Expert Systems With Applications 140 (2020) 112916

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Text-line extraction from handwritten document images using GAN

Soumyadeep Kundu[a], Sayantan Paul[a], Suman Kumar Bera[a,*], Ajith Abraham[b,c], Ram Sarkar[a]

[a] Computer Science and Engineering Department, Jadavpur University, Kolkata, India
[b] Department of Computer Science, University of Pretoria, South Africa
[c] Scientific Network for Innovation and Research Excellence, Machine Intelligence Research Labs (MIR Labs), WA, USA

ARTICLE INFO

Article history:
Received 17 March 2019
Revised 3 August 2019
Accepted 31 August 2019
Available online 2 September 2019

Keywords:
GAN
Deep Learning
Text-line extraction
Handwritten documents
HIT-MW dataset
ICDAR dataset

ABSTRACT

Text-line extraction (TLE) from unconstrained handwritten document images is still considered an open research problem. Literature survey reveals that use of various rule-based methods is commonplace in this regard. But these methods mostly fail when the document images have touching and/or multi-skewed text lines or overlapping words/characters and non-uniform inter-line space. To encounter this problem, in this paper, we have used a deep learning-based method. In doing so, we have, for the first time in the literature, applied Generative Adversarial Networks (GANs) where we have considered TLE as image-to-image translation task. We have used U-Net architecture for the Generator, and Patch GAN architecture for the discriminator with different combinations of loss functions namely GAN loss, L1 loss and L2 loss. Evaluation is done on two datasets: handwritten Chinese text dataset HIT-MW and ICDAR 2013 Handwritten Segmentation Contest dataset. After exhaustive experimentations, it has been observed that U-Net architecture with combination of the said three losses not only produces impressive results but also outperforms some state-of-the-art methods.

1. Introduction

We live in such an electronic era where the development of information technology is really omnipresent in day-to-day life. The rapid growth of electronic media thus emphasizes the digital transcript of paper documents. There is an abundance of ethos in the form of old manuscripts, texts and books that provide a lot of information over the years. Such documents become unusable while searching an information among thousands of documents. Thus, a necessity arises to store the paper documents in machine editable format for better storage and quick information retrieval. The performance of a document analysis and recognition (DAR) system depends on a series of stages like text non-text separation (Bhowmik, Sarkar, Nasipuri, & Doermann, 2018), text-line or word extraction (Malakar et al., 2012; Shi & Govindaraju, 2004) and their skew and slant correction (Bera et al., 2017; Kar et al., 2019), character and/or word recognition (Das, Singh, Bhowmik, Sarkar, & Nasipuri, 2016) etc. This type of pre-processing becomes more challenging for free-style handwritten documents in comparison with printed documents. In this work, we focus only on TLE from unconstrained handwritten document images. Text-line extraction (TLE) is thus an important part of the document image processing and is used in the text conversion process to identify lines of text for subsequent processing. Therefore, a large number of approaches to TLE have been published in the literature over the last few decades but most of them suffer from the inward structure of the documents pages which includes mainly the skewedness of text-lines, uneven inter line space and word gaps and irregular paragraph starting.

As the TLE in free-style environment is a challenging task, and hence many researchers have put their best effort to come up with some good solutions to this problem since few decades. It has been seen that the conventional approaches like Hough transform, projection profile, component grouping are not adequate for all types of documents due to the simplicity of these methods. The current trend in this regard, thus focuses on the learning-based methods. The learning-based methods generally use Artificial Intelligence (AI) in order to learn the significant features from a given dataset. Deep learning discovers a rich feature set through hierarchical models that actually learn probability distribution from the data encountered in particular applications. Applying a deep learning-based approach for TLE thus allows the model to learn the required features of its own, and with this we would intend to explore the task of TLE in a whole new perspective previously undiscovered. In this context, we use Generative Adversarial Network (GAN) (Goodfellow et al., 2014) to visualize

[*] Corresponding author.
E-mail addresses: berasuman007@gmail.com (S. Kumar Bera), ajith.abraham@ieee.org (A. Abraham).

https://doi.org/10.1016/j.eswa.2019.112916
0957-4174/© 2019 Elsevier Ltd. All rights reserved.

the problem of TLE in a different dimension, i.e. as a generative modeling problem. We frame TLE as an image-to-image translation task, where the model can impose the text-line separators in an input handwritten document to produce the desired output.

Multiple problems in image processing, computer vision and computer graphics have been about translating an input image into a corresponding output image using various transformations that include images, speech signals, or text data. In the recent past, deep discriminative models (Krizhevsky, Sutskever, & Hinton, 2012) are formed that mainly focused on supervised learning and mapped a feature-rich, sensory input to a class label. These are mainly based on the back-propagation algorithm that propagates information through the hidden layers, using piecewise linear units, and generally have a well-behaved gradient. Deep generative models have much less prominence, as it encounters severe problems of approximating many incomprehensible probabilistic computations that are generally found in maximum likelihood and similar strategies, where leveraging the linear units to fit into a generative context is extremely difficult.

GAN is now considered as one of the predominant models to learn generative model from complex real-world data. GANs generally use a generator to synthesize some semantically meaningful data matrices from some random signal distributions, and a discriminator to separate between the real and fake distributions. The generator is pitted against the adversary, the discriminator where each tries to out-do each another, and thus the generator model improves enough to mimic indistinguishable real data sample and the discriminator develops a keen eye on segregating the data generated by the generator and the real data samples. Generally, the training procedure continues till the generator wins the adversarial game, i.e. the discriminator is completely outperformed and has to make random guesses whether an image is real or fake. GAN has been successfully applied in many fields as image editing (Isola, Zhu, Zhou, & Efros, 2017; Wang, Wang, Xu, & Tao, 2017), image generation (Chen et al., 2016; Nguyen, Clune, Bengio, Dosovitskiy, & Yosinski, 2017), video prediction (Liang, Lee, Dai, & Xing, 2017) and multiple other tasks. The key contributions of the work can be summarized as:

i. GAN based architecture is used for the first time to extract the text lines from unconstrained handwritten document images.
ii. Two architectures of the generator namely U-Net and Encoder-Decoder, and PatchGAN architecture for the discriminator have been explored.
iii. Superiority of U-Net architecture over Encoder-Decoder framework has been shown.
iv. Effect of three different kinds of loss functions - GAN loss, L1 loss and L2 loss have been tested. Also a suitable merger of the 3 losses have been shown which outperforms some state-of-the-art methods.
v. Impressive outcomes are observed when the models have been evaluated on two standard datasets, called HIT-MW and ICDAR 2013 handwritten segmentation contest dataset.

The rest of the paper is organized as follows: The following section briefly describes the existing methods related to TLE, whereas Section 3 gives the overview of GANs and its variants. Section 4 presents the proposed methodology, followed by experimental results and discussion in Section 5. Finally, we conclude the paper in Section 6, where we also mention the future scope.

2. Related works

Till date, a lot of works have been proposed in the literature for TLE. In this section, we look at the brief history of TLE methods. The existing TLE methods can be broadly classified into a few categories - Hough transform based methods, projection profile based methods, smearing based methods, grouping based methods and learning based methods.

Hough transformation based methods are very powerful techniques to hypothesize the text-lines (horizontal or skewed) where most of the pixels are located in a document page. But the problem with this method is that, it is very time consuming when we deal with large set of inputs. The subsequent researches in this regard are thus focused on choosing the most relevant points for the voting procedure of Hough transform. Likforman-Sulem, Hanimyan, and Faure (1995) have used a hypothesis-validation strategy in an iterative way till the end of extractions where a text-line is imagined first depending on the best alignment of connected components (CCs) in the Hough domain and then its validation is tested using the contextual information in image domain. A *natural learning algorithm* based on the Hough transform is exploited to extract handwritten text-lines by Pu, Shi, and others (1998), where the Hough domain depends on the minima points of the CCs. Louloudis, Gatos, Pratikakis, and Halatsis (2006) have used a block-based Hough transform technique where the CC space is split into three subsets and each of the CCs is split into equal width and subsequently their Center Of Gravity (COG) helps in voting for Hough domain.

Vertical projection profile is considered as the most easy way (Shafait, Keysers, & Breuel, 2008; Shapiro, Gluhchev, & Sgurev, 1993) to extract text-lines from a document page with horizontal lines each having sufficient words. But it cannot achieve satisfactory results for multi-skewed and overlapping text-lines. To get the more smoothen vertical profile curve, several algorithms are designed by using different approaches like number of text pixels, black-white transitions (Marti & Bunke, 2001) or CCs. Manmatha and Srimal, 1999) have used Gaussian filter to smooth the curve by eliminating the local maxima. In Wong, Casey and Wahl (1982), the authors have partitioned the page into vertical column strips so that the curved lines break up into nearly straight lines, and then they have used vertical projection profile analysis. One of the major parameters for this process is the width of partitioning.

Run-length smoothing algorithm (RLSA) (Wong et al., 1982) is nothing but a smearing algorithm that actually fills up the blank space horizontally between two consecutive black (text) based on a certain threshold. Fuzzy RLSA proposed in Shi and Govindaraju (2004) is an extension of RLSA, where each entry in fuzzy run length matrix corresponds to the maximal extend of the background along the horizontal direction. DUTH-ARLSA proposed in Gatos, Antonacopoulos, and Stamatopoulos (2007) is based on an adaptive RLSA (ARLSA) that sets an additional smoothing constraint with respect to the geometrical properties of neighboring CCs. Malakar et al. (2012) has used the concept of spiral RLSA to detect the text lines in complex documents. Though the above-mentioned smearing based algorithms provide good results in most of the cases, but it fails in case of skewed text-lines.

In bottom up grouping strategy, the primitive components are clustered based on the positional relationship in order to achieve text-line alignments. In case of touching text-lines, choosing neighbors and factual alignment of each component is a critical issue. Some of these issues are taken care in Likforman-Sulem and Faure (1994) by applying a perceptual grouping scheme in an iterative way. Koo and Cho (2012) have considered it as a grouping problem of CCs and developed a cost function to minimize the fitting error of each text-line and the distances between two text-lines to extract the text-lines from handwritten Chinese documents. This method fails while handling Indic script documents containing text in cursive nature. In Ryu, Koo and Cho (2014), the authors have modified the method, but still it suffers from the merge of very close neighboring text-lines or a text-line with few number of components. In spite of the various challenges, some novel advancements (Du, Pan, & Bui, 2009b,a), (Li, Zheng, Doermann, & Jaeger, 2008), (Chen, Hong, & Chuang, 2012) have provided impressive results for multi-script documents, even in noisy environments but the computational complexity of the methods remains high. Jamuna and Haribabu (2015) developed the energy minimization framework to group the CCs where they have used two classifiers; one for text pixels and another one for non-text pixels. Basu, Chaudhuri, Kundu, Nasipuri, and Basu, (2007) have used the hypothetical water flows, from both left and right sides of the image frame where the stripes of un-wetted areas identifies the text lines. This is extended in piece-wise Water-flow technique by Sarkar et al. (2009).

The above-mentioned techniques are mostly rule based methods where they suffer from the inward structure of the documents pages. This includes mainly the skewedness of text-lines, uneven inter line space and word gaps and irregular paragraph starting. Recently, the trend is shifting towards using learning based methods. In Renton, Chatelain, Adam, Kermorvant and Paquet (2018), authors have proposed a method to segment text-lines based on an X-height labeling to provide representation of the text-lines and a Fully Convolutional Network (FCN) which is designed using the concept of dilated convolutions. Observing the advantages of learning based methods for different image processing work, in this paper, we attempt to apply a deep learning based method to solve the problem of TLE from handwritten document images. For this we have used GAN based architectures - U-Net architecture for generator and Patch GAN architecture for discriminator. We have also used a combination of three different losses, namely GAN loss, L1 loss and L2 loss.

The proposed text-line extraction method can be a useful application towards betterment of a generalized DAR system in many ways. Some of the important applications are briefly mentioned hereafter. In developing a comprehensive OCR system which can be applicable even if the input image is noisy, skewed, degraded and moreover for a multi-language environment. Our TLE method may also be a crucial part for the subsequent stages in a DAR system like de-warping, perspective distortion correction, word recognition, word spotting, script recognition and in general, recognition and indexing. In addition to these, it can be applied for any complex documents in free style environment like touching texts and overlapping texts or in case of sparse documents. Most importantly it can be useful for those complex images where the general rule based methods may fail to extract the text-lines.

3. GAN and its variants

In this section, we look at the basic GAN and its variants that are dedicated to removing training instability and improving the generative performance of the model. GAN provides an outstanding framework for learning generative models, which encapsulates probability distribution over predetermined real-world data. Model is easily trained by updating the generator and discriminator sub-networks using backpropagation algorithm which also results in better outcome in various generative tasks compared to other models.

In the GAN architecture, we have a generator G and discriminator D, which are trained in an adversarial manner as the generator is trained to generate realistic images from noise input z, and the discriminator is to differentiate between the real images x and those produced by the generator G(x). Using the feedback from the D, generator and discriminator losses are calculated and G learns to replicate real valued data. GAN, first proposed in Goodfellow et al. (2014), is basically a 2-player minimax game between G and D. G and D (use equation editor consistently) are two neural networks competing against each other in order to improve itself and the solution is a Nash Equilibrium.

Given some random noise, the data are assumed to be generated by a deterministic function of that noise. We can represent the generative process as $z \sim p_g(z)$, $X \sim G(z)$, where z is some random noisy sample, and $p_g(z)$ denotes the distribution of z. G is actually a neural network which takes the sample z as input and produces data X. GANs are motivated to use likelihood free algorithms (Marin, Pudlo, Robert, & Ryder, 2012), methods which assume that one can only sample from the model. Likelihood-free algorithms are based on the idea of learning from comparison (Gretton, Borgwardt, Rasch, Schölkopf, & Smola, 2012; Rubin, 1984), by analyzing differences between the generated samples from the model and those from the true data distribution, i.e. real-world samples. D is used to distinguish between the generated sample $G(z) \sim p_g(G(z))$ and the true data sample $x \sim p_{data}(x)$. So, D takes data $\mathbf{x}$ as input (either generated samples from the model or data points from the dataset), and it calculates the probability that $\mathbf{x}$ came from the true data. The minimax objective, i.e. the value function as described in Goodfellow et al. (2014) is mentioned in Eq. (1).

$$\min_G \max_D \mathbb{E}_{x \sim p_{data}}[\log D(x)] + \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z))] \quad (1)$$

This optimization problem is bi-level; it requires a minima solution with respect to generative parameters and a maxima solution with respect to discriminative parameters. This is addressed by alternatively optimizing the generator and the discriminator. The corresponding optimization goal for the discriminator and the generator are given in Eqs. (2) and (3) respectively.

$$\max_D \mathbb{E}_{x \sim p_{data}}[\log D(x)] + \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z))] \quad (2)$$

$$\min_G \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z))] \quad (3)$$

When GAN was introduced in Goodfellow (2016), training the generator was equivalent to minimizing the Jensen-Shannon divergence between the generated distribution and the real data distribution. But it easily resulted in a vanishing gradient problem. As, optimizing the minimax problem was difficult and often unstable, the non-saturating heuristic objective function was introduced in Goodfellow et al. (2014) (i.e. ‘ – log D’ mechanism) to replace the minimax objective function that was previously used to penalize the generator. In Salimans et al. (2016), authors have introduced network architectures (DCGAN) and proposed different heuristic tricks as virtual batch normalization, one side smoothing, feature matching, etc. to improve training accuracy. Least squares GAN (Mao et al., 2016) has improved training accuracy by deploying different kinds of training accuracies which partly increased training stability but still required a great deal of hyper parameter optimization.

DCGAN (Radford, Metz, & Chintala, 2015) is one of most successful network designs that was implemented based on GAN, and this architecture is base for many recent architectures. The DCGAN architecture consists of convolutional layers only and uses convolutional strides for down sampling and also transposes convolution in up sampling. Conditional GAN (cGAN) which was introduced (Mirza & Osindero, 2014), was to act like a conditional model, as both the generator and discriminator networks are conditioned on some information y. For the generator, y is combined with $p_z(z)$ to from hidden representations with some added flexibilities, while in discriminator, it is directly fed along with the input x. The resulting objective function of the minimax game as given in Mirza and Osindero (2014) is shown in Eq. (4).

$$\min_G \max_D V(D,G) = \mathbb{E}_{x \sim p_{data}}[\log D(x|y)] + \mathbb{E}_{x \sim p_z}[\log(1 - D(G(z|y))] \quad (4)$$

Pix2Pix was first developed by Isola et al. (2017) for image-to-image translation for paired images, whose model was based

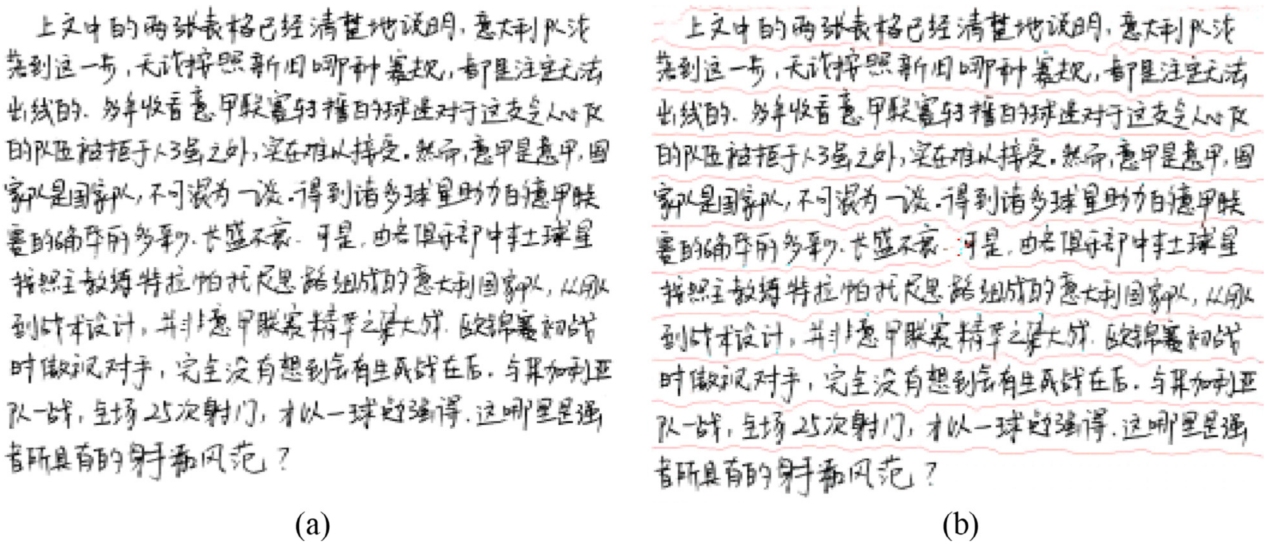


Fig. 1. (a) Input image, and (b) corresponding output image.

on cGAN, to learn the mapping between an input image and an output image. Image-to-image translation is transforming an image from one domain to an image in other domain, like black-and-white images to color images, aerial to map, edges to photo, day scene to night scene, etc.

This image-to-image translation is adopted in this paper to perform TLE from handwritten documents. The qualities of the input images, i.e. the handwritten documents are captured de facto, like CCs and vertical projection by the proposed model and an output image with separator lines is produced as the generated output. The model self-learns important factors of text-lines such as inter-line distance, handwriting patterns, continuity of text, and the paragraph separations. The ultimate goal here is to make use of the generative prowess of the GAN architecture and use it to perform a script independent TLE method that can work on any handwritten documents.

4. Proposed method

Acknowledging the dominance of deep learning based models in the field of computer vision, here we explore GAN based models to extract text-lines from handwritten document images. In this paper, the problem of TLE is visualized as an image-to-image translation task. Fig. 1 represents both the input image and its corresponding target image. Process of generating target image is described in Section 4.1. The task of TLE is described herewith as an image-to-image translation task where the model needs to learn the mapping of the red separator lines in the output image, given an input image. A red separator line separates two text-lines from one another accurately. The GAN architecture used for the present work is shown in Fig. 2.

4.1. Architectures used

Our model is inspired from the Pix2Pix model for paired image-to-image translation (Isola et al., 2017). The Pix2Pix model is based on cGAN (Mirza & Osindero, 2014). It contains two networks – the generator and the discriminator. Theoretically, in the said paper, translation is stated between two domains of images if they maintain the similar structure. Here, the input and output images have exactly the same structure with the addition of the separator lines. L1 loss along with the normal GAN losses are considered in the Pix2Pix model in Isola et al. (2017), where, L1 loss prevents GAN from producing completely new results, as the output image is related with the input image, while, the GAN loss accounts for accurate, non-blurry translation of the image.

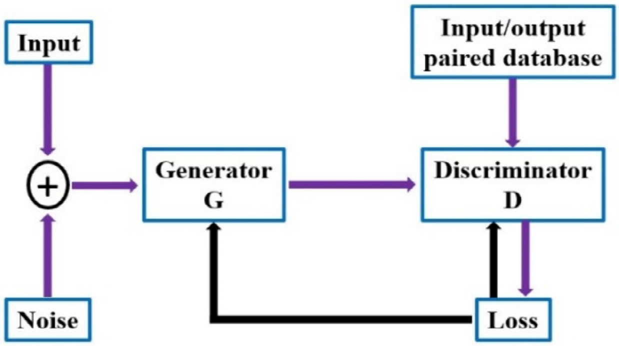


Fig. 2. Basic GAN architecture.

$$L_{L1}(G) = E_{x,y,z}[\|(y - G(x,z))\|_1] \quad (5)$$

In this paper, we have explored the Encoder-Decoder architecture and the U-Net architecture, as an improvement over the former architecture, for the generator. In the Encoder-Decoder architecture, the generator takes an input and tries to reduce it with a series of encoders, which encode into a smaller representation and the decode layers reverse the action of the encoder layers, to get the output. The encode layers contain convolution layers, whereas the decode layers contain deconvolution layers. The U-Net is an Encoder-Decoder architecture with skip connections. The outputs from the encoder are concatenated with their mirrored counterparts in the decoder. These skip connections when included, prevent network to be bottlenecked. The skip connections also give the network an option of bypassing the encoding/decoding part decisively.

The discriminator architecture is a Deep CNN (DCNN) network applying the concept of PatchGAN, i.e. the scores of the Discriminator is calculated in patches of the output image and an average of the scores is considered as the final output. This ensures the image has a higher and uniform resolution. The generator architectures are shown in Figs. 3 and 4, and the discriminator architecture is shown in Fig. 5.

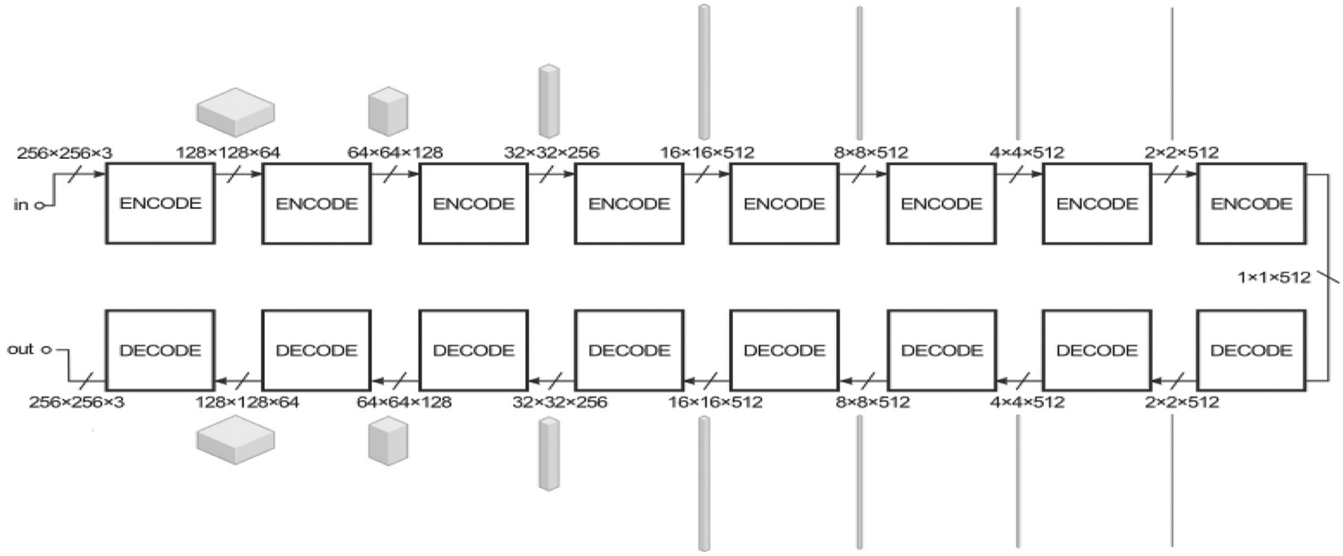


Fig. 3. Overview of the Encoder-Decoder framework.

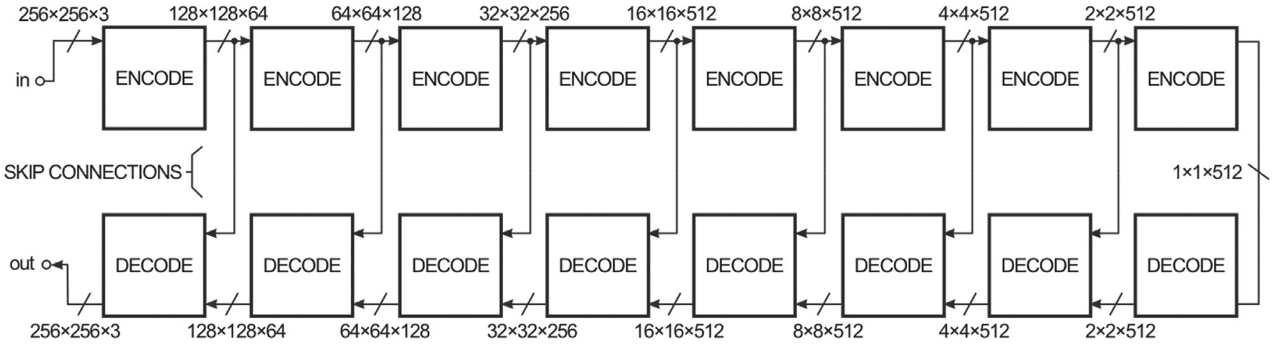


Fig. 4. Overview of the U-Net architecture (Encoder-Decoder architecture with skip connections).

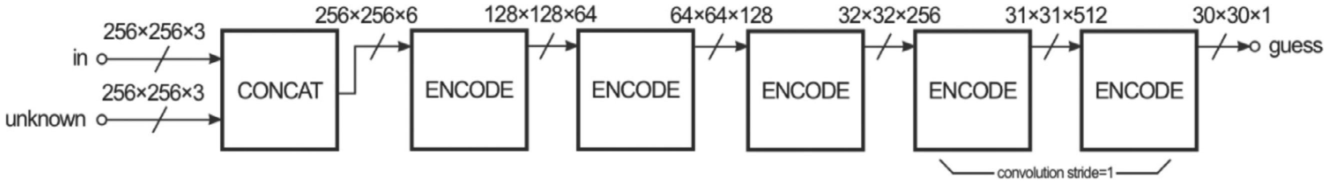
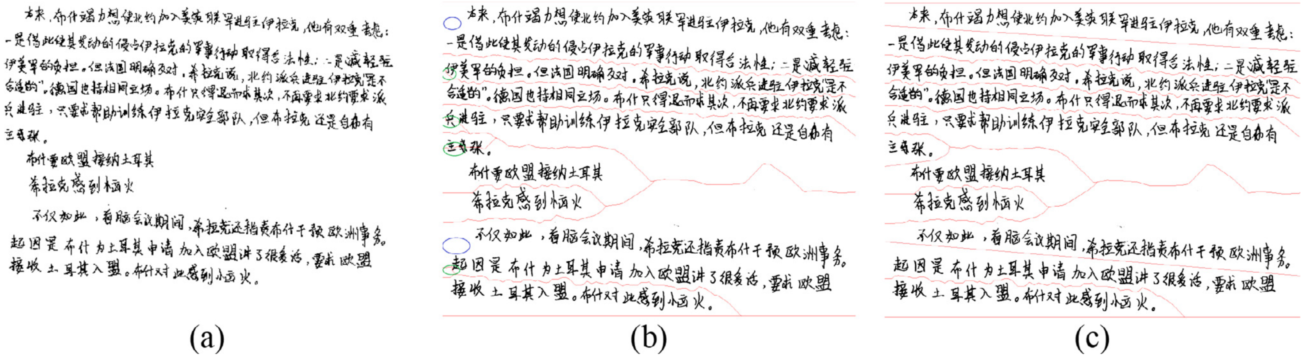


Fig. 5. Architecture of the discriminator.

As shown in Fig. 3, the Encoder-Decoder architecture takes an input image of size $256 \times 256 \times 3$. The value 3 resembles the three-color channels of an image i.e. red, green and blue. After a series of encode layers (convolution, batch normalization, ReLU), the model gives a representation of the image in the form of a vector of size $1 \times 1 \times 512$. This vector is fed into the decoder framework which applies a series of decode layers (deconvolution, batch normalization, ReLU), and finally the decoder framework outputs a generated image of the same size as that of the input image i.e. $256 \times 256 \times 3$. Also, the first and the last layers in the Encoder-Decoder framework do not have the batch normalization, and dropout (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014) has been used in the middle layers in both the encoder and decoder frameworks to prevent overfitting.

The U-Net architecture used in our model is shown in Fig. 4. The skip connections connect the encode layers to the decode layers and are shown using arrows in Fig. 4. It helps the proper flow of information across the bottleneck from the encoder to the decoder. Other details are same as that of the Encoder-Decoder architecture.

The discriminator architecture shown in Fig. 5 is actually a DCNN containing 5 encode layers, i.e. 5 convolution layers. The input to this architecture is an image (from the training set) and its corresponding generated image. These two images are concatenated and then fed to the series of encode layers which produce an output vector of size $30 \times 30 \times 1$. This vector consists of 0's and 1's, which represent whether the corresponding patch is fake or real respectively. The average of all the values is considered to decide the overall image as real or fake as in PatchGAN.



本来，布什竭力想使北约加入美英联军进驻伊拉克，他有双重考虑：
一是借此使美英发动的侵略伊拉克的军事行动取得合法性；二是减轻驻
伊美军的负担。但法国明确反对，希拉克说，北约派兵进驻伊拉克是不
合适的"。德国也持相同立场。布什只得退而求其次，不再要求北约要求派
兵进驻，只要求帮助训练伊拉克安全部队，但希拉克还是自有
主张。

布什要欧盟接纳土耳其
希拉克感到恼火

不仅如此，首脑会议期间，希拉克还指责布什干预欧洲事务。
起因是布什为土耳其申请加入欧盟讲了很多话，要求欧盟
接收土耳其入盟。布什对此感到恼火。

(a) (b) (c)

Fig. 6. (a) Sample document page, (b) Text-lines are separated using the method (Saabni et al., 2014); over and under segmentation cases are represented by green and purple ellipses respectively, (c) Accurately generated line separators after manually corrected error cases to prepare the corresponding GT image.

4.2. Objective

The objective function of cGAN can be expressed as Eq. (6) where G tries to minimize the objective against an adversarial D that tries to maximize it.

$$L_{cGAN}\ (G,\ D)\ = E_{x,y}[log\ D(x,\ y)] +\ E_{x,z}[log(1\ -\ D(x,\ G(x,\ z)))], \quad (6)$$

$$G^{*} = \arg min_{G} max_{D}\, L_{cGAN}\ (G,\ D). \quad (7)$$

Previous approaches have found it beneficial to mix the GAN losses with some traditional losses such as L1 loss (Isola et al., 2017). So, in this work, we apply L1 loss and L2 loss as two additional losses apart from cGAN losses. L1-norm is also known as least absolute deviations (LAD), least absolute errors (LAE). L1-norm minimizes the sum of the absolute differences between the target values and the estimated values. L2-norm is known as least squares, and it minimizes the sum of the square of the differences between the target values and the estimated values. The discriminator's task remains the same, whereas the generator's task is not only to outwit the discriminator but also to stay near the ground truth in L1 and L2 senses. The mathematical representations for L1 and L2 losses are shown in Eq. (8) and Eq. (9) respectively. The final objective function of the G can be represented as shown in Eq. (10).

$$L_{L1}(G) = E_{x,y,z}[\|(y - G(x,z))\|_{1}] \quad (8)$$

$$L_{L2}(G) = E_{x,y,z}[\|(y - G(x,z))\|_{2}] \quad (9)$$

$$G^{*}\ = \arg min_{G}\ max_{D}\ \Big[\big(gan_{weight} *\ L_{cGAN}\ (G,\ D)\big) + \big(l_{weight} *\ \ (L_{L1}(G) + L_{L2}(G))\big)\Big] \quad (10)$$

where, gan_{weight} and l_{weight} are the corresponding ratios in which the GAN losses and the normalization losses are considered.

5. Experimental results and discussion

This section presents the dataset description and the performance of our proposed method. The comparison with different state-of-the-art methods is also followed subsequently. We have experimented the GAN based architectures on several conditions.

5.1. Database and ground-truth (GT) preparation

Our proposed method has been tested on two benchmark datasets. First one is the recent handwritten segmentation contest ICDAR 2013 Handwritten Segmentation Contest dataset (Stamatopoulos, Gatos, Louloudis, Pal, & Alaei, 2013) and second one is the HIT-MW dataset (Su, Zhang, & Guan, 2007), prepared by Harbin Institute of Technology. The ICDAR 2013 dataset consists of 150 binary images written in English, Greek and Bangla languages where each language contributes equal number of pages. The HIT-MW is a handwritten Chinese text dataset that consists of 853 images containing Chinese handwritten documents, with 8664 text-lines. According to the database, the images are scanned at 300 dpi and were binarized using Otsu's algorithm (Otsu, 2008) and saved as bmp images without any compression. As our deep learning based method requires paired images (e.g. original and GT image) for the training of proposed model, we have used a traditional method proposed by Saabni, Asi, and El-Sana (2014) for generating the initial level line separators in a document page. The failure cases of this method due to the irregular starting points of text lines as well as uneven word gap, are handled manually. Such a failure case and its corresponding correction are shown in Fig. 6. The red lines throughout the document page correspond the line separators between two consecutive text-lines.

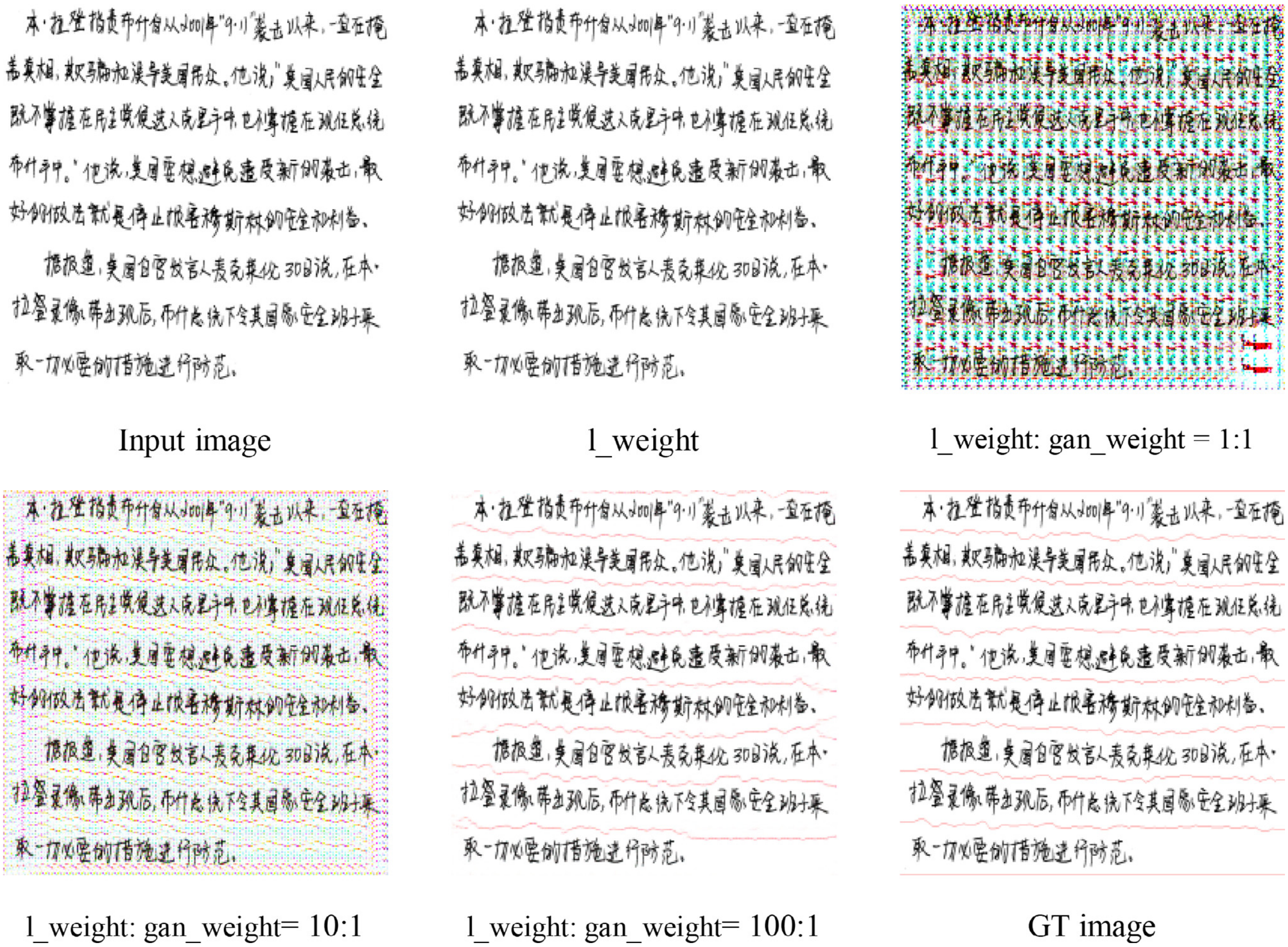
5.2. Preparation of training and testing set

Our models are trained on Quadro M4000, with 16GB RAM and 8GB GPU memory. The models are implemented in Python and TensorFlow (Abadi et al., 2016) is used as the deep learning framework. We have trained the models considering a batch size of 1, learning rate of 0.002 using Adam optimizer with 0.5 momentum. In encode and decode layers, dropout has been used with 0.5 probability. The models have been trained for 200 epochs. We have used 3-fold cross validation scheme for each dataset to evaluate our proposed method. In case of HIT-MW dataset, a total of 568 handwritten document pages are considered for training the model whereas for testing the model, we have taken 285 document pages. Similarly, for ICDAR2013 Handwritten Segmentation Contest dataset, a set of 100 and 50 document images (written in three different languages) are taken as the training set and test set respectively.

In our model, we have considered two hyper parameters – one for the weightage of the cGAN loss and another for the weightage of L1 or L2 loss, i.e. gan_weight and l_weight respectively. The performance of the system depends on these two parameters. We have conducted a few experiments by considering only L1 loss and cGAN loss as shown in Table 1. Fig. 7 displays the sample results of a particular image when evaluated with four combinations of different losses. It is noticed that the optimal ratio between the weightage of cGAN loss and L1 (or L2) loss is 1:100 and only L1 loss has given the same output as the input image. So, some weightage of cGAN loss is required. A ratio of 1:1 between cGAN

Table 1
Different criteria of L1 loss and cGAN loss.

l_weight	gan_weight	Loss
100	0	Only L1 loss considered.
100	100	Both L1 loss and cGAN loss given equal importance.
100	10	L1 loss and cGAN losses considered in the ratio of 10:1
100	1	L1 loss and cGAN losses considered in the ratio of 100:1


Input image

l_weight

l_weight: gan_weight = 1:1

l_weight: gan_weight= 10:1

l_weight: gan_weight= 100:1

GT image

Fig. 7. Resultant images by varying the hyper parameters for a sample image taken from HIT-MW dataset.

weightage and L1 weightage has resulted in an image having various color formation in an anomalous way, and a ratio of 1:10 has given relatively less color formation, but the red marks could be seen to distinguish the text-lines in spite of the image being unclear due to various other colors.

5.3. Why U-Net architectures?

We have implemented the Encoder-Decoder architecture against the U-Net architecture for comparison, and shown that U-Net architecture is an improvement over the Encoder-Decoder architecture. Some results are shown in Fig. 8, which show the outputs of the Encoder-Decoder and the U-Net architectures when cGAN, L1 and L2 losses are considered.

It is noticed in Fig. 8 that the Encoder-Decoder architecture does not work well for the task of TLE. Using Encoder-Decoder architecture, the L1+cGAN model seems to diverge as it produces exactly straight lines in a similar pattern. The L2+cGAN model is not able to get the exact mapping between the images; it also produces straight lines, but still the straight lines are dependent on the white spaces in the image. For the L1+L2+cGAN model, the representation is better than the previous two loss functions but still the model could not learn the mapping accurately. In U-Net architecture, skip connections have been used for the proper flow of information across the bottleneck of the GAN architecture. Because of this, the decoder output generates a better representation of the translated image. The U-Net architecture performs better than the Encoder-Decoder framework in the domain of the TLE and has been thus used in this paper.

5.4. Results

We have implemented two GAN based architectures namely Encoder-Decoder and U-Net, using a combination of three different losses, i.e. cGAN loss, L1 loss and L2 loss. We have implemented the U-Net architecture as the generator architecture and

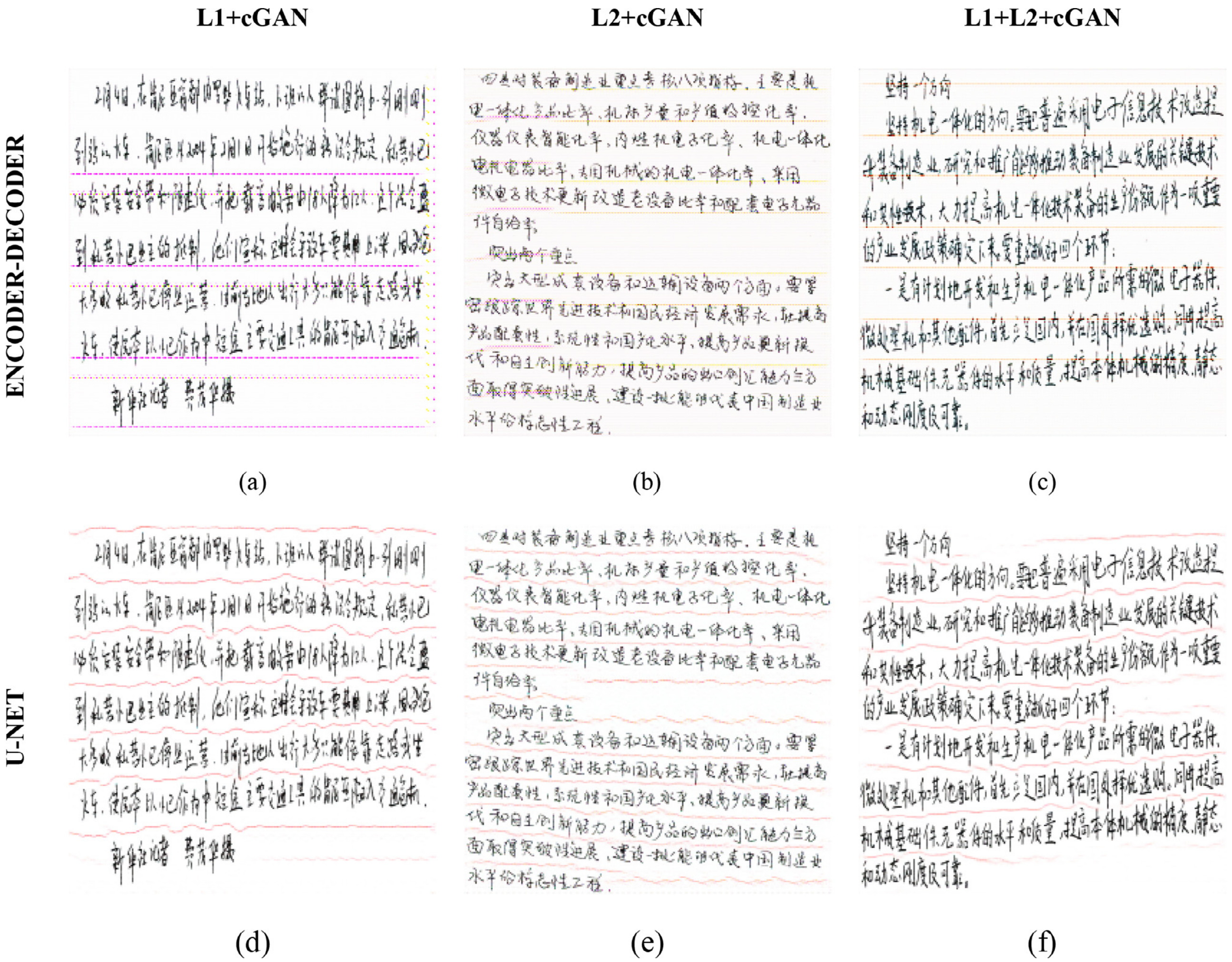


Fig. 8. Results of Encoder-Decoder and U-Net architectures for three sample images and different losses. (a–c) Results of Encoder-Decoder architecture, and (d–f) Results of U-Net architecture.

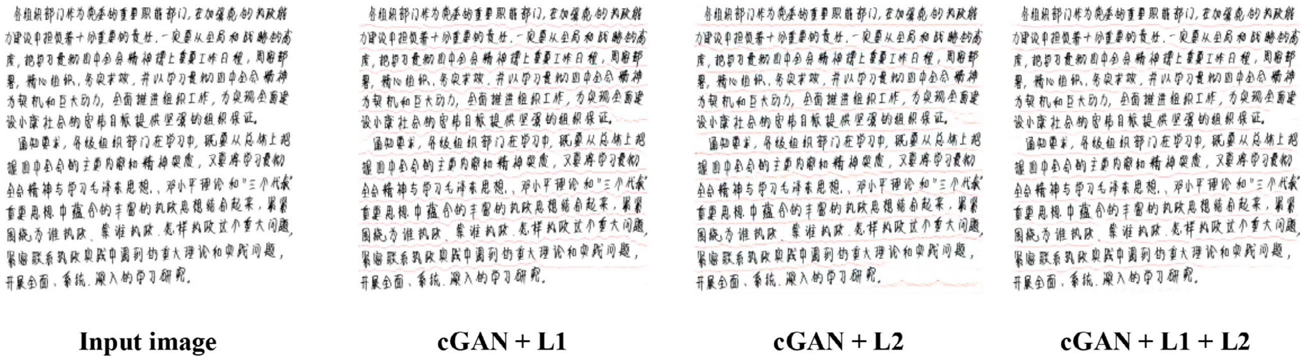


Fig. 9. Examples showing the results for various combination of losses (using U-Net architecture).

considered a PatchGAN in the discriminator architecture with a patch size of 70 × 70. We have considered three different combinations of the three losses to evaluate our system – cGAN + L1, cGAN + L2 and cGAN + L1 + L2. By illustrating some output images in Fig. 9, we show the performance of the model for each of the three combinations of losses. Also, the variation of the losses during the training period can be visualized with the help of the graphs shown in Fig. 10.

We have compared our method with other methods over the two said databases. The results are shown in Tables 2 and 3 (for HIT-MW and ICDAR databases respectively). The detection rate (*DR*), recognition accuracy (*RA*) and error rate (*ER*) and

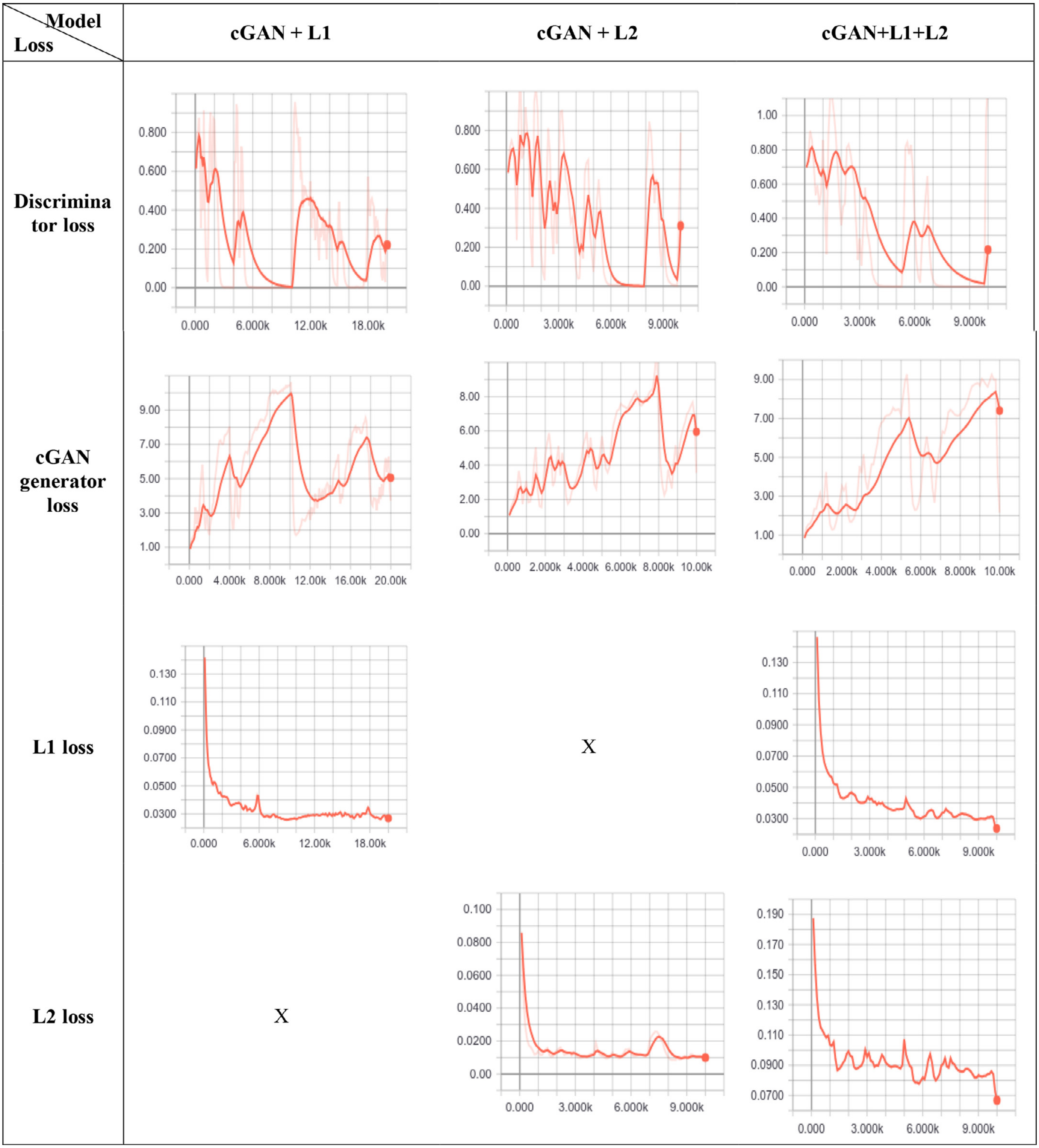


Fig. 10. Graphs showing the variation of losses during training (using U-Net architecture).

F-measure (FM) are defined as

$$DR = \frac{o2o}{N}, \quad RA = \frac{o2o}{M}, \quad ER = 1 - RA, \quad FM = \frac{2DR \cdot RA}{DR + RA} \qquad (15)$$

where o2o is one-to-one mapping, M and N are the number of text-lines in detected resultant and GT images respectively.

Fig. 10 interprets that in case of L1+L2+cGAN losses, the training becomes quite accurate, as we can see from the nature of the graphs. We also see that how L1 loss and L2 loss decrease gradually over time, thus making the system stable, and producing accurate image translation.

The results provided in Table II and Table III imply that our method achieves highest accuracy for the said datasets. So, we can

Table 2
Experimental results on HIT-MW dataset (using U-Net architecture).

Methods			# Images	DR (%)	RA (%)	ER (%)	FM (%)
Energy Minimization Framework (Koo & Cho, 2012)			853	99.52	99.68	0.32	99.59
Distance Metric Learning (Yin & Liu, 2009)			803	98.02	97.53	2.47	97.77
Mumford–Shah model (Du et al., 2009b)			853	95.92	96.86	3.14	96.38
MST Clustering with Learned Metric (Yin & Liu, 2007)			803	95.02	–	–	–
Modified Piece-wise Projection (Pal & Datta, 2003)			803	92.07	92.28	7.72	92.17
Docstrum Method (O'Gorman, 2009)			803	65.38	55.62	44.38	60.10
Proposed	cGAN+L1	Fold#1		98.89	99.57	0.43	99.22
		Fold#2		99.19	99.60	0.40	99.39
		Fold#3		99.43	99.69	0.31	99.55
		Average		99.17	99.62	0.38	99.38
	cGAN+L2	Fold#1		98.92	99.48	0.52	99.19
		Fold#2		99.32	99.51	0.49	99.41
		Fold#3		98.64	99.72	0.28	99.17
		Average		98.96	99.57	0.43	99.25
	cGAN+L1+L2	Fold#1		99.59	99.61	0.39	99.59
		Fold#2		99.61	99.72	0.28	99.66
		Fold#3		99.57	99.74	0.26	99.65
		Average		**99.59**	**99.69**	**0.31**	**99.63**

Table 3
Experimental results on ICDAR 2013 Handwritten Segmentation Contest dataset (using U-net architecture).

Methods			# Images	DR (%)	RA (%)	ER (%)	FM (%)
TEI(SoA)			150	97.77	96.82	3.18	97.30
CUBS				97.96	96.94	3.06	97.45
GOLESTAN				98.23	98.34	1.66	98.28
NUS				98.34	98.49	1.51	98.41
INMC				98.68	98.64	1.36	98.66
Proposed	cGAN+L1	Fold#1		98.32	97.56	2.44	97.93
		Fold#2		97.89	98.47	1.53	98.17
		Fold#3		98.52	98.49	1.51	98.50
		Average		98.24	98.27	1.73	98.20
	cGAN+L2	Fold#1		98.45	97.53	2.47	97.98
		Fold#2		97.88	98.29	1.71	98.08
		Fold#3		98.48	98.57	1.43	98.52
		Average		98.27	98.19	1.81	98.19
	cGAN+L1+L2	Fold#1		98.65	98.65	1.35	98.65
		Fold#2		98.70	98.66	1.34	98.67
		Fold#3		98.72	98.69	1.31	98.70
		Average		**98.69**	**98.66**	**1.34**	**98.67**

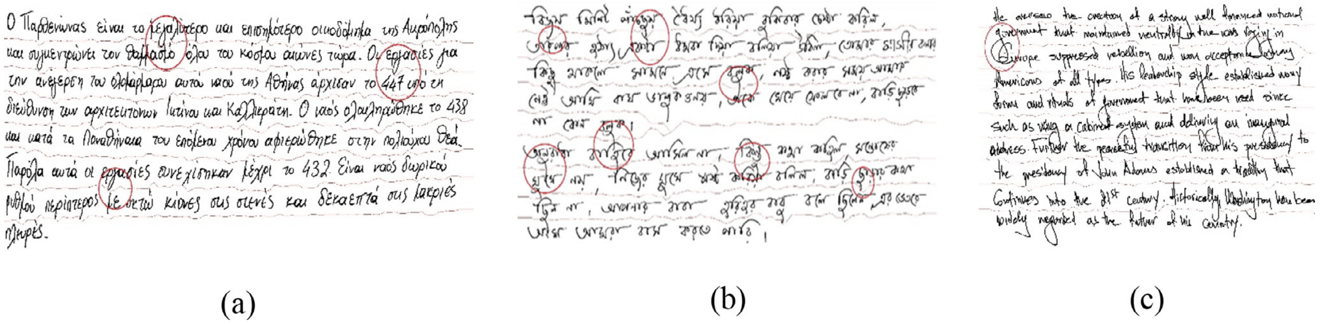


Fig. 11. Output of our TLE method on three sample images taken from ICDAR 2013 handwritten competition dataset. (a) Greek document, (b) Bangla document, and (c) English document. Overlapping and touching components are shown in circles.

conclude that GAN based architecture (using U-Net architecture) performs quite well for the task of TLE, posed as an image-to-image translation problem. The outputs of the three sample images from ICDAR 2013 handwritten segmentation competition dataset are displayed in Fig. 11. We have used three samples having touching lines and overlapping characters. The outputs are very promising to prove the robustness of our model.

6. Conclusion and future work

GANs have been a proven deep learning architecture to learn probability distributions and mimic the same including all generative tasks. In this paper, we have explored GAN based architectures for TLE from handwritten document images that have been shown to outperform some state-of-the-art TLE methods when we have evaluated the same on the HIT-MW dataset and ICDAR 2013 Handwritten Segmentation Contest dataset. We have achieved a maximum accuracy of 99.63% F-measure in HIT-MW dataset and 98.67% F-measure in ICDAR dataset on 3-fold cross validation over entire datasets. Extensive testing has been performed on how the behavior of the L1 and L2 loss functions in the current domain correlates to improving the performance of the model along with the preconceived cGAN loss. The tested GAN model is particularly sensitive to input hyper parameters and a thorough study of the same using the U-Net architecture has been carried out. U-Net architecture is shown to perform better than the Encoder-Decoder architecture on the same loss functions due to the presence of skip connections in the former. In the future, we can use this model to explore other datasets on TLE, and develop a loss function specific to the domain of TLE rather than generalized loss functions. We also plan to extend these GAN based architectures to other domains of DAR as it remains vastly unexplored and GANs show excellent promises to better understandings and approaches to these unexplored domains.

Declaration of Competing Interest

There is no conflict of interest.

Credit authorship contribution statement

Soumyadeep Kundu: Conceptualization, Data curation, Formal analysis, Resources, Software, Methodology, Writing - original draft, Writing - review & editing. **Sayantan Paul:** Conceptualization, Data curation, Formal analysis, Resources, Software, Methodology, Writing - original draft, Writing - review & editing. **Suman Kumar Bera:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Writing - original draft, Writing - review & editing. **Ajith Abraham:** Conceptualization, Supervision, Visualization, Writing - original draft, Writing - review & editing. **Ram Sarkar:** Conceptualization, Investigation, Methodology, Supervision, Writing - original draft, Writing - review & editing.

Acknowledgment

This work is partially supported by the CMATER research laboratory of the Computer Science and Engineering Department, Jadavpur University, India, and the co-author Ram Sarkar is partially funded by DST grant (EMR/2016/007213).

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016). TensorFlow : A system for large-scale machine learning this paper is included in the proceedings of the tensorflow : A system for large-scale machine learning. *Proc 12th USENIX conference on operating systems design and implementation* https://doi.org/10.1126/science.aab4113.4.

Basu, S., Chaudhuri, C., Kundu, M., Nasipuri, M., & Basu, D. K. (2007). Text line extraction from multi-skewed handwritten documents. *Pattern Recognition, 40*(6), 1825–1839.

Bera, S. K., Kar, R., Saha, S., Chakrabarty, A., Lahiri, S., Malakar, S., et al. (2017). A one-pass approach for slope and slant estimation of tri-script handwritten words. *Journal of Intelligent Systems*. https://doi.org/10.1515/jisys-2018-0105.

Bhowmik, S., Sarkar, R., Nasipuri, M., & Doermann, D. (2018). Text and non-text separation in offline document images: A survey. *International Journal on Document Analysis and Recognition (IJDAR), 21*(1–2), 1–20.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems* (pp. 2172–2180).

Chen, Y.-L., Hong, Z.-W., & Chuang, C.-H. (2012). A knowledge-based system for extracting text-lines from mixed and overlapping text/graphics compound document images. *Expert Systems with Applications, 39*(1), 494–507.

Das, S., Singh, P. K., Bhowmik, S., Sarkar, R., & Nasipuri, M. (2016). A harmony search based wrapper feature selection method for Holistic Bangla word recognition. *Procedia Computer Science, 89*, 395–403.

Du, X., Pan, W., & Bui, T. D. (2009a). Text line segmentation in handwritten documents using Mumford–Shah model. *Pattern Recognition, 42*(12), 3136–3145.

Du, X., Pan, W., & Bui, T. D. (2009b). Text line segmentation in handwritten documents using Mumford-Shah model. *Pattern Recognition*. https://doi.org/10.1016/j.patcog.2008.12.021.

Gatos, B., Antonacopoulos, A., & Stamatopoulos, N. (2007). Handwriting segmentation contest. *Icdar*. https://doi.org/10.1109/ICDAR.2007.4377122.

Goodfellow, I. (2016). Generative adversarial network. *NIPS* https://doi.org/10.1016/S1634-2143(05)44979-1.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial nets ian. *Mining of Massive Datasets: Second Edition*. https://doi.org/10.1017/CBO9781139924801.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A. (2012). A kernel two-sample test. *The Journal of Machine Learning Research, 13*(2012), 723–773.

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings - 30th IEEE conference on computer vision and pattern recognition, CVPR 2017* https://doi.org/10.1109/CVPR.2017.632.

Jamuna, M., Haribabu, S. (2015). Text-Line extraction and word spotting in a handwritten document.

Kar, R., Saha, S., Bera, S. K., Kavallieratou, E., Bhateja, V., & Sarkar, R. (2019). Novel approaches towards slope and slant correction for tri-script handwritten word images. *The Imaging Science Journal, 67*(3), 159–170.

Koo, H. Il, & Cho, N. I. (2012). Text-line extraction in handwritten Chinese documents based on an energy minimization framework. *IEEE Transactions on Image Processing, 21*(3), 1169–1175.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). 1 ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*. https://doi.org/10.1016/j.protcy.2014.09.007.

Li, Y., Zheng, Y., Doermann, D., & Jaeger, S. (2008). Script-independent text line segmentation in freestyle handwritten documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2007.70792.

Liang, X., Lee, L., Dai, W., & Xing, E. P. (2017). Dual motion GAN for future-flow embedded video prediction. In *Proceedings of the IEEE international conference on computer vision* https://doi.org/10.1109/ICCV.2017.194.

Likforman-Sulem, L., & Faure, C. (1994). Extracting text lines in handwritten documents by perceptual grouping. *Advances in Handwriting and Drawing: A Multidisciplinary Approach*, 117–135.

Likforman-Sulem, L., Hanimyan, A., & Faure, C. (1995). A Hough based algorithm for extracting text lines in handwritten documents. In *Proceedings of 3rd international conference on document analysis and recognition* https://doi.org/10.1109/ICDAR.1995.602017.

Louloudis, G., Gatos, B., Pratikakis, I., & Halatsis, C. (2006). A block-based Hough transform mapping for text line detection in handwritten documents. *10th international workshop on frontiers in handwriting recognition (IWFHR 2006)*.

Malakar, S., Halder, S., Sarkar, R., Das, N., Basu, S., & Nasipuri, M. (2012). Text line extraction from handwritten document pages using spiral run length smearing algorithm. In *2012 international conference on communications, devices and intelligent systems (CODIS)* (pp. 616–619). IEEE.

Manmatha, R., & Srimal, N. (1999). Scale space technique for word segmentation in handwritten documents. *SCALE-SPACE '99 proceedings of the second international conference on scale-space theories in computer vision* https://doi.org/10.1007/3-540-48236-9_3.

Mao, X., Li, Q., Xie, H., Lau, R.Y.K., Wang, Z., Smolley, S.P. (2016). Least squares GAN. *Arxiv*. https://doi.org/10.1109/ICCV.2017.304.

Marin, J. M., Pudlo, P., Robert, C. P., & Ryder, R. J. (2012). Approximate Bayesian computational methods. *Statistics and Computing*. https://doi.org/10.1007/s11222-011-9288-2.

Marti, U. V., & Bunke, H. (2001). On the influence of vocabulary size and language models in unconstrained handwritten text recognition. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*. https://doi.org/10.1109/ICDAR.2001.953795.

Mirza, M., Osindero, S. (2014). Conditional generative adversarial nets, 1–7. https://doi.org/10.1017/CBO9781139058452.

Nguyen, A., Clune, J., Bengio, Y., Dosovitskiy, A., & Yosinski, J. (2017). Plug and play generative networks: Conditional iterative generation of images in latent space. In *Proceedings - 30th IEEE conference on computer vision and pattern recognition, CVPR 2017* https://doi.org/10.1109/CVPR.2017.374.

O'Gorman, L. (2009). The document spectrum for page layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 15*(11), 1162–1173.

Otsu, N. (2008). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*. https://doi.org/10.1109/tsmc.1979.4310076.

Pal, U., & Datta, S. (2003). Segmentation of Bangla unconstrained handwritten text. In *Proceedings of the seventh international conference on document analysis and recognition: 2* (pp. 1128–1132).

Pu, Y., Shi, Z., others. (1998). A natural learning algorithm based on Hough transform for text lines extraction in handwritten documents.

Radford, A., Metz, L., Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks, 1–16. https://doi.org/10.1051/0004-6361/201527329.

Renton, G., Chatelain, C., Adam, S., Kermorvant, C., & Paquet, T. (2018). Handwritten text line segmentation using fully convolutional network. In *Proceedings of the international conference on document analysis and recognition, ICDAR* https://doi.org/10.1109/ICDAR.2017.321.

Rubin, D. B. (1984). Bayesianly justifiable and relevant frequency calculations for the applied statistician. *The Annals of Statistics*. https://doi.org/10.1214/aos/1176346785.

Ryu, J., Koo, H. Il, & Cho, N. I. (2014). Language-independent text-line extraction algorithm for handwritten documents. *IEEE Signal Processing Letters, 21*(9), 1115–1119.

Saabni, R., Asi, A., & El-Sana, J. (2014). Text line extraction for historical document images. *Pattern Recognition Letters, 35*, 23–33.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X. (2016). Improved techniques for training GANs, 1–10. arXiv:1504.01391.

Sarkar, R., Basu, S., Das, N., Mollah, A. F., Kundu, M., & Nasipuri, M. (2009). Line extraction from unconstraint handwritten document pages using piece-wise water-flow technique. In *IICAI* (pp. 1861–1872).

Shafait, F., Keysers, D., & Breuel, T. (2008). Performance evaluation and benchmarking of six-page segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2007.70837.

Shapiro, V., Gluhchev, G., & Sgurev, V. (1993). Handwritten document image segmentation and analysis. *Pattern Recognition Letters, 14*(1), 71–78.

Shi, Z., & Govindaraju, V. (2004). Line separation for complex document images using fuzzy runlength. In *first international workshop on document image analysis for libraries, 2004. Proceedings* (pp. 306–312). IEEE.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*. https://doi.org/10.1214/12-AOS1000.

Stamatopoulos, N., Gatos, B., Louloudis, G., Pal, U., & Alaei, A. (2013). ICDAR 2013 handwriting segmentation contest. In *2013 12th international conference on document analysis and recognition* (pp. 1402–1406). IEEE.

Su, T., Zhang, T., & Guan, D. (2007). Corpus-based HIT-MW database for offline recognition of general-purpose Chinese handwritten text. *International Journal on Document Analysis and Recognition*. https://doi.org/10.1007/s10032-006-0037-6.

Wang, C., Wang, C., Xu, C., & Tao, D. (2017). Tag disentangled generative adversarial networks for object image re-rendering. *IJCAI international joint conference on artificial intelligence*.

Wong, K. Y., Casey, R. G., & Wahl, F. M. (1982). Document analysis system. *IBM Journal of Research and Development, 26*(6), 647–656.

Yin, F., & Liu, C. L. (2007). Handwritten text line extraction based on minimum spanning tree clustering. In *2007 international conference on wavelet analysis and pattern recognition: 3* (pp. 1123–1128). IEEE.

Yin, F., & Liu, C. L. (2009). Handwritten Chinese text line segmentation by clustering with distance metric learning. *Pattern Recognition, 42*(12), 3146–3157.