Can Generative Adversarial Networks Teach Themselves Text Segmentation?

Mohammed Al-Rawi, Dena Bazazian, Ernest Valveny

Computer Vision Center, Universitat Autonoma de Barcelona, Spain

ms.alrawi@gmail, {dena.bazazian,ernest}@cvc.uab.es

Abstract

In the information age in which we live, text segmentation from scene images is a vital prerequisite task used in many text understanding applications. Text segmentation is a difficult problem because of the potentially vast variation in text and scene landscape. Moreover, systems that learn to perform text segmentation usually need non-trivial annotation efforts. We present in this work a novel unsupervised method to segment text at the pixel-level from scene images. The model we propose, which relies on generative adversarial neural networks, segments text intelligently; and does not therefore need to associate the scene image that contains the text to the ground-truth of the text. The main advantage is thus skipping the need to obtain the pixel-level annotation dataset, which is normally required in training powerful text segmentation models. The results are promising, and to the best of our knowledge, constitute the first step towards reliable unsupervised text segmentation. Our work opens a new research path in unsupervised text segmentation and poses many research questions with a lot of trends available for further improvement.

1. Introduction

There has been non-trivial efforts to bring machine intelligence to the level of human intelligence. Humans can learn in an unsupervised manner, and they can understand the world around them in several ways. The simplest example of this unsupervised intelligence is how humans have a good perception of text, as they can learn the alphabets, words and sentences, and can later detect text irrelevant to the text they viewed during basic supervised learning. In addition, humans can learn text written on blank background, yet, they can spot and detect text that appear in scene images without the need for any correspondence with what they have learned / taught and the scene images. For machines, and up to writing this paper, the learning is typically supervised by having some form of correspondence between the scene text images ground-truth data, which are usually marked as bounding-box or pixel-wise annotations. Because the ultimate aim of scene text tasks usually concludes to reading up the text, the pixel-level text segmentation can provide more accuracy to the task of text recognition in comparison with the bounding box detection approach. However, text segmentation based on supervised learning has two implications 1) non-trivial efforts have to be dedicated to annotate the data and 2) machines (or computer vision systems) will lack true intelligence compared to humans. This work aims at achieving a new level of machine intelligence by addressing the problem of text segmentation using deep neural networks in an unsupervised manner.

Text segmentation in scene images is the key step for end-to-end scene text recognition systems. The aim of text segmentation is to detect text at the pixel level, *i.e*. extracting all text pixels of an image. While this may seem a simple problem, previous works have shown otherwise; this is because text usually appear in unconstrained scene environments, with various text shape and formation, scale, orientation, font, color, and complex background. These variations can lead to endless scene text combinations. Text segmentation from scene images is therefore a difficult task and is considered a challenging computer vision problem.

We propose in this work a novel idea to segment text from scene images based on Generative Adversarial Networks (GANs). Figure 1 illustrates the supervised and unsupervised learning aimed at text segmentation. The contribution of this paper can be folded as follows; 1) investigate the use of (GANs) for text segmentation from scene images, 2) propose an unsupervised text segmentation approach, 3) compare text segmentation based on Deep Convolutional Neural Networks (DCNNs) and the proposed GAN method, 4) propose a method to handle the segmented text via CycleGANs in cases of dark text, and 5) propose a novel method, based on F_1 score, to measure the performance of text segmentation. The remainder of the article is organized as fol-

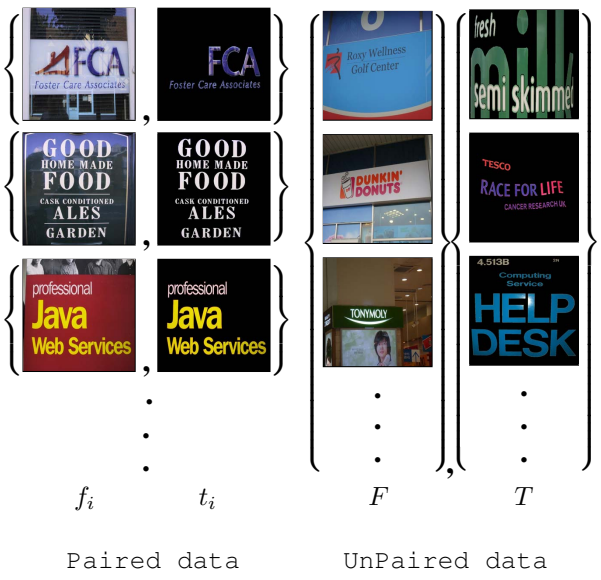

Figure 1. In supervised text segmentation learning (left), paired training data are considered, consisting of training examples $\{f_i, t_i\}_{i=1}^{N}$ containing source image f_i and its target (ground-truth) t_i, where the one-to-one correspondence between f_i and t_i exists. In unsupervised text segmentation learning (right), we consider unpaired training data, consisting of a source set $\{f_i\}_{i=1}^{N}$ ($f_i \in F$) and a target set $\{t_i\}_{i=1}^{N}$ ($t_i \in T$), with no information provided as to which source matches which target. Best viewed in color.

lows; the next section reviews the related works, Section 3 describes the GAN that we will use to perform text segmentation, then present the experimental results and conclusions in Sections 4 and 6, respectively.

2. Related Works

Text segmentation can be simply solved through thresholding of document images [10, 24]. Thresholding techniques, however, cannot be directly used to segment text in the scene images due to variations in the size and orientation of text that often appeases on a cluttered background. Nonetheless, some works have used thresholding techniques for segmenting text in scene images using a Gaussian Mixture Model to model color distributions that may have various foregrounds and background colors [21]. Drifting from thresholding techniques, text polarity and stroke width estimation based on gradient local correlation have been used in [2]. The approach in [27] extracted a binary mask from the image, by sampling the image in a row-by-row fashion such while testing each row for text existence, using a 1D adaptation of the MSER algorithm. Moreover, in [26] a multi-level MSER technique has been used to extract text candidates from scene images, based on four factors of stroke width, boundary curvature, character confidence and color constancy, which led to an improved segmentation. Recently, DCNNs have been implemented in semantic image segmentation [16, 4]. With regard to segmenting text via DCNNs, [25] performed text segmentation through three stages by performing extraction, refinement and classification.

Generative Adversarial Networks (GANs) [7] are magical machine learning systems that can be used for unsupervised machine learning and they are considered now the cutting edge of Artificial Intelligence. They are basically made up of two competing models that run in competition with one another and are able to capture and copy variations within a dataset, making them great tools for image generation and manipulation. GANs have been used to build models for a large number of applications, for example, cracking passwords [9], encryption [1], apparel and fashion industry [6], *etc*. A comparison of general semantic segmentation with GAN has been investigated in [18], but nothing relates to text segmentation. Furthermore, there are also some other works that have applied GANs in textual images, but not text segmentation, as in [20] which used GANs for Chinese calligraphy synthesis, and [28] which is to synthesize scene text images. To the best of our knowledge, the literature still lacks the use of GANs in text segmentation in an unsupervised manner, which is the main contribution of this work. The CycleGAN has been proposed as an innovative model that can be used for unsupervised Image-to-Image translation [30]. While text segmentation can be treated as a paired data problem, the unsupervised approach have great advantages, as aforementioned.

3. Methods

3.1. Supervised text segmentation using DCNNs

Semantic segmentation works by understanding an image at the pixel level, then assigning a label to every pixel in the image. Therefore, pixels with the same label should share certain characteristics. In this work, we use DeepLabV3+ [4], which is considered as one of the most promising semantic segmentation techniques used mainly for general object segmentation. Furthermore, DeepLabV3+ is an extended version of DeepLabV3 [5], but the robustness of DeepLabV3+ stems from applying several parallel atrous convolution, with different rates (called Atrous Spatial Pyramid Pooling, or ASPP) that capture the contextual information at multiple scales. This is very important for detecting text in scene images because in pixel-wise text segmentation techniques, the deconvolution layer will degrade some spatial information such as FCN [16]. The use of atrous convolution will assist preserve the spatial information, which is necessary to capture the scale and orientation that appears frequently in text. We will take into account that the results obtained using DeepLabV3+ as the

baseline to compare with the results we obtain from GANs.

3.2. Cycle-consistency GAN

The hypothesis of this work is that a CycleGAN can learn the mapping function between scene image and text image domains, F and T, given unpaired training samples and targets defined as $\{f_i\}_{i=1}^{N}$ and $\{t_i\}_{i=1}^{M}$, where $(f_i \in F)$ and $(t_i \in T)$. The model includes 1) two mappings $G : F \longrightarrow T$ and $H : T \longrightarrow F$; 2) two adversarial discriminators, D_F aims to distinguish between images f and translated images $H(t)$, and D_T aims to discriminate between images t and translated images $G(f)$; and 3) two identity mapping regularizers. The objective function should thus contain six terms: two adversarial losses for matching the distribution of generated images to the data distribution text in the target domain, two cycle consistency losses to prevent the learned mappings G and H from conflicting each other [30], and two identity losses. Hence, the objective function is given by:

$$\begin{aligned}\mathcal{L}(G, H, D_F, D_T, F, T) = \\ \lambda_{\overrightarrow{g}}\mathcal{L}_{G_{F\to T}}(D_T, F) + \lambda_{\overleftarrow{g}}\mathcal{L}_{H_{T\to F}}(D_F, T) \\ + \lambda_{\overrightarrow{c}}\mathcal{L}^{cycle}_{G_{F\to T}}(F) + \lambda_{\overleftarrow{c}}\mathcal{L}^{cycle}_{H_{T\to F}}(T) \\ + \lambda_{\overrightarrow{i}}\mathcal{L}^{\texttt{identity}}_{G_{F\to T)}}(T) + \lambda_{\overleftarrow{i}}\mathcal{L}^{\texttt{identity}}_{H_{T\to F}}(F),\end{aligned} \quad (1)$$

where the set of λ values are selected / optimized to control the behavior of the objective function and the expected output, and

$$\begin{aligned}\mathcal{L}_{G_{F\to T}}(D_T, F) = \mathbb{E}_{f\sim p(f)}[(D_T(G(f)) - \imath)^2], \\ \mathcal{L}_{H_{T\to F}}(D_F, T) = \mathbb{E}_{t\sim q(t)}[(D_F(H(t)) - \jmath)^2], \\ \mathcal{L}^{\texttt{cycle}}_{G_{F\to T}, H_{T\to F}}(F) = \mathbb{E}_{f\sim p(f)}[\|H(G(f) - f\|_1], \\ \mathcal{L}^{\texttt{cycle}}_{H_{T\to F}, G_{F\to T}}(T) = \mathbb{E}_{t\sim q(t)}[\|G(H(t) - t\|_1], \\ \mathcal{L}^{\texttt{identity}}_{H_{T\to F}}(F) = \mathbb{E}_{f\sim p(f)}[\|H(f) - f\|_1], \\ \mathcal{L}^{\texttt{identity}}_{G_{F\to T}}(T) = \mathbb{E}_{t\sim q(t)}[\|G(t) - t\|_1],\end{aligned} \quad (2)$$

where $f \sim p(f)$ and $t \sim q(t)$ denote data distributions, $\mathbb{E}$ denotes the Expectation, and $\imath = 1$ and $\jmath = 0$ denote valid and fake images used to fool the discriminators. The aim is to minimize the objective function shown above in Eq. 1. The right arrow in $\lambda_{\overrightarrow{g}}$ denotes the forward generator and the left arrow in $\lambda_{\overleftarrow{g}}$ denotes the backward generator. Similarly, left and right arrows of λ_c and λ_i denote forward and backward cycle and identity, respectively. The discriminators' loss functions, on the other hand, are defined as follows:

$$\begin{aligned}\mathcal{L}_{D_f}(F, B_f) = \mathbb{E}_{f\sim p(f)}[(D_f(f) - \imath)^2] \\ + \mathbb{E}_{f\sim p(f)}[(D_f(b_f) - \jmath)^2], \\ \mathcal{L}_{D_t}(T, B_t) = \mathbb{E}_{t\sim q(t)}[(D_t(t) - \imath)^2] \\ + \mathbb{E}_{t\sim q(t)}[(D_t(b_t) - \jmath)^2],\end{aligned} \quad (3)$$

where B_f and B_t are buffers that store the 50 previously created images [30]. These buffers are necessary to reduce model oscillation, as has been shown in [22] that updating the discriminators using a history of generated images rather than the ones produced by the latest generators adds more stability.

Furthermore, because the dark text affects the output of the CycleGAN, *i.e.* cannot be separated from the black background, we suggest in this work a CycleGAN that simultaneously learn from both the negative and positive images. In this case, we modified Eq. 2 to be as follows:

$$\begin{aligned}\mathcal{L}_{G_{F\to T}}(D_T, F) = \mathbb{E}_{f\sim p(f)}[D_T[G(f^+) + G(f^-)] - \imath)^2], \\ \mathcal{L}_{H_{T\to F}}(D_F, T) = \mathbb{E}_{t\sim q(t)}[(D_F[H(t^+) + D_F(H(t^-)] - \jmath)^2], \\ \mathcal{L}^{\texttt{cycle}}_{G_{F\to T}, H_{T\to F}}(F) = \mathbb{E}_{f\sim p(f)}[\|[H(G(f^+)) + H(G(f^-))] - f\|_1], \\ \mathcal{L}^{\texttt{cycle}}_{H_{T\to F}, G_{F\to T}}(T) = \mathbb{E}_{t\sim q(t)}[\|[G(H(t^+)) + G(H(t^-))] - t\|_1], \\ \mathcal{L}^{\texttt{identity}}_{G_{F\to T}}(T) = \mathcal{L}^{\texttt{identity}}_{G_{F\to T}}(T^+) + \mathcal{L}^{\texttt{identity}}_{G_{F\to T}}(T^-), \\ \mathcal{L}^{\texttt{identity}}_{H_{T\to F}}(F) = \mathcal{L}^{\texttt{identity}}_{H_{T\to F}}(F^+) + \mathcal{L}^{\texttt{identity}}_{H_{T\to F}}(F^-),\end{aligned} \quad (4)$$

We also modified the discriminators' loss functions to be as follows:

$$\begin{aligned}\mathcal{L}_{D_f}(F, B_f) = \mathbb{E}_{f\sim p(f)}[(D_f(f) - \imath)^2] \\ + \mathbb{E}_{f\sim p(f)}[(D_f(b_f) - \jmath)^2] \\ \mathcal{L}_{D_t}(T, B_{t^+}, B_{t^-}) = \mathbb{E}_{t^+\sim q(t^+)}[(D_t(t^+) - \imath)^2] \\ + \mathbb{E}_{t^-\sim q(t^-)}[(D_t(t^-) - \imath)^2] \\ + \mathbb{E}_{t^\pm\sim q(t^\pm)}[(D_t(b_{t^+} + b_{t^-})) - \jmath)^2],\end{aligned} \quad (5)$$

where the f^+ and f^- superscripts respectively denote positive and negative images of f, a similar argument applies to t ground-truth images, and the Bs are buffers that store the 50 previously created / generated images. Worth to mention that $f^+ = f$. In order to simplify the notations, we dropped division by two after adding the positive and negative images.

3.3. Unsupervised text segmentation via CycleGAN

We did a few exploratory analysis to investigate text segmentation via CycleGAN. Our hypothesis is that the CycleGAN will be able to extract text from images with the exact color and shape that they appear in the scene image. This, however, has been the case to some extent. Additionally, since our CycleGAN is trained to extract text and overlay it on a black background, text that have dark colors in the input (scene-text images) will not be legible on black background, and sometimes, appears to be blended with the black background. To resolve this issue, we opt to try the negative of the input image (aka image inversion), which indeed solved the problems of the dark text. As expected, using only the negative of the input image led to a similar problem when the input images have bright text

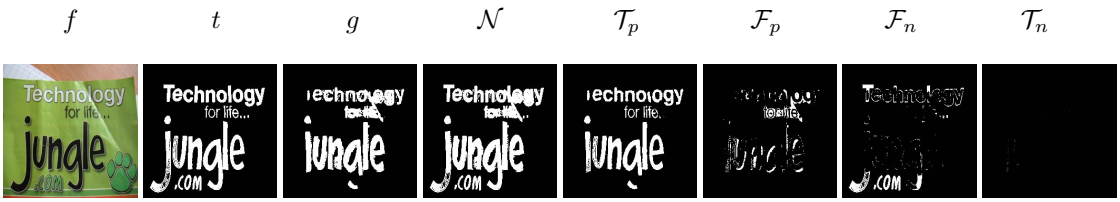

Figure 2. Results for an image f containing bright and dark text, with t as ground-truth. Values of Eq. 8 that we use to calculate the F_1 score; g is the GAN output when adding the positive and negative of the GAN, *i.e.* $g^+ + g^-$. For example, the calculation of the false positives for the this image according to Eq. 8 is $F_p = \Sigma\mathcal{F}_p$, *i.e.* over all the pixels. Best viewed in color.

(*e.g.*, white text), as the negative of the bright text will result in dark text. To resolve this problem, we propose to use both the input image and its negative; and hence, an algorithm is needed to blend the output of a CycleGAN for the input image and its negative. This is better explained in mathematical notations as shown below; let f^+ be the input image and f^- its negative, let $G_{F\to T}$ be the model trained to segment text, we can then write:

$$\begin{aligned} g^+ &= G_{F\to T}(f^+), \\ g^- &= G_{F\to T}(f^-), \end{aligned} \tag{6}$$

where g^+ and g^- are the outputs of G for the positive and negative of the input image f, respectively. Moreover, we shall also investigate more approaches to come up with a better text segmentation, including by passing the output g to G again, in addition to summing the positives and negatives as they should complement each other, as follows:

$$\begin{aligned} g^{\pm} &= g^+ + g^-, \\ gg^+ &= G_{F\to T}(g^+), \\ gg^- &= G_{F\to T}(g^-), \\ gg^{\pm a} &= gg^+ + gg^-, \\ gg^{\pm b} &= G_{F\to T}(g), \end{aligned} \tag{7}$$

where g in Eq. 7 denotes a one-pass generator of the CycleGAN of the input image f, gg denotes a two-passe generator CycleGAN approach, and the $^+$ and / or $^-$ superscripts respectively denote using the positive and / or negative input image. We also use the superscripts ^{a} and ^{b} to distinguish between two methods using a two-passe CycleGAN. We demonstrate the effect of positive and negative input images on the output of the CycleGAN in Figure 2.

3.4. Text segmentation performance metric

Evaluating the performance of text segmentation is a difficult task because of a range of problems that can affect the evaluation scores, such as: character thickness variations, merged characters, partially discovered letters, fragmented characters, and false-positives [3]. A text segmentation evaluation method based on a text detection protocol has been proposed in [3]; however, their method might not give accurate results at the pixel-wise level as it is based on a single character connected component of the segmented character. We propose in this work a novel metric for evaluating text segmentation, based on the exact pixel-wise level annotations of the test image. We calculate the segmentation performance by averaging the F_1 score over all the images of the testing set. The metric we are proposing is not similar to the ones used in other computer vision problems, where the Recall and Precision can be estimated from the predictions and the labels that do not depend on pixel-wise level annotation. To illustrate our proposed metric, let g be the image containing the segmented text, and let t be the ground-truth image that should correspond to the test image f, which has originally been used to generate g via $g = G_{F\to T}(f)$. Both g and t are in binary format. To calculate the average F_1 score over all test images, we first find the F_1 score for each segmented image g_i as follows:

$$\begin{aligned} \mathcal{N} &= g_i \vee t_i, \\ \mathcal{T}p &= g_i \wedge t_i, \\ \mathcal{F}_p &= g_i \oplus \mathcal{T}_p, \\ \mathcal{F}_n &= t_i \oplus \mathcal{T}_p, \\ \mathcal{T}_n &= \mathcal{N} \oplus (\mathcal{T}_p \vee \mathcal{F}_p \vee \mathcal{F}_n), \end{aligned} \tag{8}$$

where t_i denotes the ground-truth of the test image f_i. Each of $\mathcal{N}, \mathcal{T}p, \mathcal{F}_n, \mathcal{F}_p, \mathcal{T}_n$ have the form of a $2D$ binary having the same size of the image g_i and the ground-truth t_i. Hence for f_i image in the test set $i = 0, 1, 2, ...$, we need to take the summation over the binary matrix to obtain the True-Positives and False-Positives. The calculation of Precision, Recall and F_1 score will therefore be given by:

$$\begin{aligned} \mathcal{P} &= \frac{\Sigma\mathcal{T}_p}{\Sigma\mathcal{T}_p + \Sigma\mathcal{F}_p}, \\ \mathcal{R} &= \frac{\Sigma\mathcal{T}_p}{\Sigma\mathcal{T}_p + \Sigma\mathcal{F}_n}, \\ \mathcal{F}_1 &= 2\frac{\mathcal{R}\mathcal{P}}{\mathcal{R} + \mathcal{P}}, \end{aligned} \tag{9}$$

and hence, the average of $\mathcal{F}_1$ will give the F_1 score:

$$F_1 = \mathbb{E}_{\{(f_i \in F,\ t_i \in T):i=1,2,...\}}[\mathcal{F}_1]. \tag{10}$$

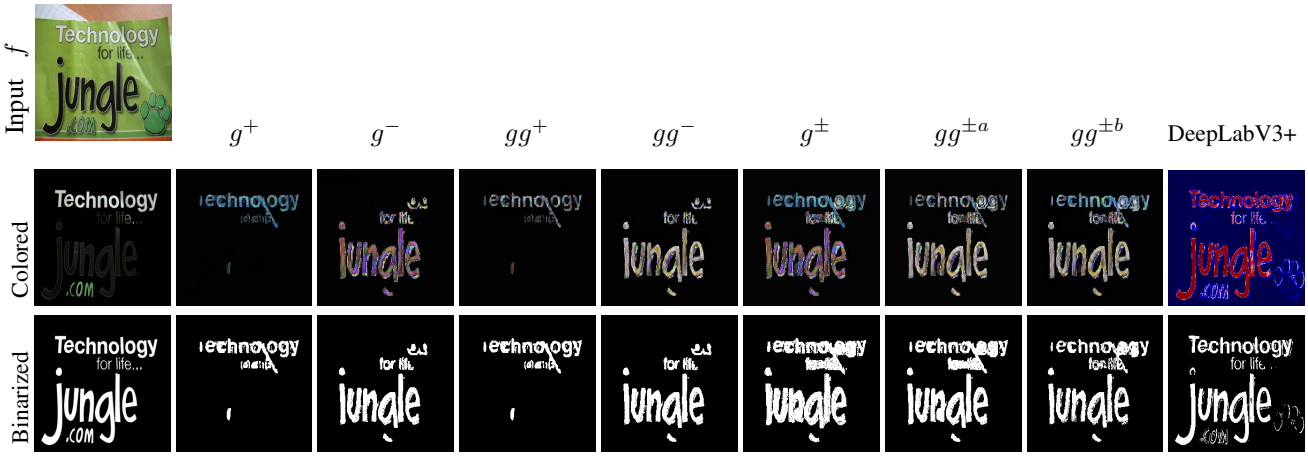

Figure 3. Segmented text using CycleGAN trained in unsupervised manner. The first column shows the input image, the ground-truth in color and binary, respectively. Columns 2 through 8 show the segmented text via seven different approaches, *e.g*. based on the positive and / or the negative of the input image f; and the last column shows the result using supervised learning of a DCNN implemented via DeepLabV3+. Best viewed in color.

As the CycleGAN output is real-valued, which could be normalized then to 0 to 255, one needs to threshold each image to binary format prior to using it in the calculation of the F_1 score. Moreover, the image that the CycleGAN generates has lots of zeros, as it tries to mimic the target domain and therefore segments text into black background. This indicates that a low threshold value will be sufficient, or one can rely on the average of the image as the threshold.

4. Results

Dateasets. We experimentally validate our models using three popular text segmentation datasets: ICDAR-2013 [12], KAIST [14], and MRRC [13]. We only consider the English images for training including the 230 images from ICDAR-2013, 310 images of KAIST and 60 image of MRRC [13]; the training set thus contain 600 images. For the testing set, we only use 233 images from the ICDAR-2013 test set. We also use 1071 Korean text images from KAIST dataset, which has been acquired by digital cameras, in zero-shot learning experiments. During training, all images were scaled up or down to 256×256 pixels, regardless of their aspect ratio.

Implementation. We forked DeepLabV3+ and CycleGAN from [29] and [15], respectively. Our text segmentation implementation code for both DeepLabV3+ [1] and CycleGAN [2] are written using PyTorch and they are publicly available.

The training of DeepLabV3+ is based on defining pixels into two classes: text and non-text. Using this supervised learning technique, we consider the pixel-level annotations, in binary format, as the ground-truth. Our DeepLabV3+ implementation uses ResNet-101 [8] as a deep learning backbone. We train DeepLabV3+ for 100 epochs with a learning rate of 0.0001 and using ADAM optimizer with AMSGrade [22] enabled. We apply Softmax on each output channel, then we obtain the predicted text pixels of the image by using $argmax$ on the two classes. One can think of the output of DeepLabV3+ as a (pseudo) probability map of text and non-text.

In unsupervised learning, we start by breaking the correspondence between each input image and its ground-truth, *i.e*. the ground-truths are randomly sampled. For the training of CycleGAN, we consider the pixel-level annotations in color as the ground-truth. We use a Residual-Net to build the generators and discriminators of the CycleGAN, having 9 and 4 Residual blocks for the generators and discriminators, respectively. We use image negation, with a probability of 0.33 as data augmentation during training. We use the following values for the objective function in Eq. 1: using $\lambda_{\overrightarrow{g}} = 1, \lambda_{\overleftarrow{g}} = 1, \lambda_{\overrightarrow{c}} = 10, \lambda_{\overleftarrow{c}} = 10, \lambda_{\overrightarrow{i}} = 5,$ and $\lambda_{\overleftarrow{i}} = 5$. Then, we train the CycleGAN from scratch for 300 epochs with a learning rate of 0.0002 using ADAM optimizer with AMSGrade [22] enabled. After the CycleGAN converges, we only use the forward generator, $G_{F \to T}$, as a text segmentor / extractor. During text segmentation, *i.e*. testing phase, we use a threshold value 10 (for images ranged between 0 and 255) to binarize the textual images generated by the CycleGAN. In addition, At testing time, the generator of the CycleGAN, which is based on Residual Net, needs 13.4 milliseconds to segment text from a scene-text image, including the time needed to read the image and place it on the GPU.

[1] https://github.com/denabazazian/scene_text_segmentation
[2] https://github.com/morawi/TextGAN

Figure 4. Qualitative results of English (first four columns) from ICDAR2013 dataset and Korean (fifth trough eighth columns) from KAIST dataset. Each column shows the input image, output of CycleGAN trained in an unsupervised manner, and DCNN text segmentation based on DeepLabV3+ trained in a supervised manner. Korean text has been segmented in zero-shot learning, *i.e*. the trained models have never seen the Korean text images. Best viewed in color.

Table 1. Text segmentation results using ICDAR-2013 dataset.

	Method	Recall%	Precision%	F_1%
Supervised	Lu *et al*. [17]	**77.27**	82.10	79.63
	BUCT-YST[11]	74.56	81.75	77.99
	I2R-NUS[12]	73.57	79.04	76.21
	USTB-FuStar[12]	69.58	74.45	71.93
	NSTsegmentator[12]	68.41	63.95	66.10
	OTCYMIST[12]	46.11	58.53	51.58
	DeepLabV3+(ours)$^{(1)}$	76.34	**83.52**	**79.76**
Unsupervised	g^+ (ours)	45.21	43.33	40.92
	g^- (ours)	45.32	48.02	43.45
	gg^+ (ours)	43.79	44.90	41.01
	gg^- (ours)	44.49	49.27	43.77
	$g^\pm$ (ours)	**81.94**	46.77	56.28
	$gg^{\pm a}$ (ours)	79.40	47.56	56.12
	$gg^{\pm b}$ (ours)	72.55	52.57	57.33

$^{(1)}$ Our implementation of text segmentation via DeepLabV3+.

We explore the approaches that we propose in Eqs. 6 and 7 to resolve the problem of dark text that the CycleGAN model might not be able to produce, or being produced correctly but blended with the dark background of the CycleGAN output. To calculate the segmentation performance, we use Eqs. 8, 9 and 10. The results, shown in Table 1, indicate that our CycleGAN approach to learning text segmentation in an unsupervised manner is a success. We present results of one sample image in Figure 3. Interestingly, the performance difference, based on the average F_1 score, between other supervised methods and the unsupervised CycleGAN that we are proposing is nontrivial. However, our approach is completely unsupervised and has significant dimensions for further improvement. We illustrate some text segmentation samples in Figure 4.

It must be noted that although we break the correspondence between the images their pixel-wise annotation, we are still using pixel-wise annotation ground-truth (via random sampling) to train the proposed unsupervised CycleGAN. We, thus, opt to take try another test using synthetically generated text images on black background. These synthetic images will only replace the ground-truth but the real-world scene text images used for input will be the same. The textual image synthesizer is very simple and its implementation is available on [link to be added], and a few synthetic text image samples are shown in Figure 5. We generated 12,000 text images and we applied (with probability of 0.5) random shearing and rotation of [-30 to 30] degrees for each text image. The CycleGAN trained with the original scene text images and the synthetically generated pixel-wise annotated text (the two datasets are totally independent) achieves an average F_1 score of 52%, which is slightly lower than the model that uses the original pixel-level annotations as ground truth (we are still speaking of unsupervised learning) that reached an average F_1 score of 56%. Still, generating synthetic textual images is a critical problem that needs further investigation due to the large number of parameters one needs to consider, *e.g*. the number of words, word size, color distribution, location and aggregation of words, *etc*.

Zero-shot learning. Zero-shot learning refers to using a trained model to predict data from a distribution that is different from the one used in training the model, *i.e*. without training the model using any instance of the new distribution. We explore in this section the ability of our models to perform text segmentation on another script/language that it has not seen during training, of which the script is totally unknown to the models. We use the Korean language dataset and we test text segmentation using 7 Cycle-

Figure 5. Synthetically generated pixel-wise text annotation ground-truth (right) independent of the real-world scene text images (left). We use this dataset to train the CycleGAN. Best viewed in color.

GAN models, we saved these models every 50 epochs while learning from English scene-text images in an unsupervised manner. Clearly, model-0 is the baseline, *i.e.* chance-level, when there has not been any training. The results we present in Figure 6 clearly demonstrate the CycleGAN model has the ability to segment totally unseen and unknown text from scene images, as it has been trained on English and tested on Korean. For a model trained up to 300 epochs, the model performance on Korean is slightly lower than English. The last four columns of Figure 4 illustrate some Korean text segmentation samples.

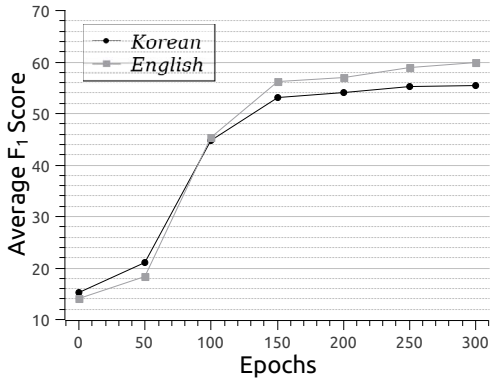

Figure 6. Evolution of the average F_1 score using ICDAR-2013 test set as input to the CycleGAN and $g^{\pm} = g^{+} + g^{-}$ as output. The GAN taught itself in an unsupervised manner using only English, and Korean text have been segmented in zero-shot learning.

4.1. Ablation study

We try a few other approaches to improve the results. First, we use different λ values aiming to have a better optimization for the objective function. In one set, we use $\lambda_{\overrightarrow{g}} = 1, \lambda_{\overleftarrow{g}} = 1, \lambda_{\overrightarrow{c}} = 10, \lambda_{\overleftarrow{c}} = 50, \lambda_{\overrightarrow{i}} = 5$, and $\lambda_{\overleftarrow{i}} = 50$ did not improve the results. The rational behind having larger values for $\lambda_{\overleftarrow{c}} = 50$ and $\lambda_{\overleftarrow{i}} = 50$ is compensating for the lower error in the generated text images compared to the scene-text images. Although the model converged using the above λ values, it did not improve the results compared to the ones presented in Table 1. Modeling the objective function shown in Eq. 1 as a one-layer neural network on top of the CycleGAN model resulted mode collapse; clearly, such neural network optimizer was trapped in a local minima due to the probable instability of the CycleGAN. Moreover, using some other randomly selected λ values resulted mode collapse too.

Fortunately, although CycleGANs are highly unstable, the model shown in Eqs. 4 and 5 did not collapse. However, the outputs do not differ from the CycleGAN based on Eq. 1. Last but not least, we investigated the thresholding effect of the text segmented via the CycleGAN with respect to the average F_1, where we normalize the output image between 0 and 255. These analysis show how rigid the background of the segmented text as the F_1 started declining after a threshold value of 50, as depicted in Figure 7.

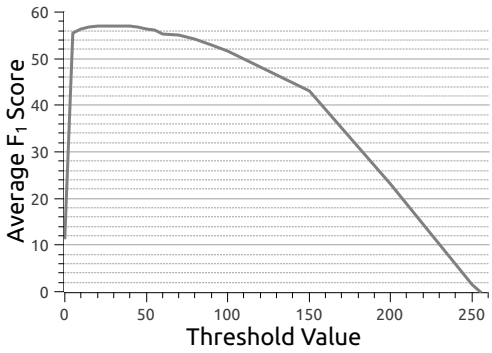

Figure 7. The effect of thresholding aimed at binarizing the segmented text via CycleGAN.

5. Discussion

In this paper, we employed Cycle-Consistency GAN to segment text from scene images, and compared our results with semantic segmentation technique that is based on DCNNs. We showed that the CycleGAN can segment text from scene images even when the ground-truth is not available. This will make it possible to build text segmentation models from large size scene-text image datasets without the need for expensive annotations. However, our supervised text segmentation results based on DCNNs are comparable with the previous works. Nonetheless, our proposed unsupervised text segmentation method is still below the supervised methods, although it is higher than one of the earliest supervised text segmentation methods that has been reported in the literature, OTCYMIST[12]. For supervised text segmenta-

tion using DCNNs, the used text ground-truth was white on black foreground, however, such white ground-truth did not work with CycleGAN and resulted mode collapse; *i.e*. yielding blank output. Clearly, the white ground-truth has negatively affected the losses during training, but the cause is not obvious as the learning is unsupervised and the GAN should restore the original image at the inverse cycle. That said, the proposed CycleGAN only converged when the text ground-truth was in color on black background. Unsupervised text segmentation via GANs has an advantage over other problems making use of GANs, when it comes to quantitatively quantifying the results. In many GANs' problems, one has to rely on the Inception Score (IS) and the Frechet Inception Distance (FID) [23, 19] to measure GANs' performance, while in text segmentation problems, one can always rely on a small set containing pixel-wise annotation text to verify the results.

One of the main advantages of employing CycleGAN technique is their capability of extracting text by maintaining the original color of the text as shown in Figure 8. Furthermore, we illustrate in Figure 9 an example using a CycleGAN to segment a dark text, which was able of extracting the dark text and some of its surrounding background pixels. By considering the task of text extraction from scene images, the CycleGAN did well and the technique is capable of extracting the dark text from images, *i.e*. by maintaining the original text color of the input scene text image. The CycleGAN came up by itself with this smart solution to show the text with its original color even though when the text is dark; the problem, however, is the black background. Therefore, CycleGAN generates, to some extent, some light-colored pixels from the background of the original image around the dark text to distinguish the dark text against the black background. However, since we evaluate the results of CycleGAN based on text segmentation evaluation method, and we have to convert the results to black and white images (by thresholding), we therefore lose the dark text extraction results. This can be one reason of low accuracy of CycleGAN in comparison with DCNNs in Table 1. However, this can be improved by further investigation, because what we propose is the first step in this direction.

Figure 8. The ability of the CycleGAN to capture the color distribution while extracting text. Best viewed in color.

Figure 9. CycleGAN has the ability to extract dark text and surround it with some highlight to distinguish it from the black background. This effect is natural to CycleGAN and no incentives were used during training to enable it produce these highlights. Best viewed in color.

6. Conclusions

Using GANs to segment text from scene images in an unsupervised manner is promising. CycleGANs performs well as text extractors from cluttered scene images, and sometimes reproduces text accurately in the original color of text from scene images. However, our unsupervised approach gives lower performance compared to the supervised text segmentation implemented via DeepLabV3+. Nonetheless, our CycleGAN generator requires only 45 MB of memory compared to 950 MB of DeepLabV3+, which also implies speed efficiency during deployment. To the best of our knowledge, this is the first time for this type of research and further efforts are needed to improve the performance of unsupervised text segmentation models. This work has thus opened many researching opportunities in text segmentation with quite a few directions available for improvement, including but not limited to: 1) employing further optimization to the CycleGAN, especially the hybrid loss functions; 2) using innovative color augmentation and geometric transformation methods on the training data; 3) developing novel methods to resolve the dark text; 4) adding attention layer(s) to enable CycleGANs attend to text; 5) designing and training double and/or triple pass GANs; and 6) experimenting with other state-of-the-art unpaired / unsupervised GANs. We also show that the proposed text segmentation models have the advantage of zero-shot learning to segment text not seen during training; *e.g*. Korean text, although our models have only been trained with images containing English text.

Acknowledgement

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No. 665919 and from the project TIN2017-89779-P (AEI/FEDER, UE).

References

[1] M. Abadi and D. Andersen. Learning to protect communications with adversarial neural cryptography. *In Proc. ICLR*, 2017. 2

[2] B. Bai, F. Yin, and C. Liu. A seed-based segmentation method for scene text extraction. *In Proc. ICDAR*, pages 262 – 266, 2014. 2

[3] S. Calarasanu, J. Fabrizio1, and S. Dubuisson. From text detection to text segmentation:a unified evaluation scheme. *In Proc. ECCV workshops IWRR*, pages 378 – 394, 2016. 4

[4] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *In Proc. ECCV*, 2018. 2

[5] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, 2017. 2

[6] Y. R. Cui, Q. Liu, C. Y. Gao, and Z. Su. Fashiongan: Display your fashion design using conditional generative adversarial nets. *Computer Graphics Forum*, 37:109–119, 2018. 2

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *In Proc. NIPS*, 2014. 2

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition supplementary materials. *In Proc. CVPR*, 2016. 5

[9] B. Hitaj, P. Gasti, G. Ateniese, and F. Pérez-Cruz. Passgan: A deep learning approach for password guessing. In *ACNS*, 2019. 2

[10] N. Howe. A laplacian energy for document binarization. *In Proc. ICDAR*, pages 6 – 10, 2011. 2

[11] W. Hu. Result of ICDAR2015 robust reading competition. *Available: http://rrc.cvc.uab.es*, 2015. 6

[12] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. Bigorda, S. Mestre, J. Mas, D. Mota, J. Almazan, and L. Heras. ICDAR 2013 robust reading competition. *In Proc. ICDAR*, pages 148 – 1493, 2013. 5, 6, 7

[13] D. Kumar, M. Prasad, and A. Ramakrishnan. Multi-script robust reading competition in icdar2013. *In Proc. ICDAR Workshops MOCR*, 2013. 5

[14] S. Lee, M. Cho, K. Jung, and J. Kim. Scene text extraction with edge constraint and text collinearity. *In Proc. ICPR*, pages 3983 – 3986, 2010. 5

[15] E. Linder-Noren. Pytorch implementation of GANs. https://github.com/eriklindernoren/PyTorch-GAN. [Online; accessed 24-April-2019]. 5

[16] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *In Proc. CVPR*, pages 431 – 3440, 2015. 2

[17] S. Lu, T. Chen, S. Tian, J. Lim, and C. L. Tan. Scene text extraction based on edges and support vector regression. *International Journal on Document Analysis and Recognition (IJDAR)*, 18:125–135, 2015. 6

[18] P. Luc, C. Couprie, S. Chintala, and J. Verbeek. Semantic segmentation using adversarial networks. *In Proc. NIPS*, 2016. 2

[19] M. Lucic, K. Kurach, M. Michalski, O. Bousquet, and S. Gelly. Are gans created equal? a large-scale study. In *In Proc. NeurIPS*, pages 698–707, 2018. 8

[20] P. Lyu, X. Bai, C. Yao, Z. Zhu, T. Huang, and W. Liu. Auto-encoder guided gan for chinese calligraphy synthesis. *In Proc. ICDAR*, pages 1095–1100, 2017. 2

[21] A. Mishra, K. Alahari, and C. Jawahar. An MRF model for binarization of natural scene text. *In Proc. ICDAR*, pages 11 – 16, 2011. 2

[22] S. Reddi, S. Kale, and S. Kumar. On the convergence of adam and beyond. *In Proc. ICLR*, 2018. 3, 5

[23] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *In Proc. NIPS*, pages 2234–2242, 2016. 8

[24] B. Su, S. Lu, and C. Tan. Binarization of historical document images using the local maximum and minimum. *In Proc. IAPR workshop at ACM*, pages 150 – 166, 2010. 2

[25] Y. Tang and X. Wu. Scene text detection and segmentation based on cascaded convolution neural networks. *IEEE Transactions on Image Processing*, 26:1509–1520, 2017. 2

[26] S. Tian, S. Lu, B. Su, and C. Tan. Scene text segmentation with multi level maximally stable extremal regions. *In Proc. ICPR*, pages 2703 – 2708, 2014. 2

[27] I. Yalniz, D. Gray, and R. Manmatha. Efficient exploration of text regions in naturalscene images using adaptive image sampling. *In Proc. ECCV workshops IWRR*, pages 427 – 439, 2016. 2

[28] F. Zhan, H. Zhu, and S. Lu. Spatial fusion gan for image synthesis. *In Proc. CVPR*, 2019. 2

[29] J. Zhang. Pytorch implementation of DeepLab-V3-Plus. https://github.com/jfzhang95/pytorch-deeplab-xception. [Online; accessed 24-Feb-2019]. 5

[30] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *In Proc.ICCV*, pages 2242–2251, 2017. 2, 3