

Article

A Pix2Pix Architecture for Complete Offline Handwritten Text Normalization

Alvaro Barreiro-Garrido [†], Victoria Ruiz-Parrado [†], A. Belen Moreno [†] and Jose F. Velez ^{*,†}

Higher Technical School of Computer Engineering, Universidad Rey Juan Carlos, c/Tulipan sn, Mostoles, 28922 Madrid, Spain; alvaro.barreiro@urjc.es (A.B.-G.); victoria.ruiz.parrado@urjc.es (V.R.-P.); belen.moreno@urjc.es (A.B.M.)

* Correspondence: jose.velez@urjc.es

[†] These authors contributed equally to this work.

Abstract: In the realm of offline handwritten text recognition, numerous normalization algorithms have been developed over the years to serve as preprocessing steps prior to applying automatic recognition models to handwritten text scanned images. These algorithms have demonstrated effectiveness in enhancing the overall performance of recognition architectures. However, many of these methods rely heavily on heuristic strategies that are not seamlessly integrated with the recognition architecture itself. This paper introduces the use of a Pix2Pix trainable model, a specific type of conditional generative adversarial network, as the method to normalize handwritten text images. Also, this algorithm can be seamlessly integrated as the initial stage of any deep learning architecture designed for handwritten recognition tasks. All of this facilitates training the normalization and recognition components as a unified whole, while still maintaining some interpretability of each module. Our proposed normalization approach learns from a blend of heuristic transformations applied to text images, aiming to mitigate the impact of intra-personal handwriting variability among different writers. As a result, it achieves slope and slant normalizations, alongside other conventional preprocessing objectives, such as normalizing the size of text ascenders and descenders. We will demonstrate that the proposed architecture replicates, and in certain cases surpasses, the results of a widely used heuristic algorithm across two metrics and when integrated as the first step of a deep recognition architecture.

Keywords: offline handwriting; scanned text preprocessing; image normalization; IAM dataset; GANs; pix2pix; deep learning



Citation: Barreiro-Garrido, A.; Ruiz-Parrado, V.; Moreno, A.B.; Velez, J.F. A Pix2Pix Architecture for Complete Offline Handwritten Text Normalization. *Sensors* **2024**, *24*, 3892. <https://doi.org/10.3390/s24123892>

Received: 24 April 2024
Revised: 28 May 2024
Accepted: 12 June 2024
Published: 16 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Despite recent advances driven by deep learning, the reliable recognition of handwritten text, captured from either cameras or scanners, is still part of the future. Therefore, handwriting could become a significant means of communication with computers, given factors such as privacy, the widespread use of paper, and its importance in early human learning processes. Many attempts have been made over the years to develop precise HTR systems. Prior to 2013, many solutions were based on Hidden Markov Models (HMMs) as prevailing architectures [1–3]. From 2013 onwards, however, deep learning models have been considered as the standard methodology for offline text recognition.

The intra-personal variability between writers has always been a limiting factor in the capabilities of offline handwritten text recognition (HTR) systems, as each person has their own handwriting style. Also, several other factors such as the thickness of the strokes due to the type of pen used may have an impact on the readability of a given text. The improvement of this readability may boost subsequent recognition results. In order to overcome these issues, a previous text normalization stage has been established as common practice for most HTR algorithms. This preprocessing step is in consonance with

the classic methodology in many types of computer vision problems: the images have to be normalized before the subsequent recognition models are applied [4,5].

In this context, the aim of the normalization step is to reduce the intrinsic variability within the handwritten text images before passing them to an HTR model. This preprocessing task may include several substeps, such as text binarization, image noise removal, slope correction, slant correction, and normalization of ascenders and descenders [6–8]. Classical solutions were based on heuristics algorithms [9,10]. Some authors omitted this normalization process [11,12] with mixed results, but those who have applied normalization have typically obtained as good and usually better results with similar recognition models [5,6,13–16]. In recent years, many authors have researched preprocessing tasks, but most of them were focused only on one of the substeps mentioned above.

Generative Adversarial Neural Networks (GANs) are popular deep learning models that are able to generate high-quality images [17]. In particular, Pix2Pix [18] models are conditional GANs that are able to face what is called paired image-to-image translation. Some works within the literature deal with the use of GAN for image normalization purposes [19]. In the area of automatic document analysis, GAN models have been used to delete background and printed text elements [20], denoise [5], deblur [21] or binarize [22] the images, among others. Some GAN models also normalize the text width with erosion and dilation transformations [23] or try to remove the noise present in document images [24]. However, note that none of these works are focused on the complete normalization process. In this sense, this work presents a new way of action, in which the normalization process is performed by a network that eventually could be trained in an end-to-end configuration that would combine normalization and recognition within a single scheme.

This paper presents a novel handwritten text normalization algorithm based on a Pix2Pix model trained from scratch. The contributions of this paper can be summarized as follows:

- The proposed model reduces the intra-personal variabilities present in the handwritten text of subjects. As far as we know, the complete handwriting text normalization problem with all its substages has not been addressed as a whole by any deep learning model. Moreover, the novelty of our proposal relies on the application of a Pix2Pix network for normalizing handwritten text while preserving the characters and the legibility of the word, rather than for generative purposes.
- The adjustment made to the original Pix2Pix architecture to tackle this particular problem. This adaptation involves a decrease in the number of layers of the U-Net generator and therefore the subsequent reduction in the number of parameters that is necessary to train the model.
- A comparison between the proposed solution and an ad hoc heuristic procedure.

The rest of this paper is organized as follows: Section 2 discusses related works, Section 3 covers the dataset, the preprocessing steps, and the architecture used for normalization. Section 4 describes the experiments and results, whereas Section 5 concludes with the main findings and future directions.

2. Related Works

Different authors have proposed and applied several heuristic normalization strategies over the years in order to improve their HTR models.

Most initial works followed a pattern that typically included similar stages to accomplish the handwritten preprocessing task, namely text binarization, image noise removal, slope correction, slant correction, and normalization of ascenders and descenders. For such preprocessing, image-based techniques were focused on finding automatic thresholds to binarize text and remove noise [25]; on using either Hough transform [26] or horizontal projections [27] to correct the slope; on pixel projections [28], pixel counting [29], or other techniques to calculate the text-cursive leaning and correct the slant; or on baseline detection to normalize the size of ascenders and descenders [30].

Some authors develop complete systems to normalize the images. Those systems are based on heuristic methods. Sueiras et al. [31] propose a combination of heuristic methods to normalize the images. They try to fix most of the common problems regarding HTR, such as noise reduction, binarization, image inversion, slant correction, baseline detection, slope correction, and normalization of ascenders, descenders, and resizing.

More recent works use deep learning techniques in order to achieve automatic pre-processing. Most of them are mainly focused on image binarization and document image enhancement. For example, Akbari et al. [32] and Calvo-Zaragoza et al. [22] applied Convolutional Neural Networks (CNNs), long short-term memory (LSTM), and autoencoders to address these issues.

Other works that also tackle preprocessing are mainly focused on GAN models.

Sadri et al. [33] use a CycleGAN architecture to enhance document images. Another work of this research group uses GAN models for data augmentation [34].

Gonwirat and Surinta [24] handle image denoising via GAN models. They developed both a GAN devoted only to image enhancement and a combination of GAN and CNN models for the enhancement and recognition of handwritten characters.

Souibgui and Kessentini [21] focused their work on the enhancement of document images, this time through the use of conditional GAN models (cGANs). This improvement aimed to remove warping, blurring, watermarks, etc. Their work was extended to train a GAN to enhance the images jointly with a Convolutional Recurrent Neural Network (CRNN) devoted to handwritten text recognition [20]. In this work, the GAN architecture was composed of a U-Net generator and a Fully Convolutional Network discriminator.

Suh et al. [35] managed to remove most of the document image background, including text on the reverse page, stamps, lines, etc. The architecture selected in the first stage is a Pix2Pix GAN, which removes background and color information. Specifically, it is composed of four U-Net neural networks as four color-independent generators and a PatchGAN [18] variation as discriminator. Right after that, in the second stage, local and global binary transformations were applied to improve the first-stage images.

Kang et al. [23] pretrained and combined several U-Nets to deal with specific pre-processing tasks such as dilation, erosion, histogram-equalization, Canny-edge, and Otsu binarization. Their system is specially designed to handle small datasets.

Zhao et al. [36] use conditional GANs to binarize images. More specifically, a first conditional GAN is trained to generate images, whereas a subsequent cascade of conditional GANs is used to compose the image by taking into account multi-scale features.

Kumar et al. [37] describe an unsupervised model which does not need paired images for binarization. These authors create augmented texture images through a GAN model, pseudo-pairing them with the original ones and train a discriminator to differentiate them.

The problem of mimicking the writing style of authors has been investigated by Davis et al. [38]. These authors proposed a model that combined both GANs and autoencoders to capture global style variations of the writers, such as size or slant.

3. Materials and Methods

This section provides a brief review of the handwritten text image databases used to train the models. Then, we explain the heuristic normalization scheme that will be performed. The normalized images of each database will be the ground truth for our trainable models, which will be outlined in the last subsection.

3.1. Handwriting Datasets Used

In this work, we utilized three distinct handwriting datasets: IAM, Bentham, and Osborne.

IAM [39] is an English dataset comprising scanned forms of handwritten text. The forms are segmented into text lines, sentences, and words. The latest version of IAM consists of 115,320 isolated and labeled words distributed over 13,353 text lines from 657 different writers. This was achieved by scanning 1539 pages of text at a 300 DPI resolution. The scanned images were saved in PNG format with 256 gray levels. The IAM partitions used throughout this

work are made up of 47,926 words for the training set, 7558 for validation, and a total amount of 20,292 words for the test set.

Bentham is also an English handwritten dataset [40], but in this case, it has been obtained from historical documents written by a single author: the philosopher Jeremy Bentham (1748–1832). The database comprises 11,473 text lines partitioned into 9198 lines for training, 1415 for validation, and 860 for testing. As this database does not provide segmentation at word level, we have used pseudo-words created by cropping the lines at a fixed step size, with the width being six times the height.

The Osborne dataset is based on the Osborne Archive [41], which includes a collection of handwritten letters exchanged between notable figures from various periods, spanning from 1830 to 1883. The corpus of this dataset is made of 200 RGB-digitized and transcribed documents from the original archive at a 400 DPI resolution. The Osborne dataset is a notable example of a multi-language historical handwritten text database, featuring primarily Spanish texts, with some English and Italian ones. This word image dataset, the smallest of the three, is divided into 7149 words for training, 288 for validation, and 194 images for testing. As this dataset does not provide information at the line level, we have used the adjacent words as context when necessary.

3.2. Normalization Scheme

The heuristic normalization performed on the handwritten images follows the procedure developed by Sueiras et al. [14] and will serve as ground truth for our trainable model.

The following preprocessing steps have been applied to the text line images, rather than directly to the word images. In doing so, some of the selected algorithms severely increase their robustness. This is especially determining in the case of the slope removal algorithm. The normalization method is described in Algorithm 1. The next subsections will describe the normalization steps in detail.

Algorithm 1 Heuristic normalization

```

function NORMALIZE(imgDatabase)
  ouputDatabase = [ ]
  for lineImg ∈ imgDatabase do:
    base = detectBaseLine(lineImg)
    lineImg = slopeCorrection(lineImg, base)
    lineImg = slantCorrection(lineImg)
    b, u = detectBaseAndUpperLine(lineImg)
    lineImg = normAscendersDescenders(lineImg, b, u)
    wordList = cropWords(lineImg)
    for wordImg ∈ wordList do:
      ouputDatabase ← resizeWordImage(wordImg)
    end for
  end for
  return ouputDatabase
end function

```

3.2.1. Image Inversion

We inverted the images with respect to the maximum value of 255 so that the handwritten text strokes have the highest values. On the contrary, background pixels are close to 0. It has been observed in several experiments that this helps CNN networks to produce slightly better results [42].

3.2.2. Slant Correction

Handwritten text images were deslanted using the algorithm provided by [43]. A threshold value of 128 was used to distinguish between black and white pixels. If a pixel is considered as a boundary one (a black pixel with a white one on its right), then the values of the top

three adjacent pixels with respect to that position are analyzed. In doing so, the algorithm accumulates evidence about the local inclination of the text strokes. Once the slant angle α is estimated, an affine transformation is applied to correct it.

3.2.3. Baseline Detection and Slope Correction

To correct the slope of the text lines, we have aimed for a previous identification of the base and upper lines, using an approach similar to [30]. Baseline detection of each text line was carried out by applying a Random Sample Consensus (RANSAC) regression [44] over the lowest points of each column of pixels (with values above a threshold equal to 20). The baseline position with respect to a horizontal line yields an estimate of the slope angle, that will be corrected by an affine transformation. The upper line is obtained afterwards, with the RANSAC procedure being repeated over the highest points of the line image that fulfill the same criterion (although in this case, a threshold of 128 was used).

3.2.4. Normalization of Ascenders and Descenders

After slope correction, base and upper lines are recalculated with RANSAC, assuming that they are now straight horizontal lines. The image gap between them delimits the main body of text, leaving ascenders and descenders as the horizontal top and bottom ends, respectively. These regions are rescaled to half their original size and concatenated again with the core region.

3.2.5. Resizing

Finally, we extract the word images from this line-preprocessing procedure. This has been achieved using the x, y, w, h bounding box coordinates of each word within a given line image. Next, their sizes are fixed to a 48×288 resolution, preserving the aspect ratio of the contained handwritten words.

3.3. The Pix2Pix Architecture

Our Pix2Pix model aims to normalize handwritten text by mapping non-normalized images to their normalized counterparts. The model receives two sets of images: one normalized according to Section 3.2 (ground truth), and a second raw version of the very same set. The Pix2Pix network normalizes the non-normalized images while preserving legibility, using a generator and a discriminator like other GAN architectures.

3.3.1. The Pix2Pix Generator: U-Net

The first component of a Pix2Pix model is the U-Net generator [45]. Figure 1 displays a scheme of our U-Net architecture. The input (top-left corner) of the U-Net generator would be the raw, non-normalized handwritten word images. U-Nets usually work as an encoder–decoder model. Thus, during the decoder expanding path (right side of Figure 1), the U-Net produces as output the normalized version of the input image. In order to avoid losing information during the encoding stage, the model introduces skip connections (gray arrows in Figure 1).

Notice that both the encoding and the decoding paths have been reduced to 4 layers (in contrast with the 8 layers of the original Pix2Pix scheme [18]). This is quite reasonable since our training images only have a resolution of 48×288 . Nevertheless, the latter implies a significant reduction in the trainable parameters and therefore a considerable decrease in the training time as well.

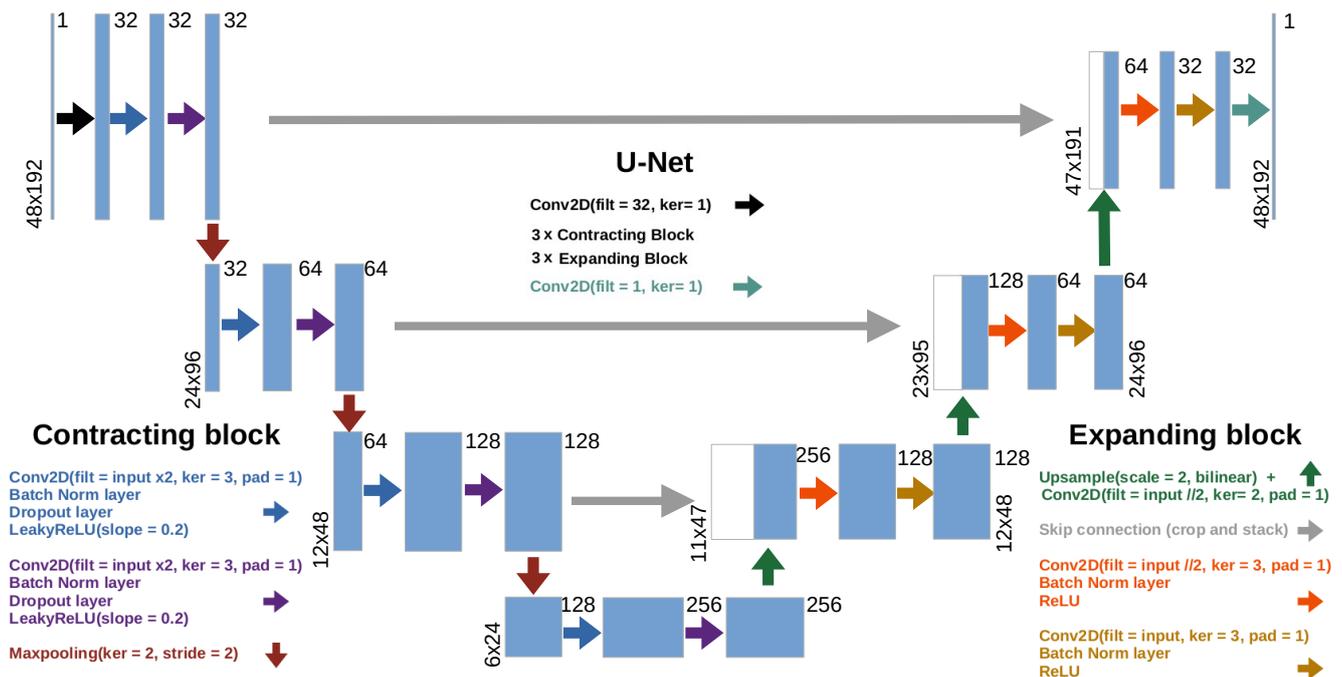


Figure 1. Our U-Net architecture. Each blue box represents a multi-channel feature map. Different arrows represent different operations. The number of channels is shown on top of each box, whereas height and width sizes are specified at the left or right lower edges of the box. White boxes are the cropped feature maps of the encoder part that will be used as skip connections.

3.3.2. The Pix2Pix Discriminator: PatchGAN

The Pix2Pix discriminator, PatchGAN, outputs a matrix of classifications. In our case, each value of this matrix scores the normalization performance of the generator with respect to the raw, non-normalized, image. It takes as input a combination of the raw images and either those normalized using the heuristic preprocessing scheme (Section 3.2), or those provided by the generator. By sliding its field of view across all the patches of the input image, the PatchGAN will then give feedback on how good the normalization process has been for each of these patches. The closer the score to 1, the better the preprocessing of that region. Ideally, the PatchGAN will try to output a matrix of zeros for an image coming from the generator and a matrix of ones for an image preprocessed using the heuristic. However, as the generator improves in its normalization task, it becomes more difficult for the PatchGAN to determine if the given image comes from the generator or has been preprocessed with the scheme introduced in Section 3.2. The PatchGAN discriminator can also be expressed as a sequence of the following: a 2D convolution with 8 filters and kernel 1; two contracting blocks (see Figure 1), the first of which lacking BatchNorm layers; a final 2D convolution with 32 filters and kernel 1. This outputs a final $(1 \times 12 \times 48)$ tensor containing the normalization scores.

3.3.3. Loss Function and L1 Distance

Originally, GANs were generative models, G , that learn to map from a random noise vector z to an output image y , $G : z \rightarrow y$ [17]. However, in conditional GANs, they learn the mapping from an observed image x (i.e., our raw handwritten text images) and some random noise vector, z to y , $G : \{x, z\} \rightarrow y$. Our Pix2Pix U-Net does not even need that random noise vector, which yields: $G : x \rightarrow y$. As stated above, the U-Net generator, which will be the map G , will be trained to output normalized images that cannot be distinguished from those produced using the heuristic procedure stated in Section 3.2. Quite the contrary, the PatchGAN discriminator, D , will try to measure the quality of the normalization by telling whether the image produced came from the generator or from

the heuristic preprocessing scheme (ground truth images) given by [14]. The typical loss function for a conditional GAN is defined as Equation (1) [18]:

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_x[\log(1 - D(x, G(x)))] \quad (1)$$

Bearing in mind that the goal of the generator is to produce normalized handwritten text images as close as possible to the normalization provided by the ground truth set, we introduce an $L1$ distance term to the loss function.

The $L1$ distance measures the similarity between generated and ground truth images. It is a reconstruction term with a weight parameter λ_{recon} used to control its impact on the global loss function. The $L1$ term is defined as

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y}[\|y - G(x)\|_1] \quad (2)$$

Hence, the final expression of the loss function is

$$\mathcal{L}(G, D) = \mathcal{L}_{cGAN}(G, D) + \lambda_{recon}\mathcal{L}_{L1}(G) \quad (3)$$

4. Experiments and Results

The very next step consists of computing the accuracy and effectiveness of our Pix2Pix model. The workflow followed in each experiment of this section (including train and test stages) has been summarized in Figure 2.

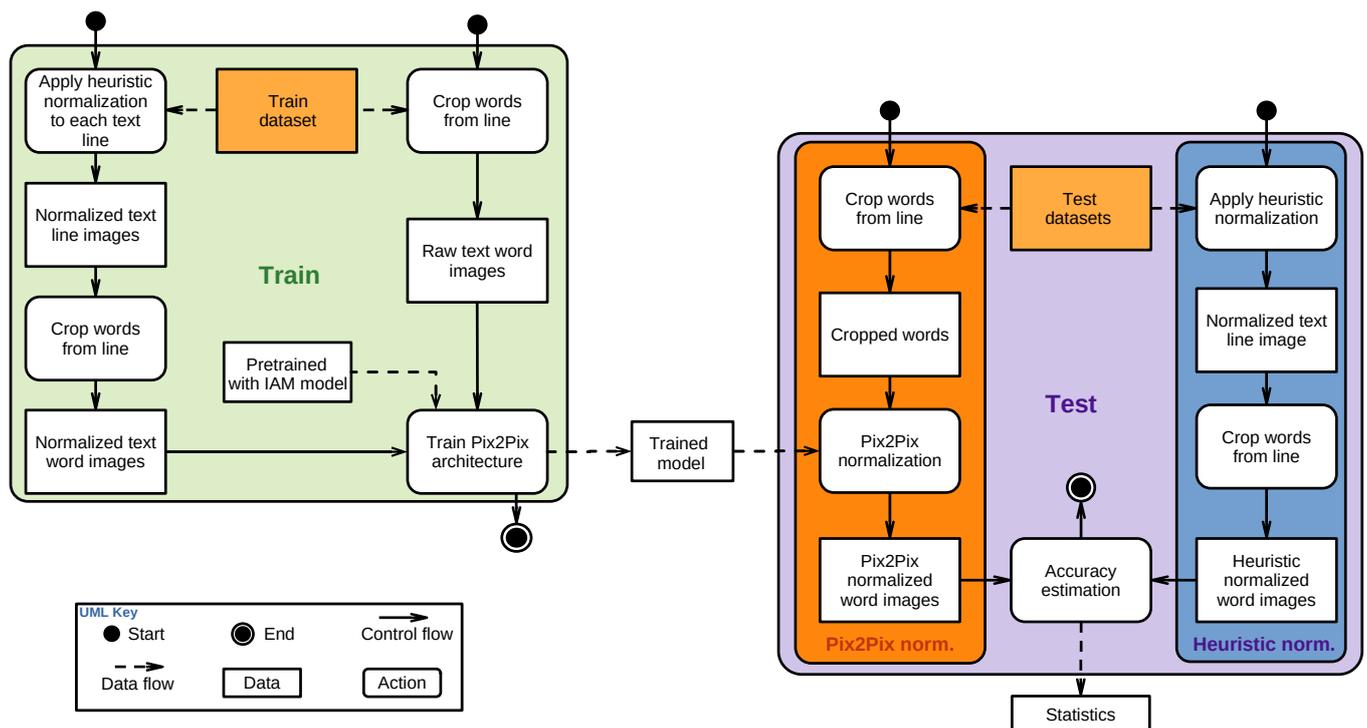


Figure 2. Workflow followed in the different experiments.

The first metric selected to estimate the accuracy of the normalization process for this workflow is the average Distance Per Pixel (DPP hereafter).

Once trained (train region in Figure 2), the Pix2Pix model will perform normalization on the unseen test set images (test region in Figure 2). Then, we will evaluate the pixel-to-pixel $L1$ distance given by Equation (2) between the Pix2Pix output images and a certain set of ground truth counterparts. Let i be an image of the test image set I with range in $[0, 255]$, and let i_n be the n -th pixel of i . We define $N_1(i)$ and $N_2(i)$ as two different normalizations that we want to compare after being applied on i . Then, for an individual image, the DPP is calculated as

$$DPP(i) = \frac{1}{255 \cdot h \cdot w} \sum_{n=1}^{h \cdot w} \|N_1(i_n) - N_2(i_n)\|_1 \quad (4)$$

where h and w correspond to the height and width of the images.

In order to reinforce the reliability of our analysis, we computed a second metric, the *Structural Similarity Index* [46] (SSIM hereafter), to measure and compare the effectiveness of the aforementioned normalization methods. This parameter estimates the similarity between two images by mimicking the human visual perception, and it is given by the following:

$$SSIM(i) = \frac{(2\mu_{N_1(i)}\mu_{N_2(i)} + C_1)(2\sigma_{N_1(i)N_2(i)} + C_2)}{(\mu_{N_1(i)}^2 + \mu_{N_2(i)}^2 + C_1)(\sigma_{N_1(i)}^2 + \sigma_{N_2(i)}^2 + C_2)} \quad (5)$$

where μ_{N_j} , σ_{N_j} , and $\sigma_{N_jN_k}$ are the mean, variance, and covariance of the image i normalized through N_j , respectively. For images in byte integer representation, $C_1 = 255 \times 0.01$ and $C_2 = 255 \times 0.03$, as specified by the authors of [46]. In summary, and given two images, SSIM outputs values in a range between -1 and 1 depending on how close those images are between them in terms of likeness. Hence, totally different images yield a value of -1 , whereas the higher the similarity between them, the closer the outcome is to 1 . Therefore, SSIM explores global similarities between images that have been normalized through two different processes, whereas DPP is constrained to a pixel-level analysis. Once obtained using Equations (4) and (5), DPP and SSIM values are averaged over the whole considered dataset.

4.1. Summary of Experiments

We conducted four distinct experiments using the three different raw datasets. These experiments aimed to determine whether our proposed method can replicate the heuristic transformations performed by Sueiras et al.'s normalization algorithm, which serves as our benchmark. For each experiment, we applied either our normalization algorithm or the heuristic one to different variants of the raw datasets.

The experiments on the various datasets were conducted using the presented Pix2Pix architecture, which was trained with the respective training portion of each dataset. However, due to the smaller size of the Osborne dataset and the limited diversity of the Bentham dataset (a single author), the training for these datasets started from a model pretrained on the IAM dataset.

Also, to measure the impact of the reconstruction term on the final normalization performance, we trained with values of $\lambda_{recon} = 50$, $\lambda_{recon} = 500$, and $\lambda_{recon} = 5000$ for Equation (3). Thus, we tested three different variants of the network for each of the three datasets.

The main aim of each of these experiments may be summarized in the following items:

1. Identifying a value of λ_{recon} for achieving good results on the test partition of each dataset, using the reference heuristic algorithm outputs as ground truth.
2. Assessing whether our proposal restores images from the test partition of each dataset when subjected to distortions, similar to the results obtained by the reference heuristic algorithm.
3. Estimating how our proposal and the reference heuristic algorithm approximate manual ad hoc normalization performed on a subset of the IAM test set.
4. Conducting a preliminary comparison between the text recognition results obtained when using our Pix2Pix normalization approach as a step previous to recognition and those achieved using the reference heuristic normalization.

4.1.1. Experiment 1—Heuristic Replication Measurement

The first experiment consists of testing the extent to which our proposal is able to recreate the heuristic normalization, as this procedure acted as ground truth during our

model training time. To test this assertion, we ran our three variants of the proposed Pix2Pix model on the raw test sets of the three datasets. Once the output normalized images have been obtained, we compare them with those normalized using the scheme presented by Sueiras et al. [14]. Regarding the corresponding results, Table 1 provides the output metrics of the first experiment. As it can be observed, there are no clear advantages when using different values for λ . In particular, the Pix2Pix variant with $\lambda_{recon} = 5000$ (values in bold) gave the best results in terms of both DPP and SSIM for all tested databases. For example, for the IAM database, the model outputs a DPP value of 0.042, which means that, on average, each output pixel differs only by 4.2% with respect to its heuristic ground truth counterpart.

Table 1. DPP and SSIM results for the proposed normalization after being applied on the test partitions of the three different datasets. In both cases, the metrics have been computed by comparing each method with the image resulting from applying that very same procedure on each raw test set.

Dataset	Normalization	DPP	SSIM
IAM	Pix2Pix ($\lambda_{L1} = 50$)	0.052	0.700
	Pix2Pix ($\lambda_{L1} = 500$)	0.047	0.718
	Pix2Pix ($\lambda_{L1} = 5000$)	0.042	0.740
Bentham	Pix2Pix ($\lambda_{L1} = 50$)	0.051	0.470
	Pix2Pix ($\lambda_{L1} = 500$)	0.048	0.498
	Pix2Pix ($\lambda_{L1} = 5000$)	0.045	0.529
Osborne	Pix2Pix ($\lambda_{L1} = 50$)	0.051	0.181
	Pix2Pix ($\lambda_{L1} = 500$)	0.043	0.282
	Pix2Pix ($\lambda_{L1} = 5000$)	0.027	0.613

4.1.2. Experiment 2—Distorted Datasets

We also provide an alternative version of the databases (IAM, Bentham, and Osborne) to test the accuracy of the normalization procedures when being applied to a different set of images. This alternative sets have been obtained by applying several mild-deforming transformations on the original raw test images. Thus, we will refer to these data as the *distorted datasets* henceforth. To obtain this modified versions, we have performed a pipeline of transformations on the raw text images. These transformations are made up of several steps and each of them is applied according to a certain probability. Hence, for each image, the pipeline of transformations involves the following steps: A 50% chance of undergoing an elastic distortion as the one proposed by Yousef and Bishop [47]. A 40% probability per image of being rotated with a random angle within the range of ± 5 degrees. An 80% probability of adding a slant to the image, with random angles between ± 0.5 degrees. Finally, a 30% probability of undergoing through either a 2×2 dilation or a 2×2 erosion. These probabilities have been determined by previous experiments.

These distorted images will hence be normalized by the three Pix2Pix reconstruction loss-variants described above and also by the reference heuristic method in order to compare the effectiveness of the normalization approaches when facing “unseen” handwritten text images. This time, however, each normalization was compared to itself being applied on the raw, non-distorted images. The results in terms of DPP and SSIM are presented in Table 2.

Even though the best outcomes were obtained by our model with $\lambda_{recon} = 5000$, the obtained results were very similar for all the metrics. Therefore, we can conclude that both our proposal and the reference heuristic algorithm restore the test dataset images in a similar way, perhaps with some advantage for the proposed method.

Table 2. DPP and SSIM results for the proposed normalization after being applied on the *distorted datasets*. In both cases, the metrics have been computed by comparing each method with the image resulting from applying that very same procedure on each raw (non-distorted) dataset.

Dataset	Normalization	DPP	SSIM
Distorted IAM	Pix2Pix ($\lambda_{L1} = 50$)	0.036	0.758
	Pix2Pix ($\lambda_{L1} = 500$)	0.031	0.784
	Pix2Pix ($\lambda_{L1} = 5000$)	0.016	0.862
	Heuristic method [14]	0.046	0.772
Distorted Bentham	Pix2Pix ($\lambda_{L1} = 50$)	0.032	0.737
	Pix2Pix ($\lambda_{L1} = 500$)	0.033	0.747
	Pix2Pix ($\lambda_{L1} = 5000$)	0.032	0.738
	Heuristic method [14]	0.049	0.643
Distorted Osborne	Pix2Pix ($\lambda_{L1} = 50$)	0.025	0.854
	Pix2Pix ($\lambda_{L1} = 500$)	0.025	0.858
	Pix2Pix ($\lambda_{L1} = 5000$)	0.023	0.858
	Heuristic method [14]	0.030	0.806

4.1.3. Experiment 3—Comparison with Manual Ad Hoc Normalization

In this third experiment, we randomly selected a sample of 100 images from the *raw IAM dataset*. These images were manually normalized by visual inspection, accounting especially for slope and slant corrections and preserving the aspect ratio. In addition, to fit the 48×288 resolution of our model, we ensured that the main body parts of these images were constrained to the central 24 pixels, whereas the other regions were left to ascenders and descenders. In doing so, we have generated an alternative ground truth set with which we are able to compare all the previously explained normalization algorithms. Figure 3 displays the raw version of one of these IAM test set images as well as its ad hoc normalized counterpart, whereas Table 3 encapsulates the values obtained for the computed metrics. In this case, we find that based on the DPP metric, our Pix2Pix proposal slightly surpasses the results of the heuristic algorithm for every considered λ parameter. The SSIM results, however, were very much alike for all the normalizations performed.



Figure 3. Example of the ad hoc normalization dataset: left word belongs to the *raw IAM dataset*, whereas the right word displays the same image after being manually normalized by visual inspection.

Table 3. DPP, SSIM, and slant-angle metrics obtained using as ground truth a manual normalization of 100 random images from the *raw IAM dataset*.

Normalization	DPP	SSIM
Pix2Pix ($\lambda_{L1} = 50$)	0.089	0.556
Pix2Pix ($\lambda_{L1} = 500$)	0.081	0.582
Pix2Pix ($\lambda_{L1} = 5000$)	0.073	0.595
Heuristic method [14]	0.102	0.545

4.1.4. Experiment 4—Examining Normalization Architecture in Deep Recognition

For our final experiment, we conducted a preliminary comparison between the results obtained from our normalization approach and those achieved using the heuristic normalization method proposed by Sueiras et al. and their own recognition model [14]. Sueiras et al. reported a Word Error Rate (WER) of 23.8% and a Character Error Rate (CER, based on Levenshtein distance) of 8.8% when applying their CNN+Seq2Seq LSTM-based

attention model to the IAM test dataset with their heuristic normalization procedure, which we have adopted as the baseline for our work. After replacing their heuristic normalization with our Pix2Pix normalization method (see Figure 4), we observed a decrease in WER to 18.1% and in CER to 5.7% (see Table 4). This replacement occurred in two stages: first, we substituted the input layer of the CNN+Seq2Seq LSTM-based attention model with our pretrained Pix2Pix architecture; then, we trained the resulting model using the IAM training dataset, stopping if no further improvements in validation loss were observed for 100 epochs.

Table 4. End-to-end preliminary recognition results when using a CNN+Seq2Seq LSTM-based attention model [14] after the normalization.

Normalization	WER	CER
Pix2Pix ($\lambda_{L1} = 5000$)	18.1%	5.7%
Heuristic method [14]	23.8%	8.8%

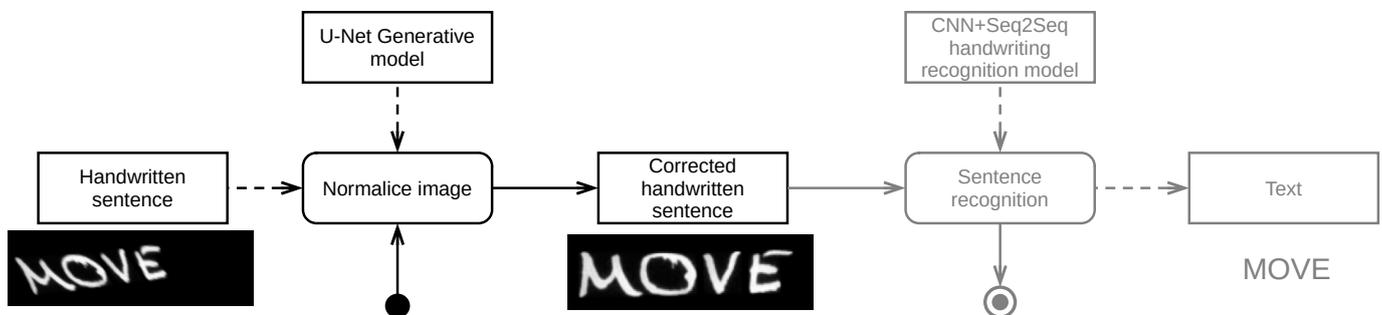


Figure 4. Preliminary end-to-end architecture used in experiment 4.

4.2. Discussion

Theoretically, the normalization stage quality should improve as the DPP decreases and SSIM increases, respectively. Following this rationale, the experiment with $\lambda_{recon} = 5000$ provides some of the best results (see Figure 5). Nevertheless, a visual inspection of some of the network outcomes allows us to see whether this is true or not. In doing so, we will have a more qualitative and reliable intuition about the Pix2Pix normalization performance with respect to the “ideal” preprocessing of the ground truth images.

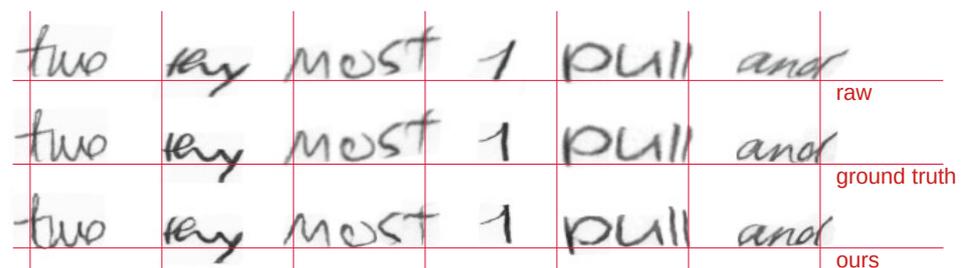


Figure 5. Examples of validation word images. From top to bottom: Raw, ground truth, and generated images by our Pix2Pix architecture with $\lambda_{recon} = 5000$.

To illustrate the effectiveness of the model in a challenging case, we present the results of applying different normalization methods to a word image example, from the Osborne dataset, that exhibits overlapping text strokes (see Figure 6). It can be seen that our approach yields better results than the heuristic method and additionally reduces some of the noise present within the image.

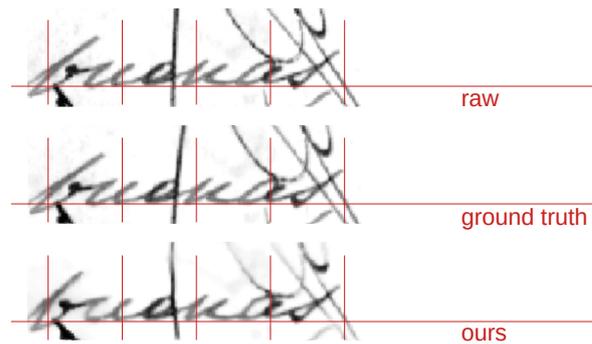


Figure 6. Example of a Spanish handwritten word image from the Osborne dataset that presents overlapping of the handwritten text strokes.

We found that our Pix2Pix provides a good normalization with respect to heuristic preprocessing as long as the resizing factor (understood as the difference in the size of text between raw and heuristic images, relative to the entire image) is low. In such cases, most of the other features present in raw input images (slope, slant, low contrast between background and text) are properly corrected. Nevertheless, whenever this resizing of the text with respect to the raw input image becomes noticeable, the normalization performance of our model drops significantly.

Nonetheless, we mostly observe that for our conducted experiments, the trained Pix2Pix architecture effectively replicates the outcomes achieved by the reference heuristic algorithm for handwritten text normalization. These favorable results have facilitated the successful integration of this normalization stage into an end-to-end recognition architecture, enabling a comprehensive optimization of all model parameters.

5. Conclusions and Future Work

We present a Pix2Pix conditional GAN model to normalize handwritten text images. A normalized version of the IAM dataset, applying typical preprocessing routines on the original images, has served as the pretrained ground truth for the presented model. Also, a fine-tuning training has been performed for both the Bentham and the Osborne datasets. Four experiments were conducted on different variants or subsamples of the three aforementioned datasets. These experiments tried to measure both the reliability of the proposed normalization and its capability to generalize the results obtained on the training ground truth. To quantify the latter, we have built our analysis of the results upon two metrics, the *Distance Per Pixel* and the *Structural Similarity Index Measure*.

Concerning our four conducted experiments, and even though most of the word images were properly normalized, we observed that those undergoing a significant text resize during the preprocessing stage were significantly more difficult to normalize for the Pix2Pix model. This suggests that resizing is a limiting factor for the capacity of Pix2Pix architectures regarding this normalization task. However, its removal from the global preprocessing procedure may affect the recognition performance of a subsequent recognition model. On the other hand, some adjustments in the proposed architecture and in the steps of the ground truth normalization could be helpful in order to overcome the effect of resizing.

Once the Pix2Pix model proposed in the present work has been trained, it has been jointly integrated with a recognition architecture in an end-to-end configuration. In doing so, the recognition architecture provides Pix2Pix with feedback to improve the normalization phase. So, this new combined HTR model has improved the overall recognition performance of the equivalent non-normalized model. Furthermore, splitting the model into two components (normalization and recognition) enhances its interpretability. Moreover, this architecture could be used for other border normalization tasks, like the recognition of living organisms' contours. As future works, we will explore some modifications to the

Pix2Pix architecture in order to resolve the detected resizing problems, and we will also try to apply this architecture to other similar problems.

Author Contributions: Conceptualization, J.F.V. and V.R.-P.; methodology, V.R.-P.; software, A.B.-G. and A.B.M.; validation, A.B.M.; formal analysis, V.R.-P.; investigation, A.B.-G.; data curation, A.B.-G. and A.B.M.; writing—original draft preparation, A.B.-G., V.R.-P. and J.F.V.; writing—review and editing, A.B.M., A.B.-G., J.F.V. and V.R.-P.; resources, supervision, project administration and funding acquisition, J.F.V. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge the financial support given by the Spanish MICINN Project: PID2021-124064OB-I00.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All the algorithms explained in this paper, as well as the manually normalized 100 images from the IAM test dataset used in experiment 3, can be found in https://github.com/jfvelezserrano/Handwritten_Analysis/tree/main/sources/normalization (accessed on 14 June 2024).

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Networks
DPI	Dots per Inch
DPP	Distance Per Pixel
GAN	Generative Adversarial Neural Networks
HTR	Handwritten Text Recognition
IAM dataset	IAM Handwriting Database
LSTM	Long Short-Term Memory
RANSAC	Random Sample Consensus
SSIM	Structural Similarity Index

References

1. España-Boquera, S.; Castro-Bleda, M.; Gorbe-Moya, J.; Zamora Martinez, F. Improving offline handwritten text recognition with hybrid HMM/ANN models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 767–779. [[CrossRef](#)] [[PubMed](#)]
2. Graves, A.; Liwicki, M.; Fernández, S.; Bertolami, R.; Bunke, H.; Schmidhuber, J. A novel connectionist system for unconstrained handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 855–868. [[CrossRef](#)] [[PubMed](#)]
3. Vinciarelli, A.; Bengio, S.; Bunke, H. Offline recognition of unconstrained handwritten texts using HMMs and statistical language models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 709–720. [[PubMed](#)]
4. Pei, S.C.; Lin, C.N. Image normalization for pattern recognition. *Image Vis. Comput.* **1995**, *13*, 711–723. [[CrossRef](#)]
5. Neji, H.; Ben Halima, M.; Noguera-Iso, J.; Hamdani, T.M.; Lacasta, J.; Chabchoub, H.; Alimi, A.M. Doc-Attentive-GAN: Attentive GAN for historical document denoising. *Multimed. Tools Appl.* **2023**, *83*, 55509–55525. [[CrossRef](#)]
6. Dutta, K.; Krishnan, P.; Mathew, M.; Jawahar, C. Improving CNN-RNN Hybrid Networks for Handwriting Recognition. In Proceedings of the 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), Niagara Falls, NY, USA, 5–8 August 2018; pp. 80–85.
7. Ganeshiah, K.P.; Hegde, V. Handwritten character recognition using optimization based skewed line segmentation method and multi-class support vector machine. *Int. J. Adv. Technol. Eng. Explor.* **2023**, *10*, 1476–1490. [[CrossRef](#)]
8. Bera, S.K.; Chakrabarti, A.; Lahiri, S.; Barney Smith, E.H.; Sarkar, R. Normalization of unconstrained handwritten words in terms of Slope and Slant Correction. *Pattern Recognit. Lett.* **2019**, *128*, 488–495. [[CrossRef](#)]
9. Nidhi; Ghosh, D.; Chaurasia, D.; Mondal, S.; Mahajan, A. Handwritten Documents Text Recognition with Novel Pre-processing and Deep Learning. In Proceedings of the 2021 Grace Hopper Celebration India, GHCI 2021, Bangalore, India, 19 February–12 March 2021. [[CrossRef](#)]
10. Chu, S.C.; Yang, X.; Zhang, L.; Snásel, V.; Pan, J.S. Hybrid optimization algorithm for handwritten document enhancement. *Comput. Mater. Contin.* **2024**, *78*, 3763–3786. [[CrossRef](#)]

11. Kass, D.; Vats, E. AttentionHTR: Handwritten Text Recognition Based on Attention Encoder-Decoder Networks. In *Document Analysis Systems*; Uchida, S., Barney, E., Eglin, V., Eds.; Springer International Publishing: Cham, Switzerland, 2022; pp. 507–522.
12. Kang, L.; Toledo, J.I.; Riba, P.; Villegas, M.; Fornés, A.; Rusiñol, M. Convolve, Attend and Spell: An Attention-based Sequence-to-Sequence Model for Handwritten Word Recognition. In *Pattern Recognition, Proceedings of the 40th German Conference, GCPR 2018, Stuttgart, Germany, 9–12 October 2018*; Brox, T., Bruhn, A., Fritz, M., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 459–472.
13. Michael, J.; Labahn, R.; Grüning, T.; Zöllner, J. Evaluating Sequence-to-Sequence Models for Handwritten Text Recognition. In *Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, NSW, Australia, 20–25 September 2019*; pp. 1286–1293.
14. Sueiras, J.; Ruiz, V.; Sanchez, A.; Velez, J. Offline continuous handwriting recognition using seq2seq neural networks. *Neurocomputing* **2018**, *289*, 119–128. [\[CrossRef\]](#)
15. Pesch, H.; Hamdani, M.; Forster, J.; Ney, H. Analysis of Preprocessing Techniques for Latin Handwriting Recognition. In *Proceedings of the 2012 International Conference on Frontiers in Handwriting Recognition, Bari, Italy, 18–20 September 2012*; pp. 280–284.
16. Kozielski, M.; Forster, J.; Ney, H. Moment-Based Image Normalization for Handwritten Text Recognition. In *Proceedings of the 2012 International Conference on Frontiers in Handwriting Recognition, Bari, Italy, 18–20 September 2012*; pp. 256–261.
17. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *3*, 2672–2680.
18. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017*; pp. 5967–5976.
19. Jabbar, A.; Li, X.; Omar, B. A Survey on Generative Adversarial Networks: Variants, Applications, and Training. *ACM Comput. Surv. (CSUR)* **2022**, *54*, 1–49. [\[CrossRef\]](#)
20. Khamekhem Jemni, S.; Souibgui, M.; Kessentini, Y.; Fornés, A. Enhance to read better: A Multi-Task Adversarial Network for Handwritten Document Image Enhancement. *Pattern Recognit.* **2022**, *123*, 108370. [\[CrossRef\]](#)
21. Souibgui, M.; Kessentini, Y. DE-GAN: A Conditional Generative Adversarial Network for Document Enhancement. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 1180–1191. [\[CrossRef\]](#)
22. Calvo-Zaragoza, J.; Gallego, A.J. A selectional auto-encoder approach for document image binarization. *Pattern Recognit.* **2019**, *86*, 37–47. [\[CrossRef\]](#)
23. Kang, S.; Iwana, B.; Uchida, S. Complex image processing with less data—Document image binarization by integrating multiple pre-trained U-Net modules. *Pattern Recognit.* **2021**, *109*, 107577. [\[CrossRef\]](#)
24. Gonwirat, S.; Surinta, O. DeblurGAN-CNN: Effective Image Denoising and Recognition for Noisy Handwritten Characters. *IEEE Access* **2022**, *10*, 90133–90148. [\[CrossRef\]](#)
25. Sauvola, J.; Pietikäinen, M. Adaptive document image binarization. *Pattern Recognit.* **2000**, *33*, 225–236. [\[CrossRef\]](#)
26. Pal, U.; Chaudhuri, B. An improved document skew angle estimation technique. *Pattern Recognit. Lett.* **1996**, *17*, 899–904. [\[CrossRef\]](#)
27. Kavallieratou, E.; Fakotakis, N.; Kokkinakis, G. Skew angle estimation for printed and handwritten documents using the Wigner–Ville distribution. *Image Vis. Comput.* **2002**, *20*, 813–824. [\[CrossRef\]](#)
28. Pastor, M.; Toselli, A.; Vidal, E. Projection Profile Based Algorithm for Slant Removal. In *Image Analysis and Recognition, Proceedings of the International Conference, ICIAR 2004, Porto, Portugal, 29 September–1 October 2004*; Campilho, A., Kamel, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; pp. 183–190.
29. Vinciarelli, A.; Luetin, J. A new normalization technique for cursive handwritten words. *Pattern Recognit. Lett.* **2001**, *22*, 1043–1050. [\[CrossRef\]](#)
30. Gupta, J.D.; Chanda, B. An Efficient Slope and Slant Correction Technique for Off-Line Handwritten Text Word. In *Proceedings of the 2014 Fourth International Conference of Emerging Applications of Information Technology, Kolkata, India, 19–21 December 2014*; pp. 204–208.
31. Sueiras, J. Continuous Offline Handwriting Recognition using Deep Learning Models. *arXiv* **2021**, arXiv:2112.13328.
32. Akbari, Y.; Al-Maadeed, S.; Adam, K. Binarization of Degraded Document Images Using Convolutional Neural Networks and Wavelet-Based Multichannel Images. *IEEE Access* **2020**, *8*, 153517–153534. [\[CrossRef\]](#)
33. Sadri, N.; Desir, J.; Khallouli, W.; Shahab Uddin, M.; Kovacic, S.; Sousa-Poza, A.; Cannan, M.; Li, J. Image Enhancement for Improved OCR and Deck Segmentation in Shipbuilding Document Images. In *Proceedings of the 2022 IEEE 13th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference, UEMCON 2022, New York, NY, USA, 26–29 October 2022*; pp. 45–51.
34. Khallouli, W.; Pamie-George, R.; Kovacic, S.; Sousa-Poza, A.; Canan, M.; Li, J. Leveraging Transfer Learning and GAN Models for OCR from Engineering Documents. In *Proceedings of the 2022 IEEE World AI IoT Congress, AIIoT 2022, Seattle, WA, USA, 6–9 June 2022*; pp. 15–21.
35. Suh, S.; Kim, J.; Lukowicz, P.; Lee, Y. Two-stage generative adversarial networks for binarization of color document images. *Pattern Recognit.* **2022**, *130*, 108810. [\[CrossRef\]](#)
36. Zhao, J.; Shi, C.; Jia, F.; Wang, Y.; Xiao, B. Document image binarization with cascaded generators of conditional generative adversarial networks. *Pattern Recognit.* **2019**, *96*, 106968. [\[CrossRef\]](#)

37. Kumar, A.; Ghose, S.; Chowdhury, P.; Roy, P.; Pal, U. UdbNet: Unsupervised document binarization network via adversarial game. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 7817–7824.
38. Davis, B.; Tensmeyer, C.; Price, B.; Wigington, C.; Morse, B.; Jain, R. Text and Style Conditioned GAN for Generation of Offline Handwriting Lines. In Proceedings of the 31st British Machine Vision Conference—BMVC 2020, Virtual, 7–10 September 2020; BMVA Press: Surrey, UK, 2020; pp. 1–13.
39. Marti, U.V.; Bunke, H. The IAM-database: An English sentence database for offline handwriting recognition. *Int. J. Doc. Anal. Recognit.* **2003**, *5*, 39–46. [[CrossRef](#)]
40. Causer, T.; Wallace, V. Building a volunteer community: Results and findings from Transcribe Bentham. *Digit. Humanit. Q.* **2012**, *6*. Available online: <http://www.digitalhumanities.org/dhq/vol/6/2/000125/000125.html> (accessed on 11 June 2024).
41. Barreiro-Garrido, A.; Ruiz-Parrado, V.; Sanchez, A.; Velez, J.F. Handwritten Word Recognition on the Fundación-Osborne Dataset. In *Bio-inspired Systems and Applications: From Robotics to Ambient Intelligence*; Springer: Cham, Switzerland, 2022; pp. 298–307. [[CrossRef](#)]
42. Chowdhury, A.; Vig, L. An efficient end-to-end neural model for handwritten text recognition. In Proceedings of the British Machine Vision Conference 2018—BMVC 2018, Newcastle, UK, 3–6 September 2018; pp. 1–13.
43. Kim, G.; Govindaraju, V. A lexicon driven approach to handwritten word recognition for real-time applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **1997**, *19*, 366–379.
44. Fischler, M.A.; Bolles, R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM* **1981**, *24*, 381–395. [[CrossRef](#)]
45. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, Proceedings of the 18th International Conference, Munich, Germany, 5–9 October 2015*; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.
46. Wang, Z.; Bovik, A.; Sheikh, H.; Simoncelli, E. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)]
47. Yousef, M.; Bishop, T.E. OrigamiNet: Weakly-Supervised, Segmentation-Free, One-Step, Full Page Text Recognition by Learning to Unfold. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 14698–14707.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.