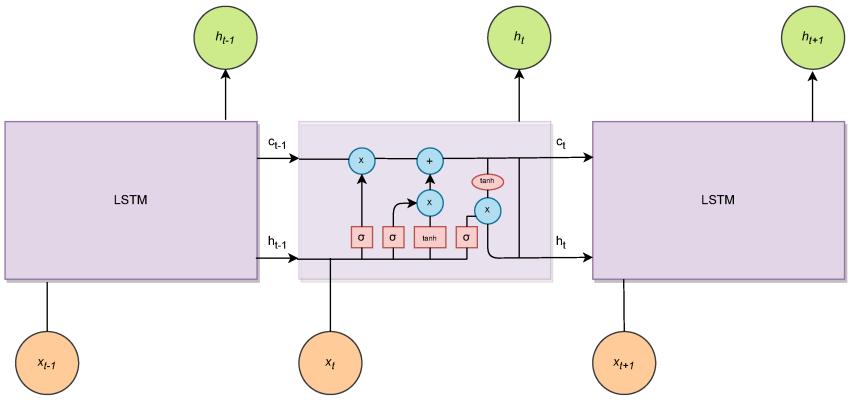


Figure 2.3: A simple LSTM module.

Figure 2.3 shows a simple LSTM module. At any time step t, we have the input vector x_t as well as the previous step hidden state h_{t-1}. The transmission of information takes place through the *Cell State* c running through all the time steps. Information is added or removed from the cell state using gates. The first σ represents the *Forget Gate*, which takes in the previous hidden state and the current input and outputs a value between 0 and 1. If previous information is no longer useful and should be completely erased, it outputs 0. If previous information is strongly valuable and should be preserved, it outputs 1. This is multiplied with c_{t-1}. The second σ represents the *Input Gate*, that is used to decide the importance of the current input information x_t. x_t and

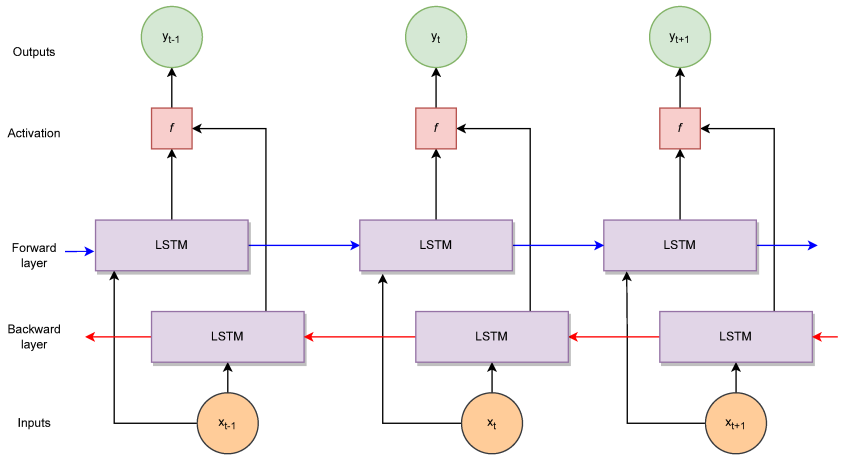


Figure 2.4: BiLSTM network.

h_{t-1} are passed through a tanh activation that clips values between -1 and 1. A new vector for the new cell state to be updated is created. This is multiplied by the output of the input gate and added to the cell state. The new cell state is C_t which is passed to the next time step $t+1$. The last σ represents the *Output Gate*, which decides the next hidden state h_t. The current cell state c_t is passed through a tanh activation and multiplied with the output of the output gate. This decides the information to be carried in the next hidden state. h_t is output for the current step t and is also passed onto to next time step $t+1$.

Bidirectional Long Short Term Memory networks (BiLSTM) [22] is an extension of LSTM, with the capability of flowing information in both forward and reverse directions. This makes them more powerful in better understanding the context. These are created by stacking two LSTM layers, one processing information in the forward direction and the other in the reverse direction. The outputs for each of these two layers are combined through an activation layer and the final outputs are generated.

Chapter 3

Preprocessing document images for OCR

This chapter familiarizes the readers with some of the previous methods used for preprocessing document images for OCR. We discuss some of the conventional and deep learning methods used for image binarization in Section 3.1 and then highlight previous works that have employed GAN for enhancing degraded document images in Section 3.2. It is important to understand that there are many tasks involved in preprocessing document images for OCR based on the type of degradation. Some of these include background noise removal [16], [37], skew correction [78], [83], [93], watermark removal [85], [91], [110], super-resolution [49], [72], [79], and deblurring [11], [20], [39], [63]. In this work, we focus mainly on the degradation in the form of uneven contrast between background and text, dark spots or ink stains, and faded characters.

3.1 Document image binarization

Conventional methods for preprocessing document images with noise, such as uneven contrast between the background and text, dark spots or ink stains, and faded characters, utilize image binarization techniques. These techniques involve classifying each pixel as text or background based on a certain threshold value. The threshold can be decided based on global or local image features. Global thresholding methods apply the same threshold to every pixel in the image. Otsu's method [67] performs global thresholding, deciding the thresh-

old value from the grayscale histogram of an image to minimize the variance between background and foreground pixels. Local thresholding involves determining the threshold for each pixel using information from its local neighbors. Popular thresholding methods like Niblack [64] and Sauvola [84] adopt local thresholding to generate binarized images. Niblack's method uses the mean and standard deviation values of the local pixels within a certain window to calculate the threshold. Sauvola's method, an improvement over Niblack's, uses adaptive thresholding, adjusting the mean and standard deviation of local pixels within the window according to the contrast values. Some methods, such as [57], [66], utilize a hybrid approach for thresholding, combining global and local image features. The main drawback of these methods is their strong dependence on the choice of window size. This parameter needs to be carefully tuned for each image to obtain the optimal thresholding. Certain learning-based methods have also been proposed, which use hand-crafted features. Xiong *et al.* [109] utilize an SVM model for this task, performing binarization in three steps. First, the image is divided into regions based on the window size, and a local contrast adjustment is performed for each region. Then, a global threshold is selected to binarize each region using an SVM model. Finally, local adaptive thresholding is performed over the entire image. However, such learning-based methods often fail to generalize to all images.

With the growth of CNNs and their strong image feature extraction capabilities, several works have explored these networks in document image enhancement and binarization tasks. Pastor-Pellicer *et al.* [71] use a CNN to classify each image pixel as belonging to the background or foreground based on the intensity values of neighboring pixels within a window. Calvo-Zaragoza and Gallego [10] utilize a very deep Residual Encoder-Decoder Network (RedNet) [59] and propose a selectional auto-encoder (SAE) model that outputs a selectional value corresponding to each pixel based on whether it belongs to the background or foreground, using these values for thresholding to generate the binarized image. Tensmeyer and Martinez [96] propose a model using a fully convolutional network (FCN) trained with a combination of pseudo-F-measure

[65] and F-measure loss. Vo *et al.* [100] propose a hierarchical deep supervised network that predicts text pixels considering image features at several levels.

3.2 GAN for document image enhancement

With the success of GANs in image-to-image translation tasks [36], [50], [102], [113], several works have utilized GAN-based models for the enhancement of document images. DE-GAN [91] shows the effectiveness of a conditional GAN [36] for document binarization, deblurring, and watermark removal tasks using paired noisy and corresponding clean images. The generator is conditioned on the input noisy image. To ensure that the text in the original noisy image is preserved during the enhancement, a log loss is added between the generated clean image and the ground truth clean image. The architecture involves the use of a single generator and discriminator. Later, Ray *et al.* [77] propose a framework for document enhancement and recognition jointly using a GAN-based framework for image enhancement and a bidirectional LSTM and Connectionist Temporal Classification (CTC) based module for text recognition. The image enhancement model utilizes a fully convolutional RED-Net [59] for image denoising followed by a deep back projection network (DBPN) [25] for super-resolution. The CTC loss between the text recognition output and ground truth text provides the supervision for training the model. Souibgui *et al.* [92] along similar lines, proposes integrating a recognizer in the discriminator of a conditional GAN to guide the generator to produce clean images with readable text. Similarly, Kodym and Hradi [46] propose a text-guided transformer GAN that uses the target text transcription as a guiding signal for conditioning the restoration. Later, Poddar *et al.* [74] propose a GAN-based framework for text restoration from deformed handwritten documents. Among all these works, there is a constraint for the availability of either ground truth text or ground truth clean images for training. As discussed previously, this requirement limits the direct application of these models in a real-world setting.

In recent years, several unpaired techniques have also been explored for

enhancing document images using GANs. Sharma *et al.* [85] explore the feasibility of applying CycleGAN for document image cleaning. The main advantage of CycleGAN is that it does not require corresponding noisy/clean image pairs. However, in this particular work, the CycleGAN model is trained using paired samples. Neji *et al.* [63] propose Blur2Sharp CycleGAN for the task of text document deblurring by adjusting the parameters of CycleGAN for effective document deblurring in an unsupervised setting. However, other types of degradation are not explored. More recently, Gangeh *et al.* [19] proposed a unified single model approach for eliminating four different noise types (salt and pepper, faded, blurred, and watermarked) by integrating a Deep Mixture-of-Experts (MOE) [103] model with a CycleGAN model for cleaning document images without clean/noisy pairs. The results show the effectiveness of their proposed framework over training separate CycleGANs for each type of noise or training a CycleGAN sequentially, starting with one type of noise, followed by others. While the work can handle effectively different types of noise present in document images, without requiring ground truth text or image, it requires the metadata about the type of noise present in the image to train the embedder network of the MOE model. Therefore, along with noisy images, a label specifying the type of noise present in the image is required as input. Moreover, there is an assumption of the presence of only a single type of noise in each image. For real-world noisy images, such an assumption is not always valid. The noisy image can consist of a combination of noise types and it is difficult to label the type of noise.

These shortcomings are addressed by our proposed model. Following these works, we propose certain modifications in the standard CycleGAN model to improve its performance in document image enhancement tasks. Moreover, it is important to highlight that we achieve this objective without any supervision in the form of prior knowledge about the type of noise, availability of ground truth text, or clean ground truth images.

Chapter 4

Modified CycleGAN with CNN-BiLSTM discriminator

This chapter provides readers with a detailed description of our proposed model. We explain the generator network architecture in Section 4.1 followed by the discriminator network architecture in Section 4.2. Section 4.3 details the loss functions used for training the proposed model. Finally, in Section 4.4, we explain the overall training algorithm used.

4.1 Generator network

Figure 4.1 shows the overall architecture of the generator used in the proposed model. The architecture is adopted from the CycleGAN generator. The generator network consists of three parts - an encoder, residual blocks, and a decoder.

The encoder network maps the input image to a feature vector by performing downsampling. The input grayscale image of shape 256×256×1 is downsampled by a series of three convolutional layers. The first convolutional layer has a kernel of size 7×7 and stride 1. The other two convolutional layers have a kernel of size 3×3 with stride 2 for performing downsampling of input. All the three convolutional layers are followed by instance normalization and ReLU activation. The downsampled feature vector has a shape 64×64×256, where 256 is the number of channels.

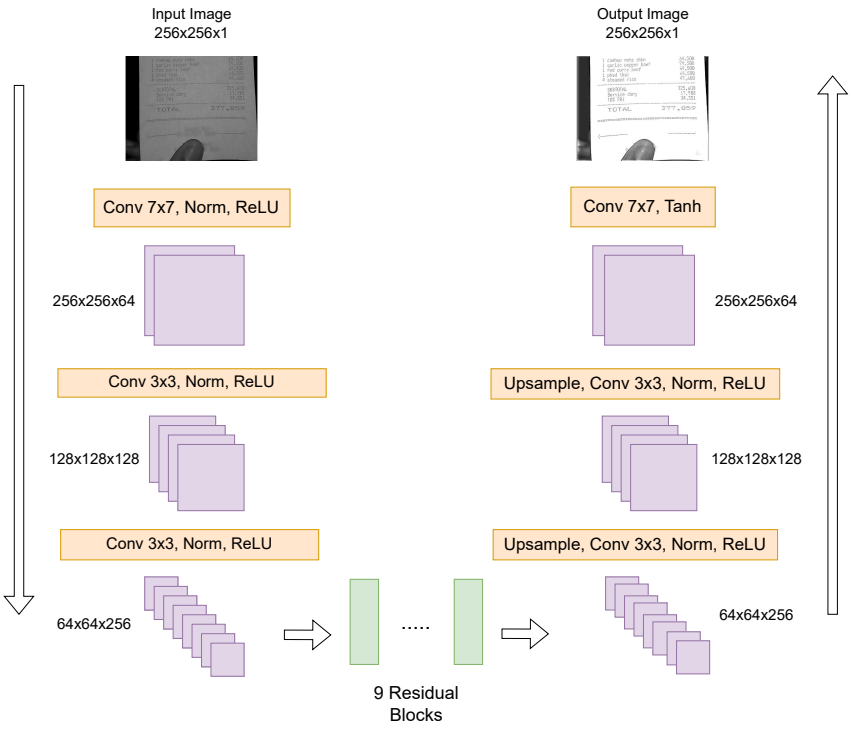


Figure 4.1: Generator network.

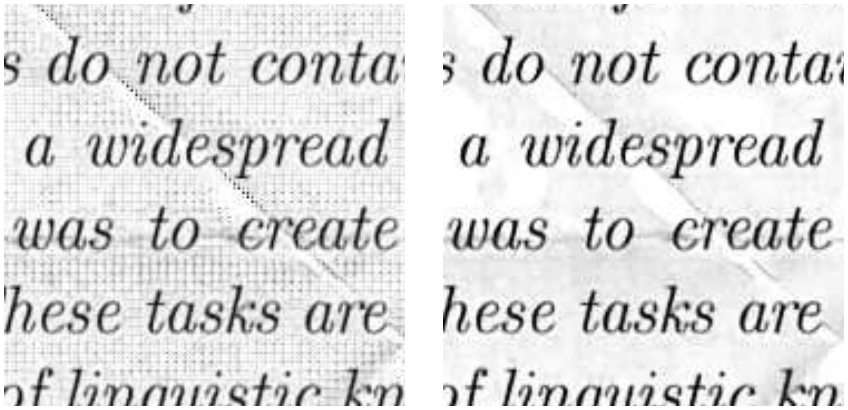
The downsampled feature vector is passed through a series of residual blocks. The main reason behind the use of residual blocks is the problem of exploding or vanishing gradients in deep convolutional neural networks resulting in the failure in convergence. Residual blocks overcome this by making use of skip connections that pass the output of a previous layer to another deeper layer. This connection provides a shortcut path through which gradients can pass. The residual blocks make use of residual function, where the output of the residual block $H(x)$ is given by:

$$H(x) = F(x) + x \tag{4.1}$$

Here, x is the output from a previous layer and $F(x)$ denotes the residual function. The residual block consists of a convolutional layer, a normalization layer, and a ReLU activation followed by another convolution layer and normalization layer. Table 4.1 shows the layers in each residual block used in the generator. Following the standard CycleGAN implementation, for 256×256 sized images, 9 residual blocks are stacked in the generator.

Type	Configuration	Activation Size
Input	-	256×256×1
Convolution	#maps: 64, k: 7×7, s: 1, p:3	256×256×64
Convolution	#maps: 128, k: 3×3, s: 2, p:1	128×128×128
Convolution	#maps: 256, k: 3×3, s: 2, p:1	64×64×256
9 Residual blocks	Each block : Convolution #maps: 256, k: 3×3, s: 2, p:1 Instance normalization+ReLU Convolution #maps: 256, k: 3×3, s: 2, p:1 Instance normalization	64×64×256
Upsample	scale=2	128×128×128
Convolution	#maps: 128, k: 3×3, s: 1, p:1	128×128×128
Upsample	scale=2	256×256×64
Convolution	#maps: 64, k: 3×3, s: 1, p:1	256×256×64
Convolution	#maps: 1, k: 7×7, s: 1, p:3	256×256×1

Table 4.1: Generator network summary. #maps, 'k', 's', and 'p' represent the number of channels, kernel size, stride, and padding respectively.



(a) (b)

Figure 4.2: Example of checkerboard pattern observed in the generated image when using transposed convolution and the generated image free of checkerboard pattern when using upsampling followed by convolution.

The output feature vector after the series of residual blocks is passed to the decoder network. The decoder network maps the feature vector to an output image by performing upsampling. The output image has the same size as the input image. The decoder network consists of two decoding blocks followed by a final convolution layer to output the generated image. The CycleGAN model uses fractionally-strided convolutional layers with stride $\frac{1}{2}$ as the decoding block. We observed the introduction of a checkerboard pattern

in the generated images when using this network. A possible reason for the appearance of these patterns is due to the transposed convolution operations causing uneven overlap when kernel size is not a multiple of the stride value [1]. A way to avoid these patterns is to separate upsampling operation from the convolutional operation. Following this, we modify the decoder blocks such that instead of using transposed convolutions, we use an upsampling operation followed by a convolutional operation. Here, we perform upsampling using the nearest-neighbor interpolation technique. Figure 4.2 shows the example of the checkerboard pattern observed in the generated image when using transposed convolutions and the generated image free of the checkerboard pattern after the modification. After the two decoding blocks, the final convolution layer uses a 7×7 kernel followed by a tanh activation. Table 4.1 summarizes the network configuration for the proposed generator.

4.2 Discriminator network

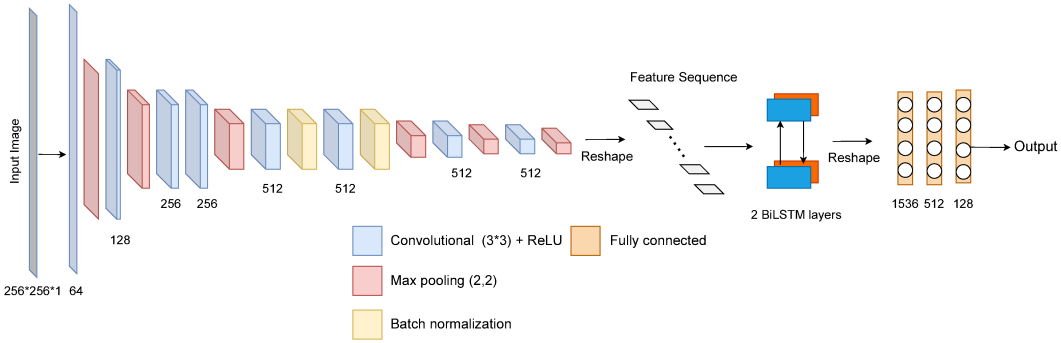


Figure 4.3: Proposed discriminator network.

For document image enhancement, to preserve the text while performing the translation, the discriminator needs to extract stronger features that capture text as well in addition to visual features while distinguishing between real or fake images.

In this work, we replace the CNN-based discriminator network in CycleGAN with a CNN-BiLSTM model. The CNN-BiLSTM model combines the

[1] https://distill.pub/2016/deconv-checkerboard/

Type	Configuration
Input	$256 \times 256 \times 1$
Convolution	#maps: 64, k: 3×3, s: 1, p:1
MaxPooling	Window: 2×2, s: 2
Convolution	#maps: 128, k: 3×3, s: 1, p:1
MaxPooling	Window: 2×2, s: 2
Convolution	#maps: 256, k: 3×3, s: 1, p:1
Convolution	#maps: 256, k: 3×3, s: 1, p:1
MaxPooling	Window: 2×2, s: 2
Convolution	#maps: 512, k: 3×3, s: 1, p:1
Batch Normalization	-
Convolution	#maps: 512, k: 3×3, s: 1, p:1
Batch Normalization	-
MaxPooling	Window: 2×2, s: 2
Convolution	#maps: 512, k: 3×3, s: 1, p:1
MaxPooling	Window: 2×2, s: 2
Convolution	#maps: 512, k: 3×3, s: 1, p:0
MaxPooling	Window: 2×2, s: 2
Map-to-Sequence	-
Bidirectional-LSTM	#hidden units: 256
Bidirectional-LSTM	#hidden units: 256
Reshape	-
Linear	1536, 512
Linear	512, 128
Linear	128, 1

Table 4.2: Discriminator network summary. #maps, 'k', 's', and 'p' represent the number of channels, kernel size, stride, and padding respectively.

advantages of CNN and BiLSTM networks. The combination has proven its success in extracting stronger features in text recognition models [27], [38], [40], [86], [87] which inspired us to utilize this combination in the discriminator network. CNN is used to extract image features from input document images. The extracted image features are flattened and passed as a one-dimensional vector to the BiLSTM network. The BiLSTM network utilizes sequential learning capabilities to generate enhanced features that better represent the text within the document images.

Figure 4.3 shows the architecture of the proposed discriminator network. The overall network consists of a CNN and a BiLSTM network followed by fully connected layers for classification. We adopt the CRNN network from [86]. The CNN network consists of convolutional and max-pooling layers to extract important local features from the input image. The original CRNN model takes in an input of size $W \times 32$, where W is the width of the image. Since in our case, we have 256×256 size input, we add two additional max-pooling layers and one convolutional layer. The $256 \times 256 \times 1$ input image is downsampled to $3 \times 3 \times 512$ feature map. This feature map is reshaped for input to the RNN network. The RNN network has two BiLSTM layers with 256 hidden units each. During preliminary experiments, we tried increasing the number of BiLSTM layers to 4. But the modification was not very useful and only increased the number of parameters for training. The output from the BiLSTM layers has shape 3×512. As we need to perform classification instead of text recognition, we remove the transcription layer (maps the output of BiLSTM to label sequence) and add three fully connected layers that finally output a single value. The network configuration for the proposed discriminator is summarized in Table 4.2.

4.3 Objective function

The overall objective function comprises two losses - $\mathcal{L}_{GAN}$ and the $\mathcal{L}_{cyc}$. $\mathcal{L}_{GAN}$ is calculated using Equations 2.10 and 2.11 whereas $\mathcal{L}_{cyc}$ is calculated using Equation 2.12.

Following Least Squares GAN (LSGAN) [60], least-squares loss is used to calculate $\mathcal{L}_{GAN}$. LSGAN helps with the problem of vanishing gradients and saturation of loss in GAN. It is implemented by changing the loss function of the discriminator to least squares loss instead of binary cross entropy loss. In a standard GAN, the discriminator acts as a binary classifier that classifies whether the generator outputs are generated or real and is trained using binary cross-entropy loss. Such binary signals do not provide informative feedback to the generator on how to improve itself.

Least Squares loss ensures that instead of providing binary feedback, the loss function provides feedback on how accurate or incorrect the predictions were. The loss is indicative of how close or far the generated images are with respect to the decision boundary. For generated data that is far from the decision boundary, the generator is penalized in proportion to the distance. This provides much more informative feedback to the generator to update itself.

With least squares loss, for a given GAN loss, $\mathcal{L}_{GAN}(G, D, X, Y)$, the generator G is trained to minimize $\mathbb{E}_{x \sim p_{\text{data}}(x)} \left[(D(G(x)) - 1)^2\right]$ and D is trained to minimize $\mathbb{E}_{y \sim p_{\text{data}}(y)} \left[(D(y) - 1)^2\right] + \mathbb{E}_{x \sim p_{\text{data}}(x)} \left[D(G(x))^2\right]$.

Cycle consistency loss is calculated using L1 loss. In addition to these losses, there is an optional identity loss that is used in CycleGANs. The identity loss ensures that $G(y)$ should be $\approx y$ and $F(x)$ should be $\approx x$. That means if an input already belongs to the target domain, the generator should perform an identity mapping ensuring no change. It is calculated as :

$$\begin{aligned} \mathcal{L}_{\text{identity}}\,(G, F) = \mathbb{E}_{y \sim p_{\text{data}}\,(y)} \left[\|G(y) - y\|_1\right] \\ + \mathbb{E}_{x \sim p_{\text{data}}\,(x)} \left[\|F(x) - x\|_1\right] \end{aligned} \tag{4.2}$$

In the CycleGAN paper, it is added with weight $\lambda_{id} \times \lambda_{cyc}$ making the overall objective function:

$$\begin{aligned}\mathcal{L}\left(G, F, D_X, D_Y\right) = & \mathcal{L}_{GAN}\left(G, D_Y, X, Y\right) \\ & + \mathcal{L}_{GAN}\left(F, D_X, Y, X\right) \\ & + \lambda_{cyc}\mathcal{L}_{cyc}(G, F) \\ & + \lambda_{id}\lambda_{cyc}\mathcal{L}_{\text{identity}}(G, F)\end{aligned} \tag{4.3}$$

The value of λ_{id} suggested by the CycleGAN paper is 0.5. We also add this loss while training our model, to ensure that if an input image is already clean, then the generator should not transform it into a different image but output it with no changes.

4.4 Training

Algorithm 1 shows the overall training procedure for the proposed model. Unpaired images a and b are randomly selected from the noisy domain X_{noisy} and clean domain X_{clean}, respectively. Fake clean image $b^{'}$ corresponding to a is generated using $G_1 : X_{noisy} \rightarrow X_{clean}$ and fake noisy image $a^{'}$ corresponding to b is generated using $G_2 : X_{clean} \rightarrow X_{noisy}$. Loss $\mathcal{L}_{G_1}$ and $\mathcal{L}_{G_2}$ are calculated. Total loss $\mathcal{L}_G$ is calculated by adding the individual losses for each of the generators. The parameters for generators G_1 and G_2 are updated by computing the gradient of $\mathcal{L}_G$. Loss $\mathcal{L}_{D_1}$ for D_1 and $\mathcal{L}_{D_2}$ for D_2 are calculated. The parameters of D_1 are updated with respect to the gradient of $\mathcal{L}_{D_1}$ and the parameters of D_2 are updated with respect to the gradient of $\mathcal{L}_{D_2}$.

Algorithm 1 Training steps for proposed model

Input: X_{noisy}, X_{clean}

Initialize: Generator $G_1(\phi_{g_1}) : X_{noisy} \rightarrow X_{clean}$, Generator $G_2(\phi_{g_2}) : X_{clean} \rightarrow X_{noisy}$, Discriminator $D_1(\theta_{d_1})$, Discriminator $D_2(\theta_{d_2})$, cycle consistency λ_{cyc}, identity mapping λ_{id}, number of epochs n

for n **do**

  for $a, b \in \{X_{noisy}, X_{clean}\}$ **do**

    $b^{'}, a^{'} = G_1(a), G_2(b)$

    $\mathcal{L}_{G_1} = \text{MSE}(D_2(b^{'}), \text{target}_{real}) + \lambda_{cyc}\text{L1}(G_2(b^{'}), a) + \lambda_{id}\lambda_{cyc}\ \text{L1}(G_1(b), b)$

    $\mathcal{L}_{G_2} = \text{MSE}(D_1(a^{'}), \text{target}_{real}) + \lambda_{cyc}\text{L1}(G_1(a^{'}), b) + \lambda_{id}\lambda_{cyc}\ \text{L1}(G_2(a), a)$

    $\mathcal{L}_G = \mathcal{L}_{G1} + \mathcal{L}_{G2}$

    Update ϕ_{g_1} using $\nabla_{\phi_{g_1}}(\mathcal{L}_G)$

    Update ϕ_{g_2} using $\nabla_{\phi_{g_2}}(\mathcal{L}_G)$

    $\mathcal{L}_{D_2} = \text{MSE}(D_2(b), \text{target}_{real}) + \text{MSE}\ (D_2(b^{'}), \text{target}_{fake})$

    $\mathcal{L}_{D_1} = \text{MSE}(D_1(a), \text{target}_{real}) + \text{MSE}\ (D_1(a^{'}), \text{target}_{fake})$

    Update θ_{d_1} using $\nabla_{\theta_{d_1}}(\mathcal{L}_{D_1})$

    Update θ_{d_2} using $\nabla_{\theta_{d_2}}(\mathcal{L}_{D_2})$

  end for

end for

Chapter 5

Creating unpaired noisy/clean training dataset

This chapter provides details about the steps used to prepare the training, validation, and test sets for the proposed model. The task of document image enhancement in this work is formulated as an unpaired image-to-image translation task, where we translate noisy images to clean images. Thus, we need a set of noisy document images and an unpaired set of clean document images. In Section 5.1, we discuss the training set that was used to train the proposed model. Firstly, we elaborate on the different datasets of noisy document images that were utilized to form the noisy domain. This is followed by details on the creation of unpaired clean documents to form the clean domain. In Section 5.2, we provide details on the data used for evaluation.

5.1 Training data

Three document image datasets - the Kaggle Denoising Dirty Documents dataset [14], the Point-of-Sale (POS) Receipts dataset [76], and the Noisy OCR Dataset (NOD) [28] are used.

The Kaggle Denoising dataset consists of noisy document images with various synthetically added noise types, such as wrinkles, stains, and faded spots. The dataset also includes a variety of text font styles. The POS dataset is a combined dataset formed from real-world noisy receipt images from three datasets: the ICDAR SROIE competition dataset [35], the Findit fraud detec-

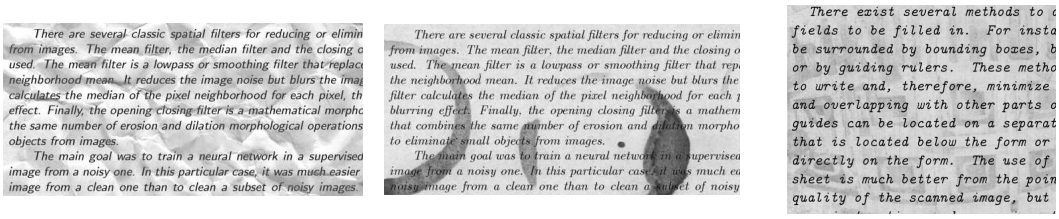


Figure 5.1: Sample noisy images from Kaggle Denoising dataset.

Figure 5.2: Sample noisy images from POS dataset.

tion dataset [2], and the CORD dataset [69]. The images in the POS dataset are extracted patches of size 500 × 400, horizontally cropped from the full-size receipts. The Noisy OCR dataset comprises pages from old English and Arabic books with different synthetically added noise types. We select images with English text and noise in the form of weak ink. Sample noisy images from these datasets are shown in Figures 5.1, 5.2, and 5.3.

For image-to-image translation, a set of images belonging to the target domain is also required for training. Since we train our model in an unsupervised setting, the target clean set should consist of unpaired clean document images. In the case of the Kaggle Denoising dataset and Noisy OCR dataset, we create this clean set from electronic research papers/books in PDF format. We extract pages from the PDFs as images. However, for the POS dataset, our preliminary experiments using pages from research paper PDFs resulted in unstable results due to the different nature of the images in the two domains. To ensure that the unpaired target domain images are not completely different from the noisy receipt images, we prepared our clean unpaired set by generating fake clean receipt images. These images contain randomly generated text with different font styles and sizes on a white background. Figure 5.4 shows samples of images from the clean set.

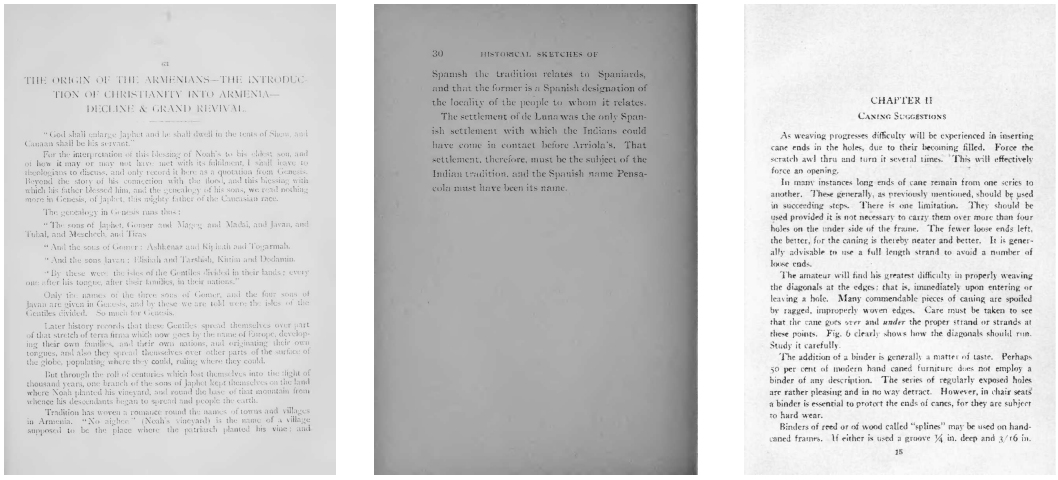


Figure 5.3: Sample noisy images from Noisy OCR dataset.

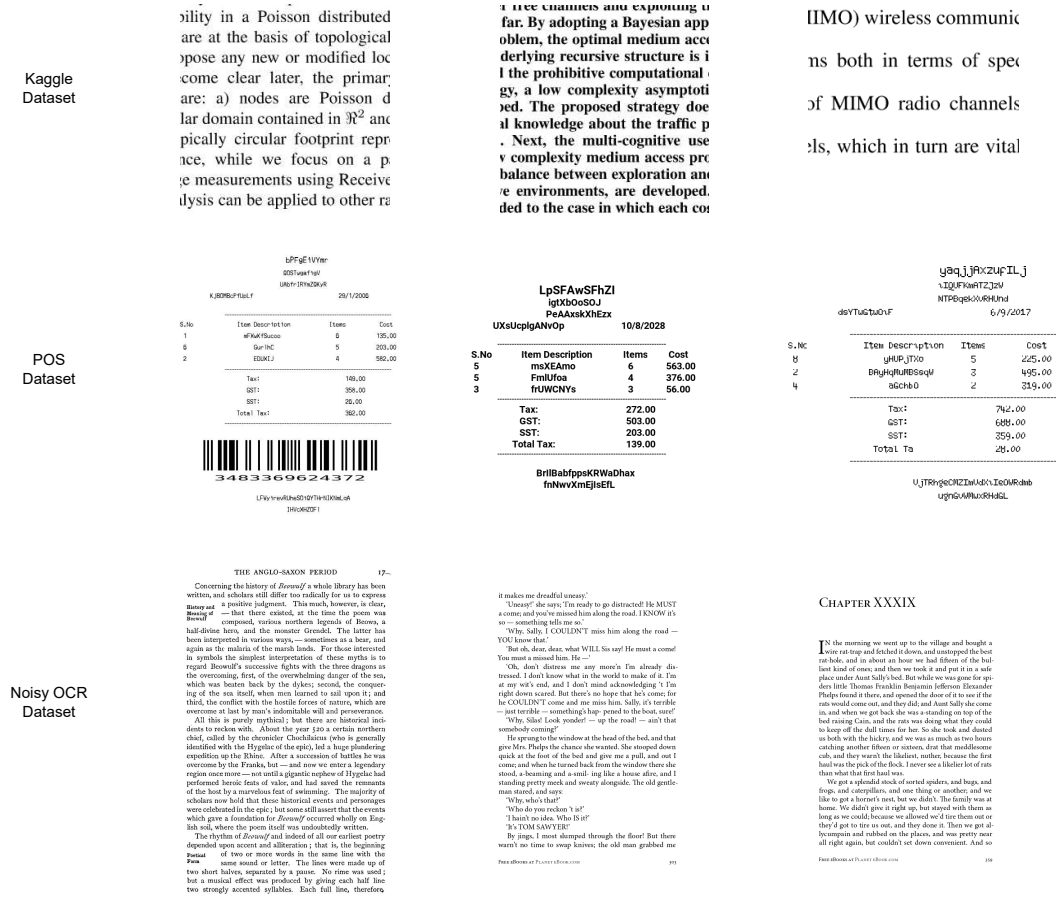


Figure 5.4: Sample clean images used for training.

Figure 5.5: Sample noisy test images from WildReceipt dataset.

The input to the generator is of size 256 × 256. Therefore, non-overlapping patches of size 256 × 256 are extracted from the noisy and clean images to form the training data. The images between the two domains, noisy and clean, are completely unpaired with no overlap in terms of text. The proposed model is trained on each of the datasets separately.

5.2 Evaluation data

The trained model is evaluated on noisy images in the validation and test set from the datasets. The validation set is used for selecting the best-performing model and for hyperparameter tuning. For the Kaggle Denoising dataset and Noisy OCR dataset, the validation set is created by randomly splitting 10% of the original dataset provided. For the POS dataset, the validation set is already available separately. Each of these datasets already has a separate test set available for inference.

During the evaluation, we input full-scale images into the trained generator to generate corresponding clean images. This allows us to perform OCR evaluation on the generated images. Therefore, the images in the validation set and test set are full-sized and not patches.

Besides these datasets, we also evaluate the generator trained with the POS receipts dataset on unseen noisy test images from another more challenging

Dataset	Training set	Test set	
	Number of noisy/clean patches	Number of full-size images	Number of words
Kaggle Denoising	288	60	5392
POS	3676	417	8366
WildReceipt	-	472	12707
Noisy OCR	2137	65	18805

Table 5.1: Dataset Summary: Number of noisy/clean image patches in the training set and the number of images and words in the test set.

and complex receipts OCR dataset - the WildReceipt dataset [94] and report the performance. Sample images from the WildReceipt dataset are shown in Figure 5.5.

Table 5.1 provides a summary of the number of noisy/clean image patches in the training set and the number of images in the test set for each dataset.

Chapter 6

Experiments and Results

This chapter provides details on the experiments performed in this work and presents the results. In Section 6.1, we provide information about the OCR engines used for evaluation, the evaluation metrics and hyperparameters, and the training setup. In Section 6.2, we present the qualitative and quantitative results that shows the performance of the proposed model compared to other baselines. In Section 6.3, we discuss some additional preliminary experiments performed during the course of this work.

6.1 Experimental setup

6.1.1 OCR engines

Three open-source OCR engines are used for evaluation - Tesseract [1], EasyOCR [2] and PaddleOCR [3]. Tesseract is a popular open-source OCR engine that uses an LSTM network for text recognition. We use the Tesseract 4.0.0 version. EasyOCR is another open-source OCR engine that uses the CRAFT model [3] as the text detection module. The recognizer module is CRNN based and consists of feature extraction using ResNet [26], followed by sequence labeling using BiLSTM networks and transcription using CTC loss. PaddleOCR is a very lightweight and comparatively newer OCR. We use the latest PP-OCRv3 version. The recognition module uses a Scene Text Recognition with a Single

[1] https://github.com/tesseract-ocr/tesseract
[2] https://github.com/JaidedAI/EasyOCR
[3] https://github.com/PaddlePaddle/PaddleOCR

Visual Model (SVTR) [13] instead of a CRNN.

6.1.2 Evaluation metrics

OCR based metrics

The primary objective of this work is to enhance noisy images in a way that improves OCR performance. To evaluate the performance of the proposed model in achieving this objective, we perform OCR on both the original noisy images and the generated images by the trained models. The OCR outputs are then compared with the ground truth text. It is important to note that the ground truth text is used solely for evaluation purposes and is not used at any point during training.

For the POS receipt and WildReceipt datasets, the ground truth text includes bounding boxes with word-level text, which are directly used for evaluation. In the case of the Kaggle Denoising dataset, manual annotation of the ground truth text was performed as the dataset did not include ground truth text files. As for the Noisy OCR dataset, document-level ground truth text is available instead of word-level text.

For POS, WildReceipt, and Kaggle Denoising datasets, we evaluate the OCR performance in terms of word accuracy and Levenshtein distance [51] based Character Error Rate (CER). Word accuracy is measured as the ratio of the words matched with the ground truth to the total number of words in the ground truth. CER is defined as:

$$CER = 100 \times (i + s + d)/m \tag{6.1}$$

where i, s, and d are the number of insertion, substitution, and deletion operations performed to match the predicted word to the ground truth. m represents the number of characters in the ground truth. Better performance is indicated by higher values of word accuracy and lower values of CER.

For the Noisy OCR dataset, in the absence of word-level ground truth bounding boxes, we evaluate the OCR performance using the Levenshtein distance over the entire document text. Lower values for Levenshtein distance indicate better performance.

No-reference image quality metrics

While the direct performance of the proposed model in effectively enhancing document images for OCR can be measured using the OCR-based metrics described above, we also perform an evaluation using image quality metrics.

In the absence of ground truth images, full-reference metrics based on Peak-Signal-to-Noise Ratio (PSNR) and SSIM [104] cannot be used. Hence, we utilize two no-reference image quality metrics - Natural Image Quality Evaluator (NIQE) [62] and Perceptual index (PI) [8]. These metrics have been used in several GAN-based works [30], [33], [68], [75] to evaluate the quality of generated images in the absence of ground truth clean images. NIQE measures the deviations in statistical properties of natural images due to distortions, while PI is a combination of the metrics proposed by Ma *et al*. [58] and NIQE. Ma *et al*. [58] introduced a no-reference metric for assessing the quality of super-resolved images using a regression model to predict scores based on designed statistical features. PI is calculated as $\frac{1}{2}((10 - \text{Ma}) + \text{NIQE})$. It is commonly used to assess the no-reference image quality of super-resolution images [12], [32], [116].

It is important to note that these metrics are based on perceptual image quality and may not serve as clear indicators of enhanced images for OCR. The results obtained from these metrics are provided for reference, while the primary evaluation remains based on the OCR-based metrics.

6.1.3 Hyperparameter and Training details

The proposed model is trained for 40, 100, and 50 epochs for the Kaggle Denoising, POS, and Noisy OCR datasets respectively. We perform a search over the hyperparameters of the model for selecting the best values. More details on hyperparameters are included in Appendix A. Based on the tuning results, we select $\lambda_{cyc} = 10$ and identity mapping loss coefficient $\lambda_{id} = 0.5$, batch size equal to 1, and Adam optimizer with learning rate equal to 2×10^{-4}. The proposed model is implemented using the Pytorch framework. It takes approximately 1.5 hours, 42 hours, and 12 hours to train on Kaggle Denoising,

POS, and Noisy OCR dataset, respectively using Nvidia Tesla V100 GPU. For comparison with the standard CycleGAN model, we use the standard Pytorch implementation as provided by [85], [121]. The same hyperparameter values as the proposed model are used for training CycleGAN.

6.2 Results and Discussion

The performance of the proposed model is directly compared with standard CycleGAN to evaluate the impact of the changes proposed in this work. Additionally, the proposed model performance is also compared with two other classical unsupervised image preprocessing techniques - Otsu [67] and Sauvola [84] discussed in Section 3. As the performance of these techniques is heavily dependent on the window size, different window sizes are used and the best results are reported. For both CycleGAN and the proposed model, the results presented here, reflect the test performance of the best-performing models selected based on word accuracy on the validation set using Tesseract OCR.

6.2.1 Qualitative results

We illustrate some of the output images generated by the proposed model and other mentioned baseline preprocessing models. Figure 6.1 shows the enhanced images generated for a sample noisy image in the Noisy OCR dataset and the predicted text using Tesseract OCR. The noisy image has degradations in the form of broken characters in the text. Comparing the various generated images, it can be observed that the image generated using standard CycleGAN appears cleaner with darker text. However, upon closer look, it is apparent that some of the characters are hampered during enhancement. This is undesirable and results in incorrect predictions by the OCR for certain words that were originally identified correctly in the noisy image. For example, the word “sumptuary” and “arms” were predicted correctly in the original image but now mispredicted as “soniptuary” and “arpis”. This deterioration negatively impacts the overall OCR performance. On the contrary, the enhanced image generated by the proposed method avoids further degradation in OCR

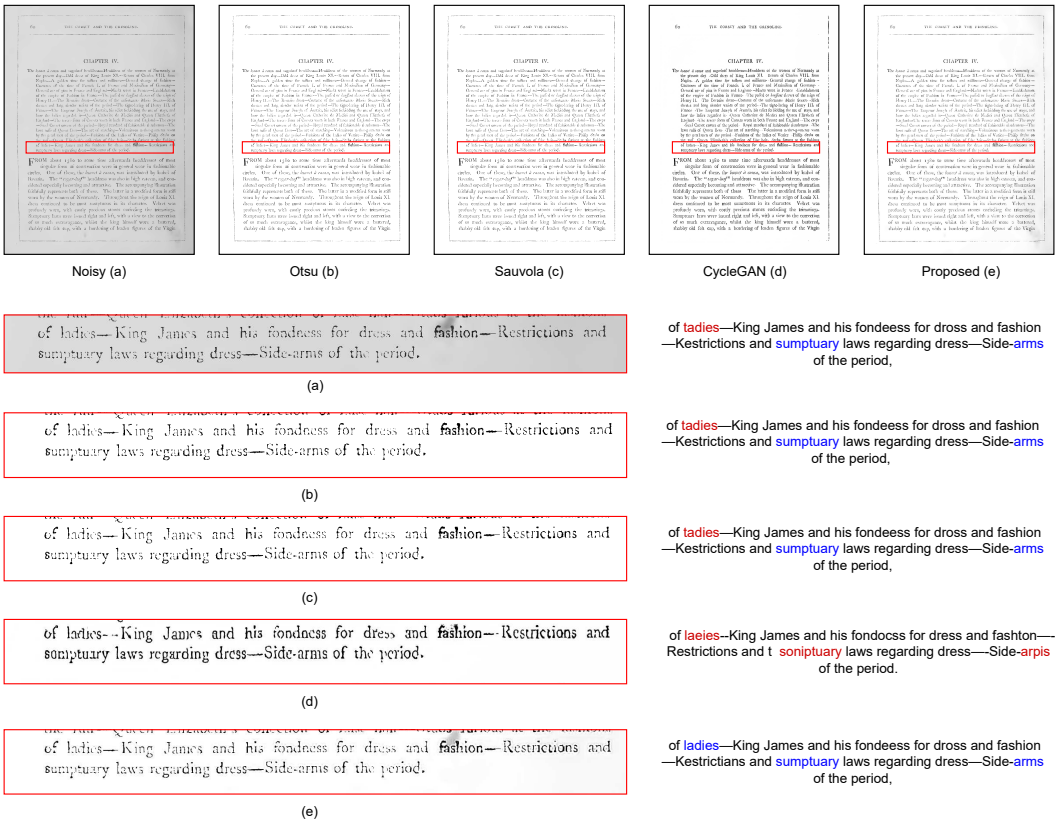


Figure 6.1: Example from the Noisy OCR dataset showing the noisy image and the generated enhanced images along with the text predictions using Tesseract OCR.

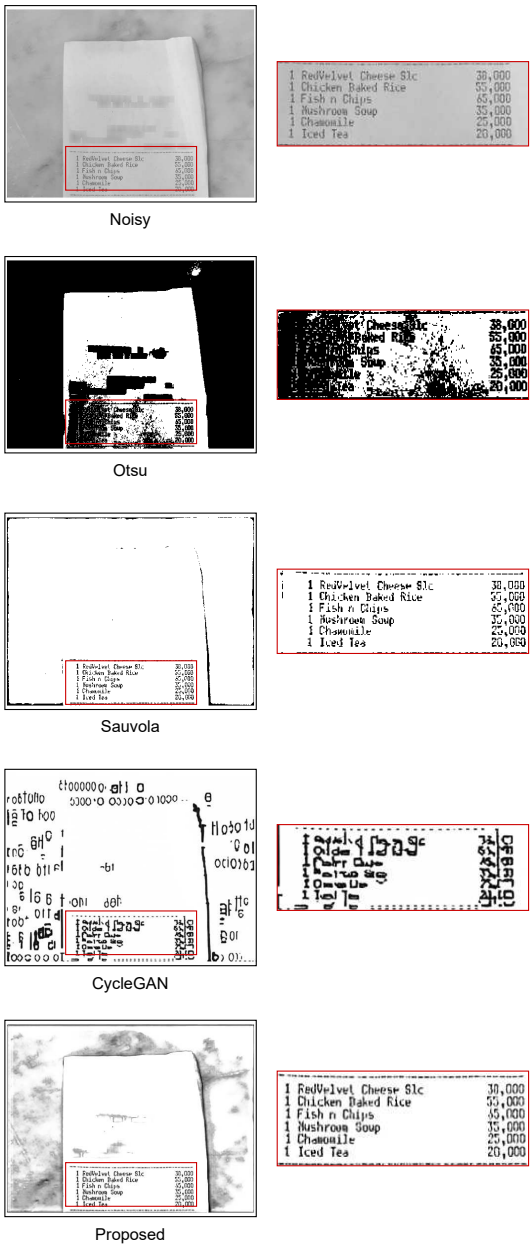


Figure 6.2: Example from the POS dataset showing the noisy image and the generated enhanced images. Zoomed-in images show the appearance of text in the images.

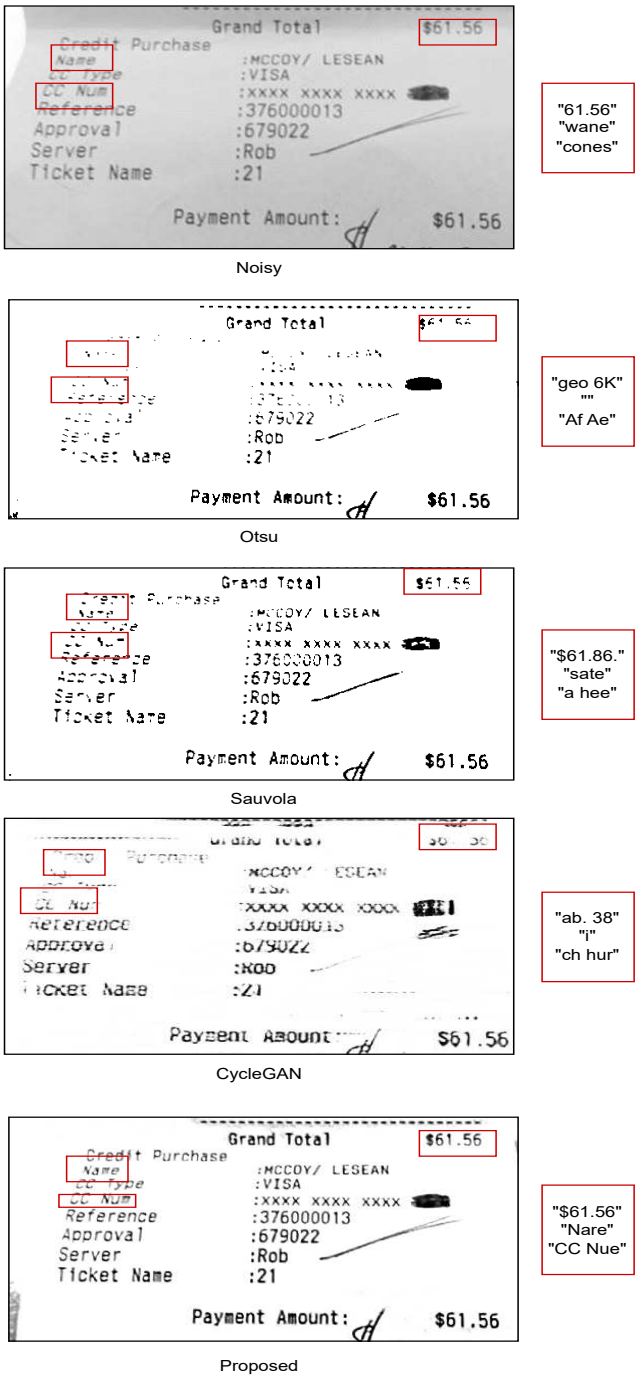


Figure 6.3: Example from the Wildreceipt dataset showing the noisy image and the generated enhanced images along with the text predictions using Tesseract OCR.

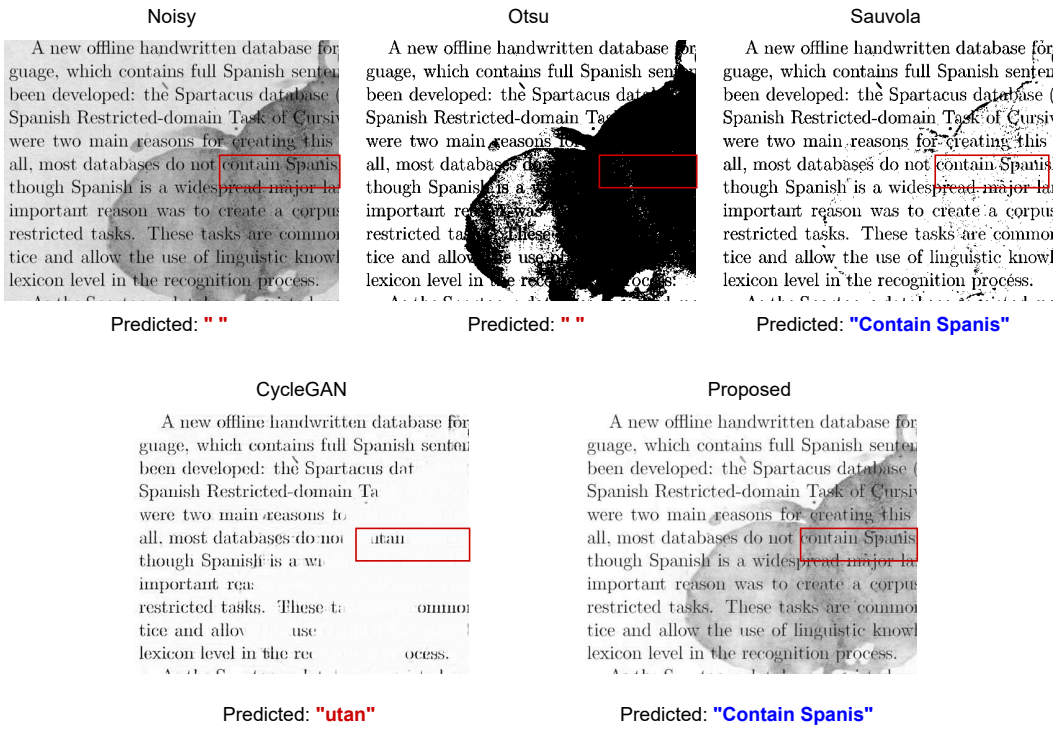


Figure 6.4: Example from the Kaggle Denoising dataset showing the noisy image and the generated enhanced images along with the text predictions using Tesseract OCR.

performance compared to the original noisy image.

Considering receipt images from POS and WildReceipt dataset, Figures 6.2 and 6.3 highlight that the proposed model has better preservation of characters in the enhanced images over all the other methods. The proposed model leads to improved OCR performance with more correctly predicted words compared to the noisy images. For the Kaggle dataset, we illustrate an example of a noisy image that contains degradation in the form of a dark patch in Figure 6.4. Clearly, Otsu and CycleGAN models cause a loss of text underneath the dark patch. As a consequence, OCR engines are unable to predict those words, causing a decline in performance, despite the visually cleaner appearance of the images. On the other hand, the proposed model and Sauvola's method can preserve the text well. Sauvola's method, being a binarization technique, aims to create a binary image by eliminating the dark patch. In contrast, the proposed model attempts to minimize the intensity of the dark patch while keeping the grayscale format of the generated image.

6.2.2 Quantitative results

In addition to the qualitative results demonstrating the effectiveness of the proposed model, we also provide quantitative results using three OCR engines for each dataset. Table 6.1 provides an overview of the OCR engine performance on both the noisy test images and the enhanced images from the Kaggle Denoising, POS, and Wildreceipt datasets.

For the Kaggle Denoising dataset, the proposed model generates clean images that increase Tesseract OCR performance in terms of both word accuracy (81.23%) and CER (18.72). However, for EasyOCR and PaddleOCR engines, enhanced images have degraded OCR performance when compared to the original noisy image. Degradation in performance of EasyOCR and PaddleOCR is observed even if the same preprocessing improved the performance of Tesseract OCR engine. A possible reason for this is that preprocessing techniques can have different effects on OCR engines depending on their underlying algorithms and architecture. EasyOCR and PaddleOCR are deep learning-based OCR engines that utilize neural networks to extract text from images. These

		Tesseract		EasyOCR		PaddleOCR	
		word accuracy% ↑	CER ↓	word accuracy% ↑	CER ↓	word accuracy% ↑	CER ↓
Kaggle	Noisy	78.18	21.62	**73.31**	**32.12**	**82.09**	**14.83**
	Otsu [67]	76.60	22.59	61.03	40.12	69.50	25.01
	Sauvola [84]	79.45	21.00	66.04	35.71	67.90	31.76
	CycleGAN [121]	43.44	58.86	33.11	67.42	41.93	58.20
	Proposed	**81.23**	**18.72**	68.70	34.84	78.28	18.24
POS	Noisy	49.58	31.13	41.63	37.01	45.21	29.70
	Otsu [67]	46.50	36.89	34.86	43.43	34.96	42.54
	Sauvola [84]	55.30	26.31	42.12	33.83	41.07	34.45
	CycleGAN [121]	37.82	41.33	27.24	48.03	27.52	47.53
	Proposed	**61.14**	**23.18**	**46.41**	**31.59**	**50.49**	**25.75**
Wildreceipt	Noisy	29.09	46.02	23.74	49.43	30.08	43.05
	Otsu [67]	33.21	41.63	19.62	51.90	27.47	49.31
	Sauvola [84]	37.20	33.81	23.66	43.08	30.87	42.33
	CycleGAN [121]	09.02	74.02	08.47	76.66	12.07	72.30
	Proposed	**41.86**	**32.48**	**26.40**	**41.54**	**34.84**	**36.47**

Table 6.1: OCR performance in terms of word accuracy and CER on the original noisy images in the test set and the generated enhanced images compared with the ground truth text. Better performance is indicated by higher values of word accuracy and lower values of CER.

	Tesseract	EasyOCR	PaddleOCR
	Levenshtein distance ↓	Levenshtein distance ↓	Levenshtein distance ↓
Noisy	65.02	**178.03**	127.79
Otsu [67]	62.19	192.39	161.20
Sauvola [84]	64.85	192.08	159.54
CycleGAN [121]	88.82	287.23	168.19
Proposed	**62.05**	181.68	**108.63**

Table 6.2: OCR performance in terms of Levenshtein distance on the original noisy images in the test set and the generated enhanced images compared with the ground truth text for the Noisy OCR dataset. Better performance is indicated by lower values of Levenshtein distance.

engines are trained on a wide variety of images, including both clean and degraded ones. In some cases, document image enhancement techniques may alter the characteristics of the input image in a way that is not beneficial for these OCR engines. It can be possible that the transformations performed by our proposed model for image enhancement results in images that affect the performance of the underlying neural network models used by EasyOCR and PaddleOCR and hence slight degradation in performance is observed.

This degradation in performance is not unique to the proposed model but is observed across all the enhancement methods. Importantly, OCR performance across both the metrics is better for the images generated by the proposed model than images generated by other enhancement methods.

For the POS dataset, the proposed model improves the performance of all the three OCR engines. Maximum improvement is observed for the Tesseract OCR engine with an increase from 49.58% to 61.14% in terms of word accuracy and a reduction from 31.13 to 23.18 in terms of CER. Moreover, the proposed model consistently outperforms all the other methods across all the OCR engines.

Additionally, as mentioned previously, we also evaluate the performance of the proposed model originally trained on the POS dataset on noisy images from the WildReceipt dataset. Here, for comparison, the CycleGAN model is trained on the POS dataset as well and evaluated for the WildReceipt dataset. As can be observed, the proposed model consistently leads to improvement in performance for all OCR engines and also outperforms the baseline methods.

For the Noisy OCR dataset, Levenshtein distance is calculated between OCR output and the ground truth text for noisy and enhanced images. From Table 6.2, it can be observed that images enhanced by the proposed model have improved performance on Tesseract and PaddleOCR engines compared to the original noisy images. The performance is better than all the other baselines. However, for the EasyOCR engine, degradation in OCR performance is observed for enhanced images generated using any of the enhancement methods, including the proposed model. However, when considering the extent of degradation, the proposed model demonstrates comparatively less deteriora-

tion compared to other baseline methods.

From the results, it can be concluded that the proposed model enhances the performance of the Tesseract OCR engine consistently across all the datasets. These results highlight that the proposed model has better text preservation capabilities during translation compared to the standard CycleGAN. This implies the effectiveness of the proposed modifications.

We report the performance of the generated images on the no-reference image quality metrics NIQE and PI in Table 6.3. Lower values on these metrics indicate better image quality. Across all datasets, the images generated by the proposed model exhibit superior scores on both metrics compared to other enhancement methods. While an improvement over the original noisy images is reported across Kaggle Denoising, POS, and Noisy OCR datasets, a degradation in quality is observed for the WildReceipt dataset. This is inconsistent with the OCR-based metrics which indicate significant improvement in the performance of all the three OCR engines for the images generated by the proposed model. Further, Otsu's and Sauvola's methods achieve better performance over the standard CycleGAN model on OCR-based metrics but have poor performance on these image quality metrics. These observations suggest that for enhancing document images for OCR, these perceptual image quality metrics might not be suitable indicators of performance.

6.2.3 Ablation study

We perform experiments on the Kaggle Denoising dataset to analyze the impact of the proposed architectural modifications to the standard CycleGAN on the overall OCR performance improvement.

Effect of BiLSTM layers in the discriminator

First, we remove the BiLSTM layers from the proposed discriminator (Figure 4.3). The output of the last convolutional layer $3 \times 3 \times 512$ is reshaped to 9×512, where 512 represents the number of channels. Another linear layer with input size 9×512 and output size 3×512 is added to make the shape compatible with the following set of linear layers in the originally proposed model.

		NIQE ↓	PI ↓
Kaggle	Noisy	22.04	7.86
	Otsu [67]	21.85	18.73
	Sauvola [84]	21.57	19.24
	CycleGAN [121]	10.29	7.04
	Proposed	**9.37**	**6.10**
POS	Noisy	23.80	7.53
	Otsu [67]	26.65	17.16
	Sauvola [84]	22.75	16.55
	CycleGAN [121]	14.35	8.67
	Proposed	**8.65**	**5.92**
WildReceipt	Noisy	**6.07**	**6.83**
	Otsu [67]	23.96	15.70
	Sauvola [84]	23.72	15.59
	CycleGAN [121]	13.75	8.70
	Proposed	11.71	7.19
Noisy OCR	Noisy	13.98	9.12
	Otsu [67]	22.69	16.34
	Sauvola [84]	22.62	16.35
	CycleGAN [121]	11.71	7.67
	Proposed	**8.99**	**6.43**

Table 6.3: NIQE and PI metric values on the original noisy images in the test set and the generated enhanced images. Better performance is indicated by lower values for both metrics.

	$Model_1$		$Model_2$		Proposed Model	
	word accuracy% ↑	CER ↓	word accuracy% ↑	CER ↓	word accuracy% ↑	CER ↓
Tesseract	56.08	40.68	74.39	26.45	**81.23**	**18.72**
EasyOCR	40.72	59.54	64.37	39.23	**68.70**	**34.84**
PaddleOCR	48.13	48.23	66.47	33.67	**78.28**	**18.24**

Table 6.4: Performance comparison between $Model_1$/$Model_2$ and the proposed model for the Kaggle Denoising dataset. $Model_1$ has the same generator as the proposed model but the discriminator without the LSTM component. $Model_2$ has the same discriminator network as the proposed model but the generator network without the proposed changes in the decoder block.

The rest of the model is kept the same. We call this Model_1. This model is trained with the same set of hyperparameters as the proposed model and the OCR performance on the generated enhanced images is evaluated. From Table 6.4, a huge gap in the performance between Model_1 and the proposed model can be observed. OCR performance on images generated by Model_1 has significantly lower word accuracy and higher CER as compared to the proposed model. These results further support the hypothesis that a combined CNN-BiLSTM model in the discriminator network can extract stronger features from document images and act as a more suitable discriminator than pure CNN-based discriminators for document image enhancement tasks.

Effect of decoder block modifications in the generator

Next, we analyze the impact of the proposed changes in the Generator architecture. As mentioned earlier, to reduce the checkerboard artifacts arising in the generated images, the decoder blocks in the generator network were modified to have an upsampling operation followed by a convolutional operation instead of transposed convolutions. We revert this modification in the proposed model and keep the rest of the model unchanged. We call this Model_2. This model is trained with the same set of hyperparameters as the proposed model and the OCR performance on the generated enhanced images is evaluated. Results from Table 6.4 indicate that the modifications performed to remove the checkerboard artifacts in the proposed model lead to better performance. However, it is important to highlight that the effect of the generator modifications in improving the performance is much lesser compared to the effect of the proposed discriminator modifications. This explains that much of the performance improvement is attributed to the proposed discriminator. This further supports the effectiveness of the proposed discriminator.

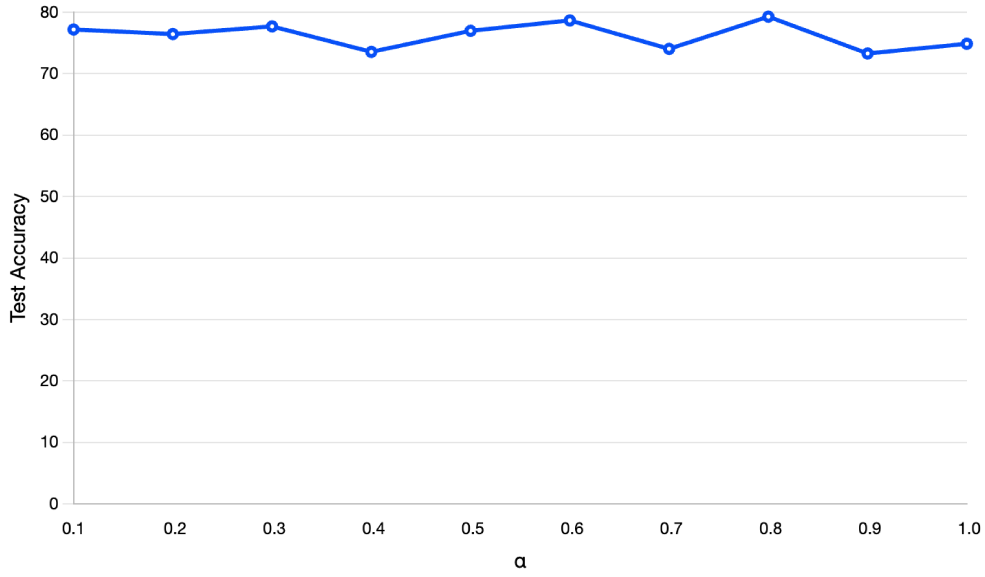


Figure 6.5: α and the corresponding word accuracy of Tesseract OCR on the images in the test set from the Kaggle Denoising dataset.

6.3 Additional experiments

6.3.1 Cycle consistency using combination of L1 and SSIM loss

In the proposed model and the standard CycleGAN model, the cycle consistency loss is calculated using L1 loss. L1 loss calculates the difference in pixel value between corresponding pixels in the real image and the reconstructed image. [89] suggests considering perceptual image quality metrics such as Structural Similarity (SSIM) index loss [104] can generate better quality images. Exploring this idea, we perform experiments with the modification of $\mathcal{L}_{cyc}$ (2.12) to a combination of L1 loss and Structural Similarity (SSIM) index loss instead of only the L1 loss. The structural similarity (SSIM) index compares the similarity between two images based on the luminance, contrast, and structural similarity information. It is calculated as:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{6.2}$$

			Tesseract		EasyOCR		PaddleOCR	
		α	word accuracy% ↑	CER ↓	word accuracy% ↑	CER ↓	word accuracy% ↑	CER ↓
Kaggle	Proposed w/o SSIM	0	**81.23**	**18.72**	**68.70**	34.84	78.28	**18.24**
	Proposed w SSIM	0.8	79.28	20.72	69.55	**33.61**	**78.52**	18.31
POS	Proposed w/o SSIM	0	61.14	23.18	46.41	31.59	50.49	25.75
	Proposed w SSIM	0.5	**65.98**	**17.60**	**48.72**	**28.95**	**51.72**	**24.70**
			Levenshtein		Levenshtein		Levenshtein	
Noisy OCR	Proposed w/o SSIM	0	**62.05**		**181.68**		**108.63**	
	Proposed w SSIM	1.0	62.27		209.62		135.12	

Table 6.5: Performance comparison between proposed model trained with and without SSIM component in $\mathcal{L}_{cyc}$.

where μ_x, μ_y is the mean of x and y, σ_x^2 and σ_y^2 is the variance of x, y and σ_{xy} is the covariance of x and y and c_1, c_2, c_3 are the small constants.

The SSIM loss is given by:

$$\mathcal{L}_{SSIM} = 1 - \text{SSIM}(x, y) \tag{6.3}$$

Combining SSIM loss with L1 loss, the cycle consistency loss $\mathcal{L}_{cyc}$ is calculated as:

$$\mathcal{L}_{SSIM+L1} = \alpha\mathcal{L}_{SSIM} + (1 - \alpha)\mathcal{L}_{L1} \tag{6.4}$$

Here, α controls the weight for each of the loss components.

We experiment with values of α in the range $[0.1, 1]$. Table 6.5 shows the best α values for each of the three training datasets and the corresponding OCR evaluation performance. While better performance compared to the proposed model without the SSIM component is achieved on the POS dataset, with $\alpha = 0.5$, the same is not observed for the other two datasets. Furthermore, huge inconsistencies were observed in the results for different values of α in the defined range.

Figure 6.5 shows the huge inconsistencies in performance for different values of α on the test set images in the Kaggle Denoising dataset. Given the lack of clarity regarding the usefulness of this loss component, we made the decision to exclude it from the final model proposed in this thesis.

Chapter 7

Conclusion

In this thesis, we focused on document image enhancement as an unsupervised image-to-image translation task. We present a modified architecture for the standard CycleGAN model that can significantly improve its performance in document enhancement tasks. Results illustrate that the combined discriminator network, which utilizes a combination of CNN and BiLSTM, achieves a significant enhancement in both text preservation and OCR performance when compared to the standard CycleGAN discriminator network.

Specifically, when evaluating the word accuracy of the Tesseract engine on real-world noisy receipt images from the POS dataset, the proposed model showed an improvement of up to 61.66% over the original CycleGAN model. This significant increase in accuracy confirms our hypothesis that for tasks involving document images that have the presence of text, the discriminator network can benefit from the addition of sequential representation learning capabilities. Moreover, the proposed model improved the performance of the Tesseract OCR engine by 23.32% in terms of word accuracy compared to the original noisy receipt images in the POS dataset. Furthermore, the proposed model consistently outperformed other unsupervised classical techniques across all OCR engines considered.

The benefit of the proposed setting is that it allows for training without the need for a ground truth clean image, text, or specific information about the noise type present in the image. This allows training on real-world noisy images, which is required for practical applications. However, creating the

unpaired clean set is a challenging task. Our experiments showed that if the unpaired clean set contains samples that have characteristics different from the noisy samples, the model training is unstable and the images are not cleaned properly. This makes it difficult to train the proposed model when noisy images of diverse nature and with complex degradations are to be used in the training set. Exploring alternative methods for creating a representative clean set that better aligns with the characteristics of the noisy images could be useful.

Additionally, as a future extension to this work, it would be worthwhile to explore different discriminator architectures in this setting to see if the performance can be further improved. Particularly, transformer-based architectures which have superior sequence modeling capabilities than recurrent neural networks can be investigated.

While in this thesis, we demonstrate that unpaired translation models such as CycleGANs can be modified to generate promising results in the context of unpaired document image enhancement, training can still be unstable due to the large number of parameters involved. To address this, future research could explore more stable and powerful generative networks, such as diffusion models, and investigate their application in the context of unpaired document image-to-image translation.

References

[1] A. Abdelhamed, S. Lin, and M. S. Brown, "A high-quality denoising dataset for smartphone cameras," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1692–1700. 2

[2] C. Artaud, N. Sidère, A. Doucet, J.-M. Ogier, and V. P. D. Yooz, "Find it! fraud detection contest report," in *2018 24th International Conference on Pattern Recognition (ICPR)*, IEEE, 2018, pp. 13–18. 30

[3] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9365–9374. 34

[4] S. Bakkali, Z. Ming, M. Coustaty, M. Rusiñol, and O. R. Terrades, "Vlcdoc: Vision-language contrastive pre-training model for cross-modal document classification," *arXiv preprint arXiv:2205.12029*, 2022. 3

[5] S. Bako, S. Darabi, E. Shechtman, J. Wang, K. Sunkavalli, and P. Sen, "Removing shadows from images of documents," in *Asian Conference on Computer Vision*, Springer, 2016, pp. 173–183. 1

[6] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994. 14

[7] A. Biró, A. I. Cuesta-Vargas, J. Martın-Martın, L. Szilágyi, and S. M. Szilágyi, "Synthetized multilanguage ocr using crnn and svtr models for realtime collaborative tools," *Applied Sciences*, vol. 13, no. 7, p. 4419, 2023. 1

[8] Y. Blau, R. Mechrez, R. Timofte, T. Michaeli, and L. Zelnik-Manor, "The 2018 pirm challenge on perceptual image super-resolution," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0. 36

[9] K. v. d. Broek, "Mp3net: Coherent, minute-long music generation from raw audio with a simple convolutional gan," *arXiv preprint arXiv: 2101.04785*, 2021. 6

[10] J. Calvo-Zaragoza and A.-J. Gallego, "A selectional auto-encoder approach for document image binarization," *Pattern Recognition*, vol. 86, pp. 37–47, 2019. 17

[11] X. Chen, X. He, J. Yang, and Q. Wu, "An effective document image deblurring algorithm," in *CVPR 2011*, IEEE, 2011, pp. 369–376. 16

[12] X. Cheng, Z. Fu, and J. Yang, "Zero-shot image super-resolution with depth guided internal degradation learning," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, Springer, 2020, pp. 265–280. 36

[13] Y. Du, Z. Chen, C. Jia, *et al.*, "Svtr: Scene text recognition with a single visual model," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, L. D. Raedt, Ed., Main Track, International Joint Conferences on Artificial Intelligence Organization, Jul. 2022, pp. 884–890. DOI: 10.24963/ijcai.2022/124. [Online]. Available: https://doi.org/10.24963/ijcai.2022/124. 35

[14] D. Dua and C. Graff, *UCI machine learning repository*, 2017. [Online]. Available: http://archive.ics.uci.edu/ml. 29

[15] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990. 14

[16] A. Farahmand, H. Sarrafzadeh, and J. Shanbehzadeh, "Document image noises and removal methods," 2013. 1, 16

[17] W. Fedus, I. Goodfellow, and A. M. Dai, "Maskgan: Better text generation via filling in the_," *arXiv preprint arXiv:1801.07736*, 2018. 6

[18] G. Ganchimeg, "History document image background noise and removal methods," *International Journal of Knowledge Content Development & Technology*, vol. 5, no. 2, pp. 11–24, 2015. 1

[19] M. J. Gangeh, M. Plata, H. R. M. Nezhad, and N. P. Duffy, "End-to-end unsupervised document image blind denoising," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7888–7897. 3, 19

[20] S. Gonwirat and O. Surinta, "Deblurgan-cnn: Effective image denoising and recognition for noisy handwritten characters," *IEEE Access*, vol. 10, pp. 90 133–90 148, 2022. 16

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020. 2, 6

[22] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005. 15

[23] C. Han, H. Hayashi, L. Rundo, *et al.*, "Gan-based synthetic brain mr image generation," in *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, IEEE, 2018, pp. 734–738. 6

[24] J. Han, M. Shoeiby, L. Petersson, and M. A. Armin, "Dual contrastive learning for unsupervised image-to-image translation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 746–755. 2

[25] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1664–1673. 18

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778. 34

[27] P. He, W. Huang, Y. Qiao, C. C. Loy, and X. Tang, "Reading scene text in deep convolutional sequences," in *Thirtieth AAAI conference on artificial intelligence*, 2016. 4, 25

[28] T. Hegghammer, *Noisy ocr dataset (nod)*, version 1.0.0, Zenodo, Jul. 2021. DOI: 10.5281/zenodo.5068735. [Online]. Available: https://doi.org/10.5281/zenodo.5068735. 29

[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997. 14

[30] M. Hong, Y. Qu, C. Li, and S. Chen, "Multi-scale iterative network for underwater image restoration," in *2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI)*, 2019, pp. 201–206. DOI: 10.1109/CCHI.2019.8901915. 36

[31] Z. Hong, X. Fan, T. Jiang, and J. Feng, "End-to-end unpaired image denoising with conditional adversarial networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 4140–4149. 3

[32] M. Hou, X. He, F. Dou, X. Zhang, Z. Guo, and Z. Feng, "Semi-supervised image super-resolution with attention cyclegan," *IET Image Processing*, vol. 16, no. 4, pp. 1181–1193, 2022. 36

[33] K. Hu, Y. Zhang, C. Weng, P. Wang, Z. Deng, and Y. Liu, "An underwater image enhancement algorithm based on generative adversarial network and natural image quality evaluation index," *Journal of Marine Science and Engineering*, vol. 9, no. 7, p. 691, 2021. 36

[34] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708. 3

[35] Z. Huang, K. Chen, J. He, *et al.*, "Icdar2019 competition on scanned receipt ocr and information extraction," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, IEEE, 2019, pp. 1516–1520. 29

[36] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134. 12, 18

[37] S. T. Javed, M. M. Fasihi, A. Khan, and U. Ashraf, "Background and punch-hole noise removal from handwritten urdu text," in *2017 International Multi-topic Conference (INMIC)*, IEEE, 2017, pp. 1–6. 16

[38] Y. Jia and X. Xu, "Chinese named entity recognition based on cnn-bilstm-crf," in *2018 IEEE 9th international conference on software engineering and service science (ICSESS)*, IEEE, 2018, pp. 1–4. 4, 25

[39] J. Jiao, J. Sun, and N. Satoshi, "A convolutional neural network based two-stage document deblurring," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, IEEE, vol. 1, 2017, pp. 703–707. 16

[40] L. Jiao, H. Wu, H. Wang, and R. Bie, "Text recovery via deep cnn-bilstm recognition and bayesian inference," *IEEE Access*, vol. 6, pp. 76 416–76 428, 2018. 25

[41] X. Jin, Z. Chen, J. Lin, Z. Chen, and W. Zhou, "Unsupervised single image deraining with self-supervised constraints," in *2019 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2019, pp. 2761–2765. 3

[42] M. Kang and J. Park, "Contragan: Contrastive learning for conditional image generation," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 357–21 369, 2020. 6

[43] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *International conference on machine learning*, PMLR, 2017, pp. 1857–1865. 2

[44] Y. Kim, J. W. Soh, G. Y. Park, and N. I. Cho, "Transfer learning from synthetic to real-noise denoising with adaptive instance normalization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 3482–3492. 2

[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 63

[46] O. Kodym and M. Hradiš, "TG^2 : Text-guided transformer gan for restoring document readability and perceived quality," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 25, no. 1, pp. 15–28, 2022. 18

[47] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951. 9

[48] R. Kumar, K. Kumar, V. Anand, Y. Bengio, and A. Courville, "Nu-gan: High resolution neural upsampling with gan," *arXiv preprint arXiv:2010.11362*, 2020. 6

[49] A. Lat and C. Jawahar, "Enhancing ocr accuracy with super resolution," in *2018 24th International Conference on Pattern Recognition (ICPR)*, IEEE, 2018, pp. 3162–3167. 16

[50] C. Ledig, L. Theis, F. Huszár, *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690. 12, 18

[51] V. I. Levenshtein *et al.*, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, Soviet Union, vol. 10, 1966, pp. 707–710. 35

[52] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," in *European conference on computer vision*, Springer, 2016, pp. 702–716. 12

[53] M. Li, T. Lv, J. Chen, *et al.*, "Trocr: Transformer-based optical character recognition with pre-trained models," *arXiv preprint arXiv:2109.10282*, 2021. 1

[54] Y. Li, Z. Gan, Y. Shen, *et al.*, "Storygan: A sequential conditional gan for story visualization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6329–6338. 6

[55] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Transactions on Information theory*, vol. 37, no. 1, pp. 145–151, 1991. 9

[56] J.-Y. Liu, Y.-H. Chen, Y.-C. Yeh, and Y.-H. Yang, "Unconditional audio generation with generative adversarial networks and cycle regularization," *arXiv preprint arXiv:2005.08526*, 2020. 6

[57] Y. Liu and S. N. Srihari, "Document image binarization based on texture features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 540–544, 1997. 17

[58] C. Ma, C.-Y. Yang, X. Yang, and M.-H. Yang, "Learning a no-reference quality metric for single-image super-resolution," *Computer Vision and Image Understanding*, vol. 158, pp. 1–16, 2017. 36

[59] X.-J. Mao, C. Shen, and Y.-B. Yang, "Image restoration using convolutional auto-encoders with symmetric skip connections," *arXiv preprint arXiv:1606.08921*, 2016. 17, 18

[60] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2794–2802. 26

[61] J. Memon, M. Sami, R. A. Khan, and M. Uddin, "Handwritten optical character recognition (ocr): A comprehensive systematic literature review (slr)," *IEEE Access*, vol. 8, pp. 142 642–142 668, 2020. 1

[62] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a "completely blind" image quality analyzer," *IEEE Signal processing letters*, vol. 20, no. 3, pp. 209–212, 2012. 36

[63] H. Neji, T. Hamdani, M. Halima, J. Nogueras-Iso, and A. M. Alimi, "Blur2sharp: A gan-based model for document image deblurring," Tech. Rep., 2021. 16, 19

[64] W. Niblack, *An introduction to digital image processing*. Strandberg Publishing Company, 1985. 17

[65] K. Ntirogiannis, B. Gatos, and I. Pratikakis, "Performance evaluation methodology for historical document image binarization," *IEEE Transactions on Image Processing*, vol. 22, pp. 595–609, 2013. 18

[66] L. O'Gorman, "Binarization and multithresholding of document images using connectivity," *CVGIP: Graphical Models and Image Processing*, vol. 56, no. 6, pp. 494–506, 1994. 17

[67] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979. 16, 37, 43, 46

[68] Y. Pang, M. Yuan, Y. Chang, and D.-M. Yan, "Sdalie-gan: Structure and detail aware gan for low-light image enhancement," 2021. 36

[69] S. Park, S. Shin, B. Lee, *et al.*, "Cord: A consolidated receipt dataset for post-ocr parsing," in *Workshop on Document Intelligence at NeurIPS 2019*, 2019. 30

[70] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, "Contrastive learning for unpaired image-to-image translation," in *European conference on computer vision*, Springer, 2020, pp. 319–345. 2

[71] J. Pastor-Pellicer, S. E. Boquera, F. Zamora-Martínez, M. Z. Afzal, and M. J. C. Bleda, "Insights on the use of convolutional neural networks for document image binarization," in *International Work-Conference on Artificial and Natural Neural Networks*, 2015. 17

[72] X. Peng and C. Wang, "Building super-resolution image generator for ocr accuracy improvement," in *Document Analysis Systems: 14th IAPR International Workshop, DAS 2020, Wuhan, China, July 26–29, 2020, Proceedings 14*, Springer, 2020, pp. 145–160. 16

[73] T. Plotz and S. Roth, "Benchmarking denoising algorithms with real photographs," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1586–1595. 2

[74] A. Poddar, A. Chakraborty, J. Mukhopadhyay, and P. K. Biswas, "Texr-gan: A deep adversarial framework for text restoration from deformed handwritten documents," in *Proceedings of the Twelfth Indian Conference on Computer Vision, Graphics and Image Processing*, 2021, pp. 1–9. 18

[75] K. Prajapati, V. Chudasama, H. Patel, *et al.*, "Direct unsupervised super-resolution using generative adversarial network (dus-gan) for real-world data," *IEEE Transactions on Image Processing*, vol. 30, pp. 8251–8264, 2021. DOI: 10.1109/TIP.2021.3113783. 36

[76] A. Randika, N. Ray, X. Xiao, and A. Latimer, "Unknown-box approximation to improve optical character recognition performance," in *International Conference on Document Analysis and Recognition*, Springer, 2021, pp. 481–496. 29

[77] A. Ray, M. Sharma, A. Upadhyay, *et al.*, "An end-to-end trainable framework for joint optimization of document enhancement and recognition," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, IEEE, 2019, pp. 59–64. 18

[78] A. Rehman and T. Saba, "Document skew estimation and correction: Analysis of techniques, common problems and possible solutions," *Applied Artificial Intelligence*, vol. 25, no. 9, pp. 769–787, 2011. 16

[79] V. Robert and H. Talbot, "Does super-resolution improve ocr performance in the real world? a case study on images of receipts," in *ICIP 2020-IEEE International Conference on Image Processing*, IEEE, 2020, pp. 548–552. 16

[80] R. Ronen, S. Tsiper, O. Anschel, I. Lavi, A. Markovitz, and R. Manmatha, "Glass: Global to local attention for scene-text spotting," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*, Springer, 2022, pp. 249–266. 1

[81] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986. 14

[82] Y. Saatci and A. G. Wilson, "Bayesian gan," *Advances in neural information processing systems*, vol. 30, 2017. 10

[83] M. Sarfraz, A. Zidouri, and S. Shahab, "A novel approach for skew estimation of document images in ocr system," in *International Conference on Computer Graphics, Imaging and Visualization (CGIV'05)*, IEEE, 2005, pp. 175–180. 16

[84] J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," *Pattern recognition*, vol. 33, no. 2, pp. 225–236, 2000. 17, 37, 43, 46

[85] M. Sharma, A. Verma, and L. Vig, "Learning to clean: A gan perspective," in *Asian Conference on Computer Vision*, Springer, 2018, pp. 174–185. 16, 19, 37

[86] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 11, pp. 2298–2304, 2016. 4, 25

[87] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4168–4176. 4, 25

[88] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. 3

[89] J. Snell, K. Ridgeway, R. Liao, B. D. Roads, M. C. Mozer, and R. S. Zemel, "Learning to generate images with perceptual similarity metrics," in *2017 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2017, pp. 4277–4281. 48

[90] J. Song, J.-H. Jeong, D.-S. Park, H.-H. Kim, D.-C. Seo, and J. C. Ye, "Unsupervised denoising for satellite imagery using wavelet directional cyclegan," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 8, pp. 6823–6839, 2020. 3

[91] M. A. Souibgui and Y. Kessentini, "De-gan: A conditional generative adversarial network for document enhancement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2, 16, 18

[92] M. A. Souibgui, Y. Kessentini, and A. Fornés, "A conditional gan based approach for distorted camera captured documents recovery," in *Mediterranean Conference on Pattern Recognition and Artificial Intelligence*, Springer, 2020, pp. 215–228. 18

[93] C. Sun and D. Si, "Skew and slant correction for document images using gradient direction," in *Proceedings of the fourth international conference on document analysis and recognition*, IEEE, vol. 1, 1997, pp. 142–146. 16

[94] H. Sun, Z. Kuang, X. Yue, C. Lin, and W. Zhang, "Spatial dual-modality graph reasoning for key information extraction," *arXiv preprint arXiv: 2103.14470*, 2021. 33

[95] C. Szegedy, W. Liu, Y. Jia, *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9. 3

[96] C. Tensmeyer and T. R. Martinez, "Document image binarization with fully convolutional neural networks," *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, pp. 99–104, 2017. 17

[97] R. S. Thakur, R. N. Yadav, and L. Gupta, "State-of-art analysis of image denoising methods using convolutional neural networks," *IET Image Processing*, vol. 13, no. 13, pp. 2367–2380, 2019. 1

[98] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin, "Deep learning on image denoising: An overview," *Neural Networks*, vol. 131, pp. 251–275, 2020. 1

[99] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, "Mocogan: Decomposing motion and content for video generation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1526–1535. 6

[100] Q. N. Vo, S. H. Kim, H. J. Yang, and G. Lee, "Binarization of degraded document images based on hierarchical deep supervised network," *Pattern Recognition*, vol. 74, pp. 568–586, 2018. 18

[101] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," *Advances in neural information processing systems*, vol. 29, 2016. 6

[102] X. Wang and A. Gupta, "Generative image modeling using style and structure adversarial networks," in *European conference on computer vision*, Springer, 2016, pp. 318–335. 18

[103] X. Wang, F. Yu, L. Dunlap, *et al.*, "Deep mixture of experts via shallow embedding," in *Uncertainty in artificial intelligence*, PMLR, 2020, pp. 552–562. 19

[104] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004. 36, 48

[105] L. Weng, "From gan to wgan," *arXiv preprint arXiv:1904.08994*, 2019. 8

[106] S. Wu, W. Zhai, and Y. Cao, "Pixtextgan: Structure aware text image synthesis for license plate recognition," *IET Image Processing*, vol. 13, no. 14, pp. 2744–2752, 2019. 4

[107] S. Wu, C. Dong, and Y. Qiao, "Blind image restoration based on cycle-consistent network," *IEEE Transactions on Multimedia*, 2022. 3

[108] X. Wu, M. Liu, Y. Cao, D. Ren, and W. Zuo, "Unpaired learning of deep image denoising," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV*, Springer, 2020, pp. 352–368. 2

[109] W. Xiong, J. Xu, Z. Xiong, J. Wang, and M. Liu, "Degraded historical document image binarization using local features and support vector machine (svm)," *Optik*, vol. 164, pp. 218–223, 2018. 17

[110] C. Xu, Y. Lu, and Y. Zhou, "An automatic visible watermark removal technique using image inpainting algorithms," in *2017 4th International Conference on Systems and Informatics (ICSAI)*, IEEE, 2017, pp. 1152–1157. 16

[111] J. Xu, X. Ren, J. Lin, and X. Sun, "Diversity-promoting gan: A cross-entropy based generative adversarial network for diversified text generation," in *Proceedings of the 2018 conference on empirical methods in natural language processing*, 2018, pp. 3940–3949. 6

[112] C. Yan, H. Chen, and Z. Yang, "End-to-end medical image denoising via cycle-consistent generative adversarial network," in *2021 International Conference on Information Science, Parallel and Distributed Systems (ISPDS)*, IEEE, 2021, pp. 30–33. 3

[113] Z. Yi, H. Zhang, P. Tan, and M. Gong, "Dualgan: Unsupervised dual learning for image-to-image translation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2849–2857. 2, 18

[114] Y. Yuan, S. Liu, J. Zhang, Y. Zhang, C. Dong, and L. Lin, "Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2018. 3

[115] F. Zamora-Martınez, S. España-Boquera, and M. Castro-Bleda, "Behaviour-based clustering of neural networks applied to document enhancement," in *International Work-Conference on Artificial Neural Networks*, Springer, 2007, pp. 144–151. 1

[116] H. Zhang, S. Su, Y. Zhu, J. Sun, and Y. Zhang, "Boosting no-reference super-resolution image quality assessment with knowledge distillation and extension," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2023, pp. 1–5. 36

[117] H. Zhang, T. Xu, H. Li, *et al.*, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5907–5915. 6

[118] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE transactions on image processing*, vol. 26, no. 7, pp. 3142–3155, 2017. 1

[119] Y. Zhang, Z. Gan, K. Fan, *et al.*, "Adversarial feature matching for text generation," in *International Conference on Machine Learning*, PMLR, 2017, pp. 4006–4015. 6

[120] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 7, pp. 2480–2495, 2020. 1

[121] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232. 2, 3, 11, 37, 43, 46

Appendix A

Hyperparameters

In this section, we discuss the different hyperparameters involved in the training and the steps used for tuning their values. The final values used for training are provided in Section 6.1.3. The goal of our hyperparameter tuning experiments was to select values such that the images generated have improved OCR performance. Table A.1 shows the different hyperparameters involved in training the proposed model. Since there are a large number of hyperparameters involved and the relationship between these parameters is not clear, we assign default values to some of the hyperparameters as proposed by the original CycleGAN paper. However, we performed a set of experiments over parameters in the objective function - λ_{cyc} and λ_{id}. We vary λ_{cyc} over integer values in the range [1,10]. This range of values was suggested by the CycleGAN authors for experimentation. For λ_{id}, the search space was defined over the range [0,1].

Hyperparameters	Details	Default value
Epochs	Number of training epochs	-
Decay epochs	Epoch to start decaying the learning rate	Half of the number of training epochs
λ_{cyc}	Controls the weight for the cycle consistency loss	10
λ_{id}	Controls the weight for the identity loss. This value is multiplied with the λ_{cyc}	0.5
Optimizer	Optimization method for G and D	Adam [45]
learning rate	Step size for updating weights	2×10^{-4}

Table A.1: Training Hyperparameters.

For selecting the optimal number of training epochs, we train models for 20, 30, 40, 50, and 100 epochs and evaluate the validation set using Tesseract OCR. The word-accuracy metric is used for selecting the best model. Figure A.1 shows the effect of the number of training epochs on the validation accuracy for the Kaggle Denoising dataset. The number of decay epochs in each is

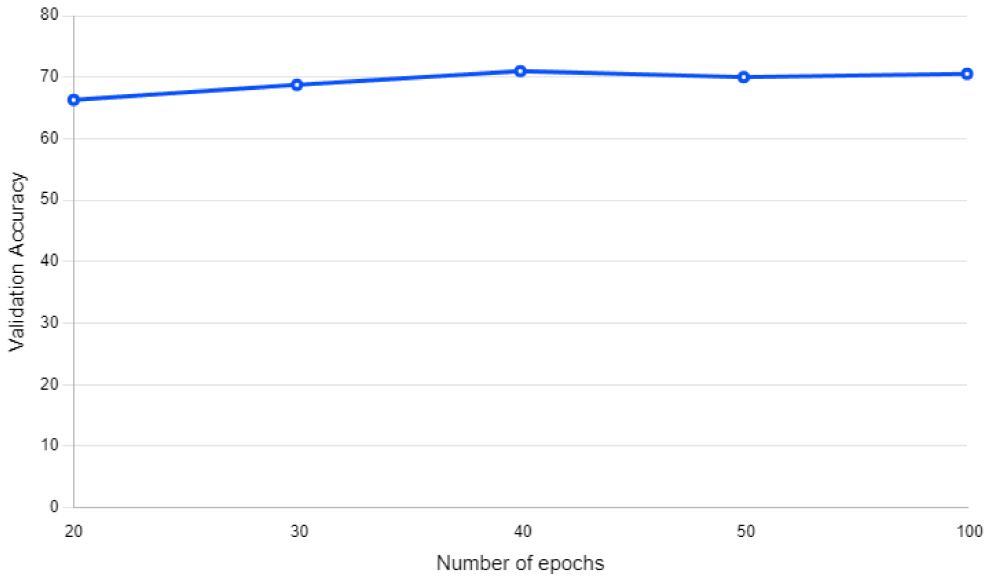


Figure A.1: Effect of number of training epochs on the validation set word-accuracy for Kaggle Denoising dataset evaluated using Tesseract OCR.

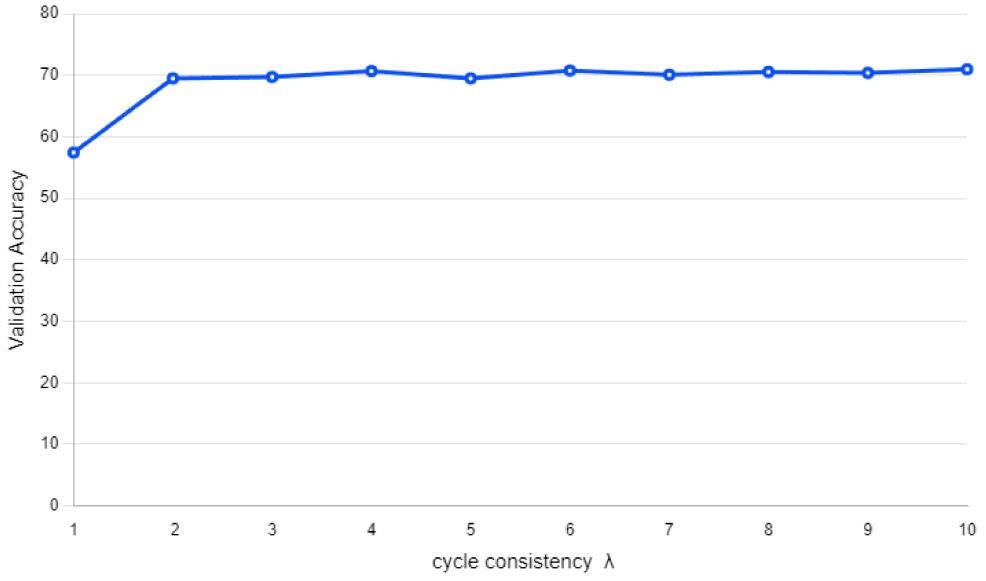


Figure A.2: Effect of cycle consistency λ_{cyc} on the validation set word-accuracy for Kaggle Denoising dataset evaluated using Tesseract OCR.

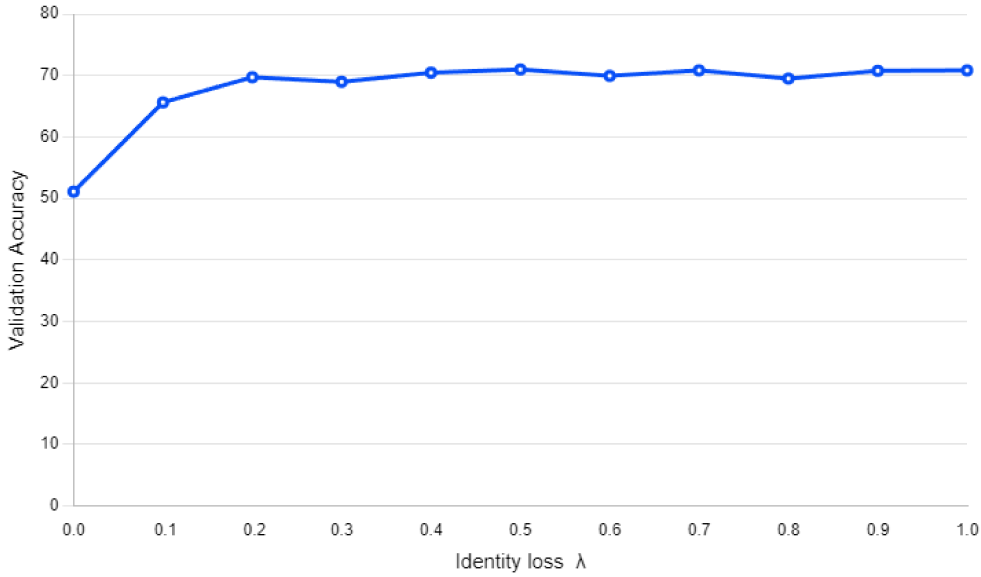


Figure A.3: Effect of identity mapping λ_{id} on the validation set word-accuracy for Kaggle Denoising dataset evaluated using Tesseract OCR.

equal to half of the total number of epochs. Based on the results, we select the number of training epochs as 40. Next, we perform tuning for λ_{cyc}, keeping the default value of 0.5 for λ_{id}. For the values in the search space for L_{cyc}, we train separate models and evaluate them on the validation set. Figure A.2 shows the validation accuracy for the different values of λ_{cyc} for the Kaggle Denoising dataset. Based on the figure, it can be observed that having an extremely low value of λ_{cyc} leads to poorer performance, as expected. Consistent with the results of the CycleGAN paper, we get the optimal value of λ_{cyc} to be 10 for our setup as well. Next, fixing the value of λ_{cyc} as 10, we perform a search over different values of λ_{id} in the defined range. Figure A.3 illustrates the effect of λ_{id} on the validation accuracy for the Kaggle Denoising dataset. The figure highlights that in the absence of identity loss, the performance is poor. Values between 0.5 and 1.0 for λ_{id} have similar performance. Based on similar experiments for other datasets, these values for λ_{cyc} and λ_{id} were consistent and yielded good performance.

Appendix B

Training Curves

In this section, we present the training curves for the generator loss, cycle consistency loss, and discriminator loss of both the standard CycleGAN model and the proposed model. Usually, the cycle consistency loss and overall generator loss tend to decrease during training, indicating improved performance. On the other hand, the discriminator losses typically oscillate. Our findings align with this observation, as we observe similar patterns in the training curves.

Specifically, the generator loss for the CycleGAN model shows oscillations and tends to saturate within a certain range. In contrast, the generator loss curves for the proposed model display a gradual and continuous decrease beyond the limits of the CycleGAN model. The behavior of the cycle consistency loss follows a similar trend.

As expected, the discriminator loss shows oscillations throughout the training process. However, it is important to note that the quality of the generated images should be the primary criteria for assessing the performance of both the CycleGAN and the proposed model. The training curves, while informative, cannot directly determine the superiority or inferiority of the models. It is even emphasized by the authors of the CycleGAN model that training curves may not be a reliable indicator of results. Ultimately, a comprehensive evaluation of the generated images is crucial for comparison and assessment of model performance.

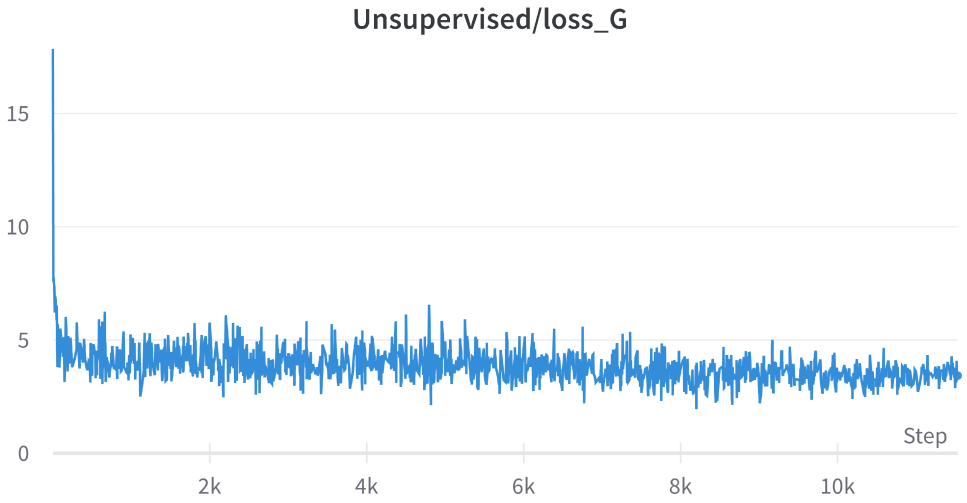


Figure B.1: Overall Generator training loss for the CycleGAN model

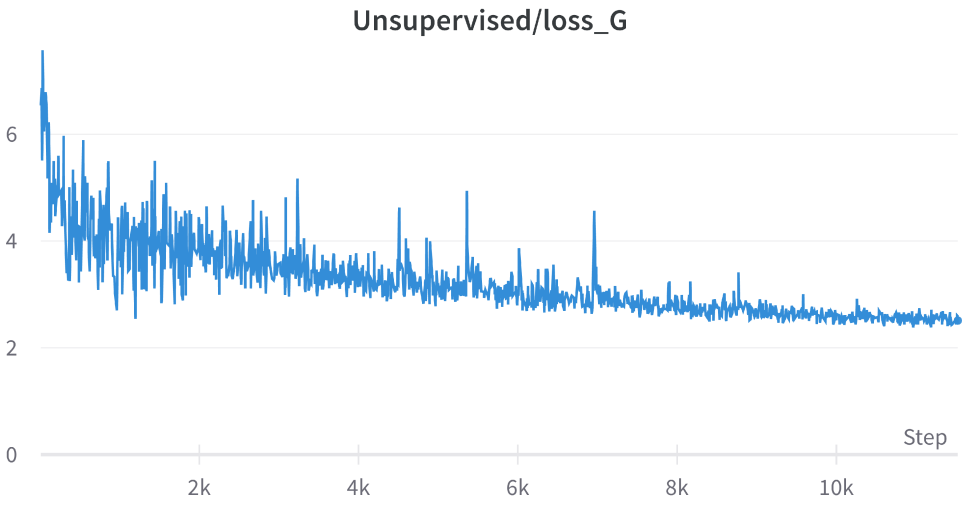


Figure B.2: Overall Generator training loss for the proposed model

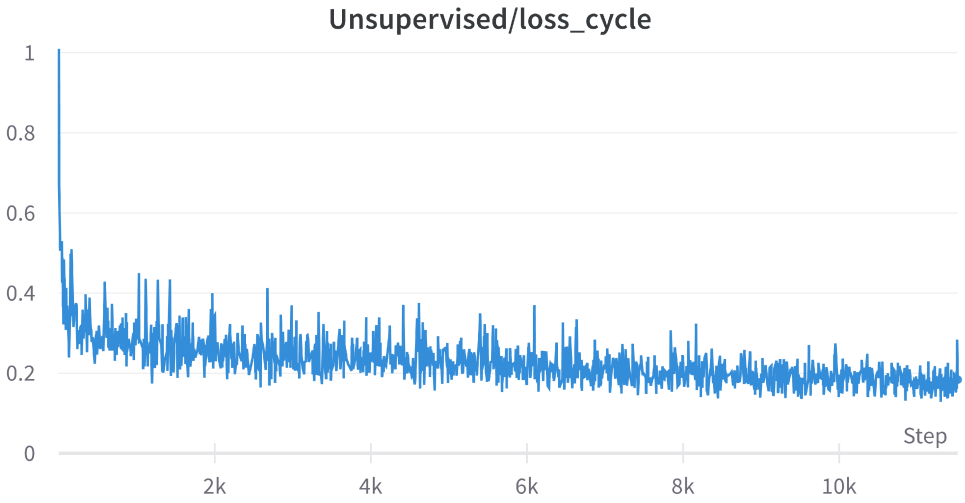


Figure B.3: Cycle consistency training loss for the CycleGAN model

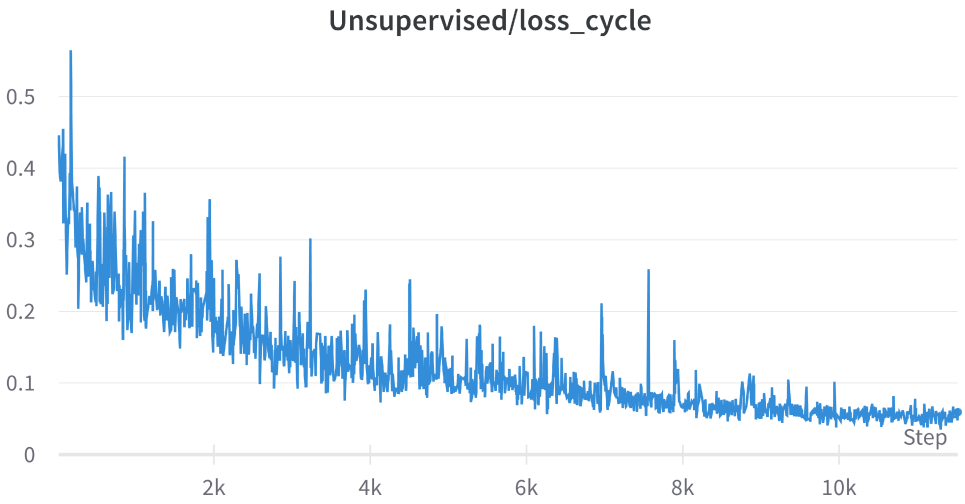


Figure B.4: Cycle consistency training loss for the proposed model

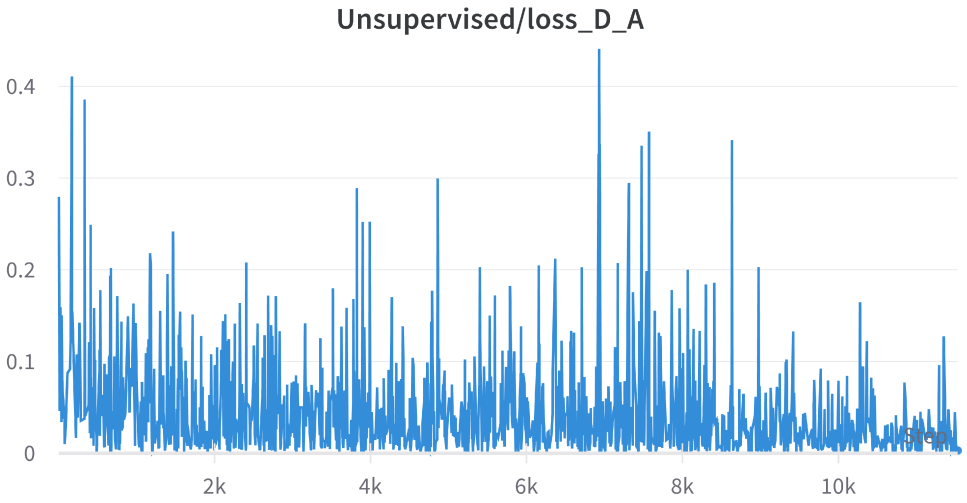


Figure B.5: Discriminator D_A loss for the CycleGAN model

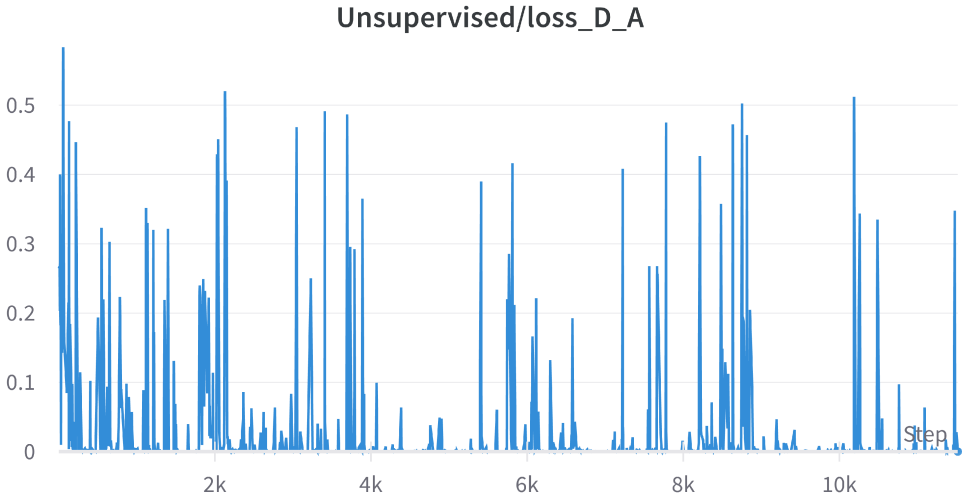


Figure B.6: Discriminator D_A loss for the proposed model

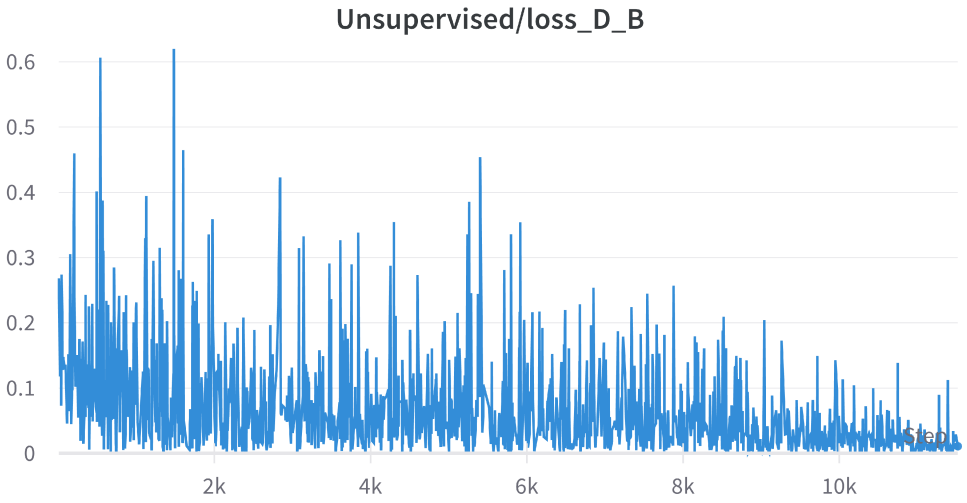


Figure B.7: Discriminator D_B loss for the CycleGAN model

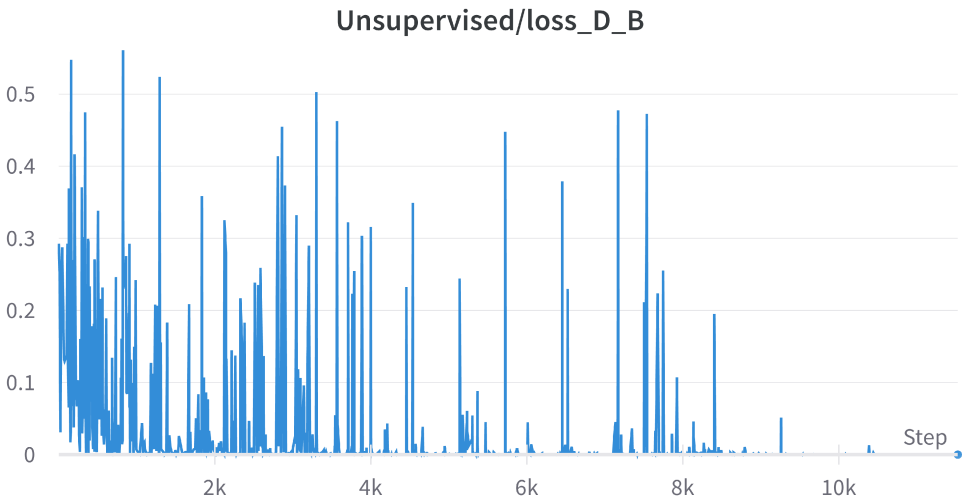


Figure B.8: Discriminator D_B loss for the proposed model