Using Object Information for Spotting Text

Shitala Prasad[1] and Adams Wai Kin Kong[2]

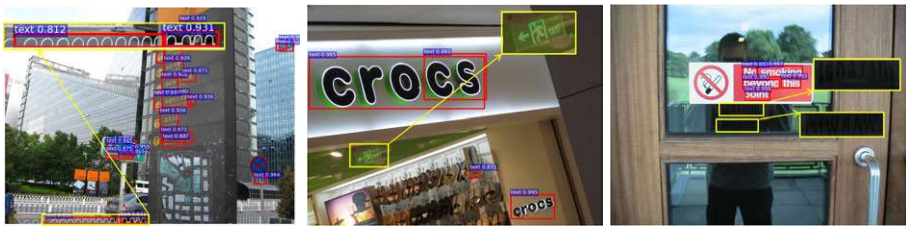
[1]CYSREN@NTU and [2]School of Science and Computer Engineering
Nanyang Technological University Singapore
{shitala,adamskong}@ntu.edu.sg

Abstract. Text spotting, also called text detection, is a challenging computer vision task because of cluttered backgrounds, diverse imaging environments, various text sizes and similarity between some objects and characters, e.g., tyre and 'o'. However, text spotting is a vital step in numerous AI and computer vision systems, such as autonomous robots and systems for visually impaired. Due to its potential applications and commercial values, researchers have proposed various deep architectures and methods for text spotting. These methods and architectures concentrate only on text in images, but neglect other information related to text. There exists a strong relationship between certain objects and the presence of text, such as signboards or the absence of text, such as trees. In this paper, a text spotting algorithm based on text and object dependency is proposed. The proposed algorithm consists of two sub-convolutional neural networks and three training stages. For this study, a new NTU-UTOI dataset containing over 22k non-synthetic images with 277k bounding boxes for text and 42 text-related object classes is established. According to our best knowledge, it is the second largest non-synthetic text image database. Experimental results on three benchmark datasets with clutter backgrounds, COCO-Text, MSRA-TD500 and SVT show that the proposed algorithm provides comparable performance to state-of-the-art text spotting methods. Experiments are also performed on our newly established dataset to investigate the effectiveness of object information for text spotting. The experimental results indicate that the object information contributes significantly on the performance gain.

Keywords: Text Detection, Natural Scenes, Deep Learning, Object Detection, RCNN

1 Introduction

Text understanding in natural images is an important prerequisite for many artificial intelligence (AI) and computer vision (CV) applications, such as autonomous robots, systems for visually impaired, context retrieval, and multi-language machine translation based on image inputs [1–9]. Researchers have demonstrated that once text is well detected, the existing text recognition methods can achieve high accuracy [4, 6]. Text spotting is the current bottleneck and is a challenging CV task, because backgrounds in natural scenes such as street



(a) (b) (c)

Fig. 1. Challenges in text spotting: the yellow box represents missed and/or wrongly detected texts by one of the state-of-the art methods [11]. (a)-(c) are respectively errors due to the road barrier, signboard and text reflection on glass. Note that the road barrier is similar to text, 'lllllllll' and 'nnnnnn'.

view images are highly cluttered and the text in them has large difference in styles (e.g., artistic fonts, Time New Roman and Sim Sun with different colors), languages (e.g., Chinese, Japanese and English), sizes (e.g., text on a signboard of a cafe and text on its food menu board), illumination conditions (e.g., offices, restaurants, bars, sunny countryside, and cloudy streets), and contrasts (e.g., over-exposed and under-exposed). Other factors, including low resolution, out-of-focus, occlusion, and similarity between objects and characters (e.g., tyre and the character 'o') impose additional difficulties on text spotting [10]. Figure 1 illustrates some of these challenges. Thus, researchers are still actively seeking more robust and accurate text spotting methods.

Currently researchers concentrate on designing more effective deep network architectures and training schemes to seek more useful information in text, including character-level, word-level, text line-level and precise text location up to accuracy of one pixel [12, 13]. For particular applications, such as shopping assistants for grocery and book stores [14, 15], more prior knowledge can be exploited for achieving higher detection accuracy. More precisely, in these environments, text can appear in particular locations with a similar style and color and the backgrounds are more predictable. However, this prior knowledge is not generally applicable to natural scene images, which likely have clutter backgrounds, because there is no control over where and how the images are taken.

Even though images are not taken from a particular environment, we still have rough idea about them, because they are taken from where we stay, live, work, and travel, such as city, street, office, cafe, and park. Text appears likely on particular man-made objects, e.g., book, computer, and signboard but unlikely on natural objects, e.g., water, sky, tree, and grass. Some objects are more often with text than others. For example, text always appears on car plate but not always on the side of car. More clearly, objects and text are not independent. The appearance of text is typically dependent on the type of objects in the scene. Figure 2 illustrates few dependence between objects and text in street view images. Furthermore, this information is possible to reduce detection errors which are due to the similarity between objects and text, e.g., tyre and 'O'. Once a car is detected, it implies that text unlikely appears in its bottom. According

(a) (b) (c) (d)

Fig. 2. Dependence between objects and text in street view images. For example, (a and c) sign board and digit, (a-b) car and car plate, (b) building and text, and (d) cloth and text.

to the best knowledge of the authors, none of the previous studies exploited this information for detecting text in natural scene images. The aim of this paper is to develop an algorithm to exploit this information for enhancing text spotting performance. In this study, the authors are particularly interested in images with cluster backgrounds, such as images taken from streets because they are challenging even to the state-of-the-art methods and likely contain objects, the target of this study.

Text spotting can be considered as a specific case of object detection. In recent years, the advancements in object detection are driven by the region proposal (RP) methods [11, 16, 17]. Fast RCNN [18] and their latest developments [11] are some of these methods. Faster RCNN sharing the convolutional layers with the region proposal networks (RPNs) and fast RCNN, is one of the best among state-of-the-art methods in object detection with low computation cost [11]. Because of its performance in terms of accuracy and speed, it is selected for this study as a baseline network.

Converting faster RCNN to detect text in images with cluttered backgrounds can be done through training the network using images with their text labels only. However, this approach does not consider object information, which is the focus of this study. If the network is trained using images with object and text labels together as the original faster RCNN training procedure, possibly, the objects would degrade its performance for text because the network would balance its performance between text and other objects. Another approach is to encode the objects and text relationship on a knowledge graph, where each node represents a specific type of object or text and each edge describes how likely two objects or text and an object appear together. This approach can use faster RCNN to first detect objects and then use the adjacency matrix of the knowledge graph to refine the results from faster RCNN [11]. It can in fact be considered as a decision level fusion, because the final results from faster RCNN, which are the bounding boxes of the objects and text, are fused with the knowledge graph information. This approach neither makes use of the object features in faster RCNN nor optimizes the network end-to-end. These potential approaches are likely sub-optimal. In this paper, an algorithm is proposed to exploit object features and text features in a deep network directly and to train it end-to-end for achieving better performance.

For this study, a new text dataset named Nanyang Technological University Unconstrained Text and Object Image Dataset (NTU-UTOI) is established. This dataset contains 22,767 natural scene images with 165,749 bounding boxes for 42 classes of objects and 111,868 bounding boxes for text[1], including English, Chinese and digits. Figure 2 shows samples in the NTU-UTOI dataset. More information about the dataset can be found in Section 4. According to our best knowledge, it is the second largest real (non-synthetic) natural scene image dataset for text spotting. NTU-UTOI is used for training and testing the proposed algorithm. In addition, three benchmarks from three different groups are also employed in the evaluations and comparisons: SVT[2], MSRA-TD500[3], and COCO-Text[4]. These three databases are challenging because their images are taken from diverse environments and with clutter backgrounds.

The rest of the paper is organized as follows: Section 2 gives a very brief summary of state-of-the-art text detecting methods. Section 3 elaborates the proposed algorithm. Section 4 reports comparison results with the state-of-the-art text detection methods on the three benchmark datasets along with NTU-UTOI dataset. Section 5 gives some conclusive remarks.

2 Related Works

Text detection in natural scene images has been studied for several decades [2, 12, 19, 20] and various methods have been proposed, which can be broadly categorized into character-region methods and sliding windows methods. The character-region methods aim to segment pixels into characters and then group the characters into words [12, 19–24] while the sliding window methods determine whether the pixels in a sliding window belong to text or not [9, 25–27]. Text detection can also be categorized as image processing-based methods and deep learning-based methods. The image processing-based methods pre-process images and then extract features and finally classify pixels into text and background. The deep learning methods exploit the capability of deep networks to automatically extract features and perform detection based on their feature maps. Generally speaking, deep learning methods perform better but demands more computational resources, particularly in training.

Epshtein et al. proposed a per-pixel output transformation called stroke width transform (SWT) for text detection [12]. Neumann and Matas [24] proposed a method based on gradient filters to detect oriented strokes, which significantly outperforms SWT. Anthimopoulos et al. proposed a sliding window method, which uses dynamically normalized edges as features and a random forest classifier to detect text in natural scene images [27]. Chen et al. used edge-enhanced maximally stable extremal regions (MSERs) for text detection

[1] In NTU-UTOI, the term text means English, Chinese and Digit.

[2] `http://tc11.cvc.uab.es/datasets/SVT_1`

[3] `http://www.iapr-tc11.org/mediawiki/index.php/MSRA_Text_Detection_500_Database_(MSRA-TD500)`

[4] `https://vision.cornell.edu/se3/coco-text-2/`

[19]. It outperforms SWT because it is more robust to blurred images and more effective for filtering out false-positive characters. Posner et al. proposed a cascade of boosted classifiers with a saliency map to create bounding boxes for text detection [28]. In 2012, Wang et al. claimed to be the first group using convolutional neural network (CNN) for text spotting [29]. They trained a CNN on a synthetic dataset [8].

In recent years, researchers consider words and text lines as a whole generic object but ignore the character components such that generic object detectors can be modified for text detection [13]. In 2017, Rong et al. proposed a recurrent dense text localization network (DTLN) using long short term memory (LSTM) for unambiguous text localization and retrieval [15]. Zhong et al. modified faster RCNN for text detection [10]. Furthermore, Liao et al. proposed TextBoxes, which is inspired by Single Shot multibox Detector (SSD) [30], to achieve higher detection accuracy and speed [31].

In fact, text can be considered as a generic object as discussed earlier. Using deep learning and region proposal network (RPN) for generic object detection has attracted great attention from many researchers. The state-of-the-art object detection methods based on RPN have achieved very significant improvement [32], [18] comparing with the traditional methods. In addition to faster RCNN, there are other region proposal methods, such as selective search (SS) [33], multiscale combinatorial grouping (MCG) [34], and edge-boxes (EB) [35]. These methods generate exceedingly large amount of region proposals, resulting in high recall but more computation demanding. To overcome this problem, RPN computes region proposals through sharing convolutional layers with fast RCNN that exponentially reduces the computational cost and achieves a promising recall rate. Inspired by [11], in this paper, RPN is trained on same images with object labels and then combined with another deep network and trained together on images with text labels. Researchers have proposed deep learning models and trained them on large datasets such as COCO-Text and SynthText [36, 37] but none of them exploited object information nearby text.

3 Methodology

This section first describes the proposed deep network architecture and training stages. Then, anchor parameters, which are designed for text spotting are given. The loss function for training the network and the implementation details are provided in the end of this section.

3.1 Network Architecture and Training Stages

To use object features in deep networks for enhancing text spotting[5] performance, a convolutional neural network (CNN) with two sub-networks and three training stages is proposed. The proposed deep network is named Text and

[5] Here, by text spotting we mean text detection and not text recognition.

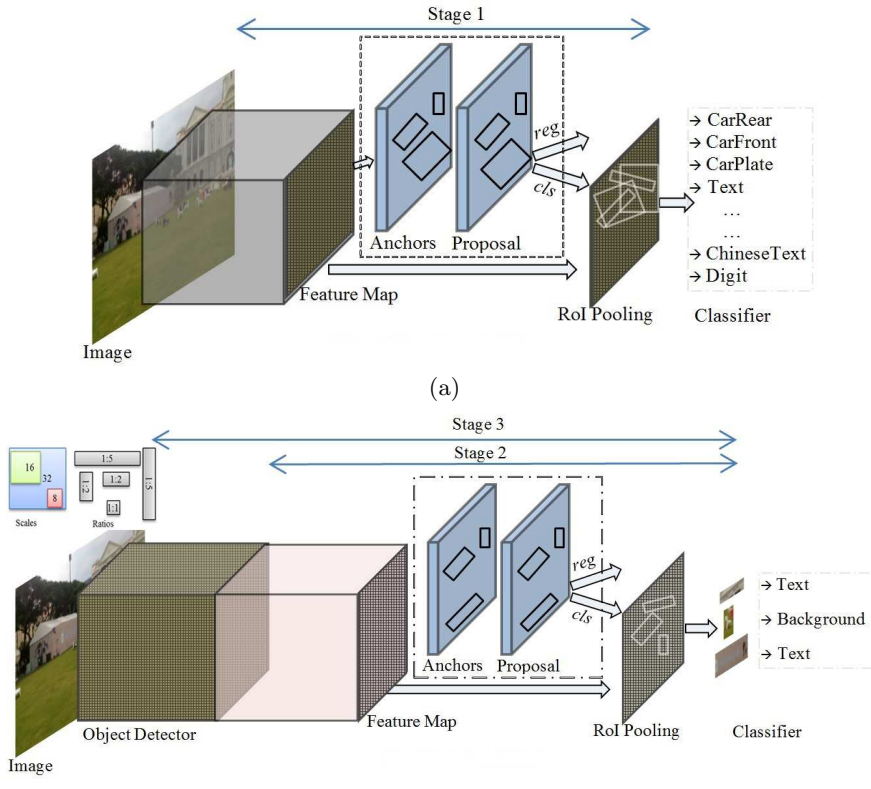


Fig. 3. The proposed TO-CNN for text spotting based on object information. (a) Illustrates the first training stage to extract object information and store in the Object CNN. (b) Illustrates the second training stage to tune the parameters in the Text CNN and the third training stage to fine tune the entire network for text spotting.

Object-based CNN (TO-CNN). Figure 3 illustrates the proposed deep network and training stages. In this study, faster RCNN with VGG-16 net [38] as a backbone is used to extract object and text information. At the first training stage, faster RCNN is trained on images with text and object labels illustrated in Figure 3(a). Once the network is fully trained, the object and text information would be stored in the VGG-16 net. For the sake of convenience, the trained VGG-16 net is called Object VGG-16 net. Note that it does store text information. Object VGG-16 net is separated from other components in the faster RCNN. A CNN which is modified from another VGG-16 network is added on the Object VGG-16 net. This CNN is called Text VGG-16 net. The details of the modification will be given later. The Object VGG-16 and the Text VGG-16 together form the backbone of TO-CNN. TO-CNN also consists of RPN and the regression networks from faster RCNN illustrated in Figure 3(b). At the second training stage, TO-CNN is trained on images with text labels only and all pa-

rameters in the Object VGG-16 net are fixed. In this stage, the Text VGG-16 net takes the object and text features from the Object VGG-16 to tune its parameters for text detection. From another point of view, the Text VGG-16 net fuses the text and object features for text detection. At the third training stage, the entire TO-CNN, including the Text VGG-16 net and the Object VGG-16 net is fine-tuned. At the end of this training stage, the network is fully optimized for text spotting based on object and text information.

The Text VGG-16 net is modified to take input feature maps from the Object VGG-16 net. There are different approaches to merge two networks together [39–41]. The stacked hourglass approach [40] is one of the effective approaches. In this paper, following the similar hourglass approach, the output of the Object VGG-16 net is up-sampled and combined to the Text VGG-16 net adding three up-sampling and one normalization layers for further RPN learning process.

In order to detect objects with different sizes, faster RCNN uses hyper-parameters, i.e., scale and ratio to control the region proposals. Ren et al. used three scales to determine the size of sliding anchors: 8, 16 and 32 with three aspect ratios: 1:1, 1:2 and 2:1 [11]. In TO-CNN, the scale is also fixed to three levels but the aspect ratio is modified, as their aspect ratios were designed for generic object detection. Text usually has different aspect ratios compared to objects, and therefore new aspect ratios are set to 1:1, 1:2, 2:1, 1:5 and 5:1 to cover almost all text lines and words in images. The summary of the anchors used in the proposed network is given in Figure 3(b) top-left. Note that, in each point on the final feature map, there are 15 anchors (5×3) at each sliding position. So for a convolutional map of $W \times H$, there are $W \times H \times 15$ anchors.

TO-CNN uses the same translation-invariant property of RPN [11], which results in 2,397,696[6] parameters in the proposal layer. More clearly, if text is translated in an image, the proposal will also be translated and the same function will be used to predict the proposal regardless of their translated locations.

3.2 Loss Functions

In the first training stage, the original loss function in faster RCNN is employed to extract object information. In the second and third training stages, the multi-task loss function $\mathfrak{L}$ given below is used [42]

$$\mathfrak{L}(p_l, v, v^*) = \mathfrak{L}_{cls}(p_l) + \alpha \mathfrak{L}_{reg}(v, v^*) \tag{1}$$

where $l = 1$ and $l = 0$ represent text and background, respectively, p_l is the corresponding probability computed using softmax, $\mathfrak{L}_{cls}$ is a classification loss and $\mathfrak{L}_{reg}$ is a regression loss between predicted and ground truth bounding boxes, α is a weight balancing these two losses and v and v^* are the predicted and ground truth bounding boxes, respectively. The bounding boxes are represented by their top left corner coordinates, width and height, i.e., $\{v_x, v_y, v_w, v_h\}$ for

[6] The dimensions of feature map, *reg* and *cls* are 512, 4, and 1 respectively. The kernel size is 3 by 3 and the number of anchors is 15. Thus, the number of parameters is $3 \times 3 \times 512 \times 512 + 512 \times 15 \times (4 + 1) = 2,397,696$.

v and $\{v_x^*, v_y^*, v_w^*, v_h^*\}$ for v^*. The classification and regression losses are defined respectively in Equations 2 and 3,

$$\mathfrak{L}_{cls}(p_l) = -\log p_l \tag{2}$$

$$\mathfrak{L}_{reg}(v, v^*) = \sum_{i\epsilon\{x,y,w,h\}} smoothL_1(v_i - v^*) \tag{3}$$

where

$$smoothL_1(x) = \begin{cases} 0.5x^2 & if|x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \tag{4}$$

In this paper, $smoothL_1$ loss is used as it is less sensitive to outliers and needs less attention on tuning the learning rate [13]. As with RPN, here the features used for regression are of the same dimension, which is 3 by 3 on the feature maps. This helps in achieving bounding box regression more efficiently [11].

3.3 Training and Implementation Details

The Object CNN and the Text CNN are initialized by VGG-16 pre-trained ImageNet classification model [38]. The weights are updated using 10^{-3} and 10^{-4} for the first 100,000 and next 350,000 iterations, respectively. The base learning rate is 10^{-3} and the learning rate decay parameter γ is 0.1. The weight decay and momentum are set to $\omega = 5 \times 10^{-4}$ and $\mu = 0.9$, respectively. These parameters are employed in all three training stages.

All the experiments are conducted on Intel Xeon E5-2690 CPU workstation with 32GB RAM, NVIDIA Quadro M6000 24GB and Ubuntu 14.04 OS. Caffe[7] is used to implement TO-CNN.

4 Experiments and Results

Three benchmark datasets: SVT, MSRA-TD500 and COCO-Text are employed to evaluate the performance of the proposed algorithm. These three databases are challenging even for the state-of-the-art methods because their images were collected from diverse environments, including inner and outdoor environments under different lighting conditions and have clutter backgrounds. The COCO-Text dataset [43] is a subset of the MS COCO dataset [44], which is used for studying object detection task. It contains 63k images taken from complex everyday scenes from which 10k is used for validation and 10k for testing. Figure 4(a) shows sample images from COCO-Text dataset. MSRA-TD500 is a multilingual dataset that includes both English and Chinese text along with digits in high resolution. MSRA-TD500 consists of 500 natural scene images. Out of them 200 are testing images and 300 of them are training images. Figure 4(b) shows sample images of MSRA-TD500 dataset. The street view text (SVT) dataset

[7] `http://caffe.berkeleyvision.org/`

(a) (b) (c) (d)

Fig. 4. Text samples from different datasets: (a) COCO-Text, (b) MSRA-TD500, (c) SVT and (d) NTU-UTOI - proposed dataset.

consists of images collected from Google street view and is annotated in word-level. It consists of smaller and lower resolution text from street view. SVT has 100 images for training and 249 images for testing with total 647 annotated words (not fully annotated). It is challenging as it has few incomplete and/or occluded texts with low image quality. Figure 4(c) shows some sample images from this dataset.

In addition to these three different benchmark datasets, TO-CNN is also examined on NTU-UTOI dataset established by the authors. NTU-UTOI dataset consists of 22,767 images from ICDAR 2011 robust scene text[8], ICDAR 2015 incident scene text[9], KAIST scene text[10], MSRA-TD500, NEOCR[11], SVT, USTB-SV1k [3], and Traffic Sign datasets [45], together with images collected from the Internet and authors' personal collections. 18,173 images are used for training and the rest 4,594 images are used for testing. It should be emphasized that the training set of NTU-UTOI neither contains any testing images from COCO-Text, MSRA-TD500 nor SVT. Thus, TO-CNN could be trained on the training set of NTU-UTOI and examined on the testing sets of COCO-Text, MSRA-TD500 and SVT. The sample images from NTU-UTOI dataset are shown in Figure 4(d). Text and 42 object classes, which positively associate or negatively associate with text, were labeled. They are common street view object. Table 1 lists all the classes. The labels are selected because they have strong relationship with

[8] http://www.cvc.uab.es/icdar2011competition/?com=introduction
[9] http://rrc.cvc.uab.es/?ch=4\&com=introduction
[10] http://www.iapr-tc11.org/mediawiki/index.php/KAIST_Scene_Text_Database
[11] http://www.iapr-tc11.org/mediawiki/index.php?title=NEOCR:_Natural_Environment_OCR_Dataset

Table 1. The object labels of the NTU-UTOI dataset and the frequency counts.

Bus (673)	Train (31)	CarSide (4728)	SignBoard (18054)	TrafficLight (697)
Tag (183)	Truck (2539)	Cartoon (228)	SpeedSign (3702)	TrafficBoard (3810)
Logo (4514)	Banner (1640)	Warning (338)	StopBoard (882)	MonitorScreen (550)
Shop (2046)	Person (18829)	Building (5198)	TreePlant (16445)	ParkingSymbol (610)
English (60561)	Poster (4779)	CarFront (7413)	BottleCan (201)	PersonCartoon (1546)
Bike (548)	Symbol (802)	CarPlate (4461)	StreetLight (12730)	VehicleSymbol (276)
Cycle (474)	Animals (251)	DoorPlate (233)	StreetName (1028)	TrafficDirection (8112)
Digit (19654)	Windows (16254)	CoffeeCup (152)	ChineseText (31653)	BuildingNumber (91)
Other (4931)	CarRear (14001)	NamePlate (352)	FoodEatable (797)	BuildingHolding (620)

Fig. 5. Example detection results of TO-CNN from MSRA-TD500 benchmark dataset.

text and commonly appear in natural scene images. Totally, 277,617 bounding boxes for text and text related objects were manually labeled and cross verified by two workers per image.

The NTU-UTOI dataset is also a challenging dataset. The images were collected from various imaging environments with patterns similar to text (e.g., windows are similar to "D", "O" and "0", railings similar to "1" and "l", and tires similar to "o" and "O") and also with multi-lingual, multi-oriented and multi-scale text. Moreover, it contains blurred and incidental text and images from indoor, outdoor, street, crowd, road, poster and mobile/TV screens. Some examples are given in Figure 2 and Figure 4(d).

Precision (P), recall (R) and F-score (F) are used as performance indexes to evaluate the proposed algorithm and compare it with the state-of-the-art text spotting methods. MSRA-TD500 and SVT have been extensively used as benchmarks for algorithm evaluation and COCO-Text is a newly released benchmark. Different research groups use different datasets to evaluate their methods and train them on different datasets. For each of the benchmark datasets, the methods reported with state-of-the-art results are selected for comparisons. Thus, different methods are selected in these comparisons. Their training sets and the baseline networks are also listed in the resultant tables. Note that IoU (intersection over union) in this paper is taken as 0.5 to be the correct match. Tables 2, 3 and 4 list respectively the precision, recall and F-score from MSRA-TD500, SVT and COCO-Text. Figure 5, 6 and 7 show sample outputs of MSRA-TD500, SVT and COCO-Text, respectively.

Table 2 shows the comparisons among TO-CNN and the state-of-the-art methods on MSRA-TD500. TO-CNN achieves the best results in terms of precision, recall and F-score. TO-CNN achieves precision rate of 0.87, which is same as EAST [37] and Lyu et al. [46]. Because of the object information in TO-CNN, it achieves recall rate of 0.90, which is significantly higher than all the other methods by at least 0.14. Figure 5 shows some outputs from MSRA-TD500.

Table 2. Comparison on the MSRA-TD500 dataset.

Methods	&Train	Baseline Network	MSRA-TD500		
			P	R	F
Kong et al. [47]	-	-	0.71	0.62	0.66
Yao et al. [48]	$\text{MSRA-TD500}_{\text{tr}}$	-	0.63	0.63	0.60
Yin et al. [3]	$\text{MSRA-TD500}_{\text{tr}}$	-	0.81	0.63	0.74
Yin et al. [49]	-	-	0.71	0.61	0.65
Zhang et al. [2]	$\text{MSRA-TD500}_{\text{tr}}$, $\text{ICDAR13}_{\text{tr}}$, $\text{ICDAR15}_{\text{tr}}$	VGG-16	0.83	0.67	0.74
Yao et al. [50]	$\text{MSRA-TD500}_{\text{tr}}$, $\text{ICDAR13}_{\text{tr}}$, $\text{ICDAR15}_{\text{tr}}$	VGG-16	0.77	0.75	0.76
RRPN [42]	$\text{MSRA-TD500}_{\text{tr}}$, $\text{HUST-TR400}_{\text{a}}$	VGG-16	0.82	0.68	0.69
SegLink [51]	$\text{SynthText}_{\text{a}}$	VGG-16	0.86	0.70	0.77
EAST [37]	$\text{MSRA-TD500}_{\text{tr}}$, $\text{HUST-TR400}_{\text{a}}$	PVANET	0.83	0.67	0.74
EAST [37]	$\text{MSRA-TD500}_{\text{tr}}$, $\text{HUST-TR400}_{\text{a}}$	VGG-16	0.82	0.62	0.70
EAST [37]	$\text{MSRA-TD500}_{\text{tr}}$, $\text{HUST-TR400}_{\text{a}}$	PVANET2x	**0.87**	0.67	0.76
Lyu et al. [46]	$\text{SynthText}_{\text{a}}$	VGG-16	**0.87**	0.76	0.81
*__TO-CNN__ (proposed)	$\text{NTU-UTOI}_{\text{tr}}$	VGG-16	**0.87**	**0.90**	**0.88**

*Note that TO-CNN exploits object information. It cannot be solely trained on the previous training sets. &The subscripts *tr*, *te* and *a* indicate respectively the corresponding training, testing and entire datasets. For example, $\text{MSRA-TD500}_{\text{tr}}$, and $\text{MSRA-TD500}_{\text{te}}$ represent the training and testing sets of MSRA-TD500, respectively. The symbol "-" indicates that either they do not use deep model or the information is not clearly described.

Table 3. Comparison on the SVT dataset.

Methods	&Train	Baseline Network	SVT		
			P	R	F
FCRN multi-scl [8]	$\text{SynthText}_{\text{a}}$	FRCN	0.47	0.45	0.46
FCRN single-scl [8]	$\text{SynthText}_{\text{a}}$	FRCN	0.51	0.41	0.46
Epshtein et al. [12]	-	-	0.54	0.42	0.47
Mao et al. [52]	-	-	0.58	0.41	0.48
FCRN+multi-filt [8]	$\text{SynthText}_{\text{a}}$	FRCN	0.62	0.52	0.56
Jaderberg [4]	$\text{SynthText}_{\text{a}}$	VGG	0.63	0.49	0.54
FCRNall+multi-filt [8]	$\text{SynthText}_{\text{a}}$	FRCN	0.65	0.60	0.63
DTLN [15]	$\text{SynthText}_{\text{a}}$	VGG-16	0.65	0.63	0.64
Zhang et al. [53]	-	-	0.68	0.53	0.60
TO-CNN (proposed)	$\text{NTU-UTOI}_{\text{tr}}$	VGG-16	**0.95**	**0.75**	**0.84**

&The subscripts *tr* and *a* indicate the corresponding training and entire dataset.

Table 3 lists the results from TO-CNN and the state-of-the-art methods on SVT. TO-CNN achieves precision rate of 0.95, recall rate of 0.75 and F-score of 0.84. Its precision and recall rates are significantly higher than the other methods at least 0.27 and 0.12 respectively. Figure 6 shows some detection results of TO-CNN. Comparing the precision rates, recall rates and F-scores of the other methods on the two datasets, it is noted that SVT is more challenging. TO-CNN still provides stable performance for SVT.

COCO-Text contains 63k images with 173k labeled text regions mainly focusing English text regions. In process of training TO-CNN, it is first trained using object and text labels in NTU-UTOI in phase one and then trained using text labels in COCO-Text in the second and third training stages. TO-CNN provides comparable results in terms of precision, recall and F-score (see Table 4 and Figure 7). Methods A, B and C developed by Google, TextSpotter and

Fig. 6. Example detection results from TO-CNN on the SVT benchmark dataset.

Table 4. Comparison on the COCO-Text dataset.

Methods	&Train	Baseline Network	COCO-Text		
			P	R	F
Baseline C [43]	-	-	0.19	0.05	0.07
Baseline B [43]	-	-	0.90	0.11	0.19
Baseline A [43]	-	-	0.83	0.23	0.36
Yao et al. [54]	SynthText_{a}	YOLO	0.31	0.18	0.22
Yao et al. [50]	-	-	0.43	0.27	0.33
He et al. [55]	ICDAR13_{tr} ICDAR15_{tr}	VGG-16	0.46	0.31	0.37
Lyu et al. [46]	SynthText_{a}	VGG-16	**0.62**	0.32	0.42
EAST [37]	COCO-Text_{tr}	VGG-16	0.50	0.32	0.39
TO-CNN (proposed)	NTU-UTOI_{tr}	VGG-16	0.41	**0.44**	0.43
TO-CNN (proposed)	NTU-UTOI_{tr}, COCO-Text_{tr}	VGG-16	0.47	**0.44**	**0.45**

&The subscripts *tr* and *a* indicate the corresponding training and entire dataset.

Fig. 7. Detection results of TO-CNN from COCO-Text dataset.

VGG have performance 0.36, 0.19 and 0.07 [43]. TO-CNN achieves the highest recall rate and F-score.

Comparisons on NTU-UTOI dataset are shown in Table 5 for demonstrating the usefulness of object information in text spotting. Here, it is compared with

Table 5. Comparison on the NTU-UTOI dataset.

Method	&Train	NTU-UTOI		
		P	R	F
RCNN	NTU-UTOI$_{tr}$	0.61	0.52	0.56
EAST [37]	NTU-UTOI$_{tr}$	**0.74**	0.50	0.60
SSTD [55]	NTU-UTOI$_{tr}$	0.59	0.34	0.43
Faster RCNN (*with objects*)	NTU-UTOI$_{tr}$	0.43	0.33	0.37
Faster RCNN (*with text only*)	NTU-UTOI$_{tr}$	0.63	0.55	0.58
TO-CNN (*without object*)	NTU-UTOI$_{tr}$	0.65	0.53	0.59
TO-CNN (*with object*)	NTU-UTOI$_{tr}$	0.70	**0.62**	**0.66**

&The subscripts *tr* and *a* indicate the corresponding training and entire dataset.

Table 6. Faster R-CNN fine-tuned on NTU-UTOI text dataset.

Anchors		COCO	NTU-UTOI
Same	**P**	0.38	0.43
	R	0.18	0.33
	F	0.24	0.37
Improved	**P**	0.52	**0.67**
	R	0.17	**0.50**
	F	0.26	**0.57**

Note:The networks were trained on COCO objects and NTU-UTOI objects, respectively and then fine tuned to our text dataset.

Fig. 8. Detection results of TO-CNN on the NTU-UTOI dataset.

Table 7. Performance of TO-CNN on NTU-UTOI with various anchors.

Settings	Parameters		P	R	F
	Scale	Ratio			
1 scale, 1 ratio	{8}	{1:1}	0.35	0.30	0.32
	{16}	{1:1}	0.34	0.28	0.31
1 scale, 5 ratio	{8}	{1:1,1:2,2:1,1:5,5:1}	0.65	0.53	0.58
	{16}	{1:1,1:2,2:1,1:5,5:1}	0.65	0.50	0.59
3 scale, 1 ratio	{8,16,32}	{1:1}	0.66	0.55	0.62
3 scale, 5 ratio	{8,16,32}	{1:1,1:2,2:1,1:5,5:1}	**0.70**	**0.62**	**0.66**

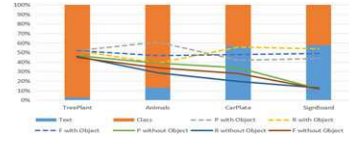


Fig. 9. Object dependence and performance analysis of TO-CNN.

RCNN and faster RCNN methods, which are the base of TO-CNN. It is also compared with the other state-of-the-art methods. For object dependency test, TO-CNN is also trained on text labels only (second last row). The experimental results show that without object information, TO-CNN and faster RCNN perform similarly. Training on images with object labels, TO-CNN outperforms RCNN, faster RCNN and TO-CNN without object information significantly. These results show clearly that objects contain valuable information for text spotting. The precision, recall and F-score for the 1st-3rd stages of TO-CNN are {0.59, 0.33, 0.42}, {0.65, 0.53, 0.59} and {0.70, 0.62, 0.66}, respectively. Some visual outputs of NTU-UTOI dataset are shown in Figure 8 that includes images taken in different environments and lighting conditions and proves that the

proposed algorithm works well in these cases. It even works well for dense text scenes, as shown in Figure 8.

To store object information in the network, the proposed algorithm combines two sub-networks. However, its size is not the largest one among the state-of-the-art text spotting networks. To further analyze how object information impacts on text detection, Figure 9 shows the percentages of four types of objects containing text and their corresponding recall and precision rates from NTU-UTOI testing set. TreePlant and Animals have negative dependence with text while CarPlate and SignBoard have positive dependence. For negative dependent objects, precision rates from TO-CNN perform better than its recall but for positive dependent objects, the recall rates are better. Note that the positive dependent objects degrade the network without object information a lot. It means that the text on objects is influenced by the objects. Note that in Figure 9, precision and recall rates are calculated based on the text and the selected object only showing their dependency on text and the selected objects. That is, if the total number of carplate images in the test set is considered to be 100% then the text overlapping is 57% leading to precision 34% and 41% without and with object information, respectively.

Catastrophic forgetting, which is a common problem in neural network, is not observed in our study. The experimental results in Table 5 show that the proposed algorithm does not suffer from such issues. The term *TO-CNN without object* in Table 5 means removing the object labels in the training set but keeping the same depth. We also tested two pre-trained models from faster RCNN and then fine-tuned on NTU-UTOI text data (Table 6). First was pre-trained on regular COCO objects and the other network was trained on NTU-UTOI dataset.

Lastly, to show significance of different scales and aspect ratios of RPN anchors, we experimented different anchor parameters on NTU-UTOI dataset, results shown in Table 7. According to this, improving anchors size and shape actually enhances the performance.

5 Conclusion

Traditionally, researchers solely used information in text for text spotting in natural scene images and objects in these images were totally neglected. Objects and text have in fact strong dependence. In this paper, TO-CNN with three training stages is proposed to exploit object information for text spotting. TO-CNN achieves comparable results to the state-of-the-art methods on COCO-Text, MSRA-TD500 and SVT. The experimental results show that object information is vital for improving text detection accuracy, in particular for recall rate. Currently, TO-CNN uses a linear network architecture. The authors will investigate other network architectures to exploit the object information more effectively and implement cluster-based RPN anchor selection.

Acknowledgement Authors would like to thank BAE Systems Applied Intelligence as this work is supported and funded by them under the research collaboration BAE-NTU fund at Cyber Security Research Centre @ NTU Singapore.

References

1. Tian, Z., Huang, W., He, T., He, P., Qiao, Y.: Detecting text in natural image with connectionist text proposal network. In: European Conference on Computer Vision, Springer (2016) 56–72
2. Zhang, Z., Zhang, C., Shen, W., Yao, C., Liu, W., Bai, X.: Multi-oriented text detection with fully convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 4159–4167
3. Yin, X.C., Pei, W.Y., Zhang, J., Hao, H.W.: Multi-orientation scene text detection with adaptive clustering. IEEE transactions on pattern analysis and machine intelligence **37**(9) (2015) 1930–1937
4. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Reading text in the wild with convolutional neural networks. International Journal of Computer Vision **116**(1) (2016) 1–20
5. He, T., Huang, W., Qiao, Y., Yao, J.: Text-attentional convolutional neural network for scene text detection. IEEE transactions on image processing **25**(6) (2016) 2529–2541
6. He, P., Huang, W., Qiao, Y., Loy, C.C., Tang, X.: Reading scene text in deep convolutional sequences. In: AAAI. (2016) 3501–3508
7. Busta, M., Neumann, L., Matas, J.: Fastext: Efficient unconstrained scene text detector. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1206–1214
8. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 2315–2324
9. Chen, X., Yuille, A.L.: Detecting and reading text in natural scenes. In: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. Volume 2., IEEE (2004) II–II
10. Zhong, Z., Jin, L., Zhang, S., Feng, Z.: Deeptext: A unified framework for text proposal generation and text detection in natural images. arXiv preprint arXiv:1605.07314 (2016)
11. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence **39**(6) (2017) 1137–1149
12. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE (2010) 2963–2970
13. He, W., Zhang, X.Y., Yin, F., Liu, C.L.: Deep direct regression for multi-oriented scene text detection. arXiv preprint arXiv:1703.08289 (2017)
14. Xiong, B., Grauman, K.: Text detection in stores using a repetition prior. In: Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on, IEEE (2016) 1–9
15. Rong, X., Yi, C., Tian, Y.: Unambiguous text localization and retrieval for cluttered scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 5494–5502
16. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. International journal of computer vision **104**(2) (2013) 154–171
17. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2014) 580–587

18. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. (2015) 1440–1448
19. Chen, H., Tsai, S.S., Schroth, G., Chen, D.M., Grzeszczuk, R., Girod, B.: Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In: Image Processing (ICIP), 2011 18th IEEE International Conference on, IEEE (2011) 2609–2612
20. Huang, W., Lin, Z., Yang, J., Wang, J.: Text localization in natural images using stroke feature transform and text covariance descriptors. In: Proceedings of the IEEE International Conference on Computer Vision. (2013) 1241–1248
21. Huang, W., Qiao, Y., Tang, X.: Robust scene text detection with convolution neural network induced mser trees. In: European Conference on Computer Vision, Springer (2014) 497–511
22. Yi, C., Tian, Y.: Text extraction from scene images by character appearance and structure modeling. Computer Vision and Image Understanding **117**(2) (2013) 182–194
23. Yi, C., Tian, Y.: Text string detection from natural scenes by structure-based partition and grouping. IEEE Transactions on Image Processing **20**(9) (2011) 2594–2605
24. Neumann, L., Matas, J.: Real-time scene text localization and recognition. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 3538–3545
25. Minetto, R., Thome, N., Cord, M., Leite, N.J., Stolfi, J.: Snoopertext: A text detection system for automatic indexing of urban scenes. Computer Vision and Image Understanding **122** (2014) 92–104
26. Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 1457–1464
27. Anthimopoulos, M., Gatos, B., Pratikakis, I.: Detection of artificial and scene text in images and video frames. Pattern Analysis and Applications **16**(3) (2013) 431–446
28. Posner, I., Corke, P., Newman, P.: Using text-spotting to query the world. In: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, IEEE (2010) 3181–3186
29. Wang, T., Wu, D.J., Coates, A., Ng, A.Y.: End-to-end text recognition with convolutional neural networks. In: Pattern Recognition (ICPR), 2012 21st International Conference on, IEEE (2012) 3304–3308
30. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision, Springer (2016) 21–37
31. Liao, M., Shi, B., Bai, X., Wang, X., Liu, W.: Textboxes: A fast text detector with a single deep neural network. In: AAAI. (2017) 4161–4167
32. Gidaris, S., Komodakis, N.: Object detection via a multi-region and semantic segmentation-aware cnn model. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1134–1142
33. Van de Sande, K.E., Uijlings, J.R., Gevers, T., Smeulders, A.W.: Segmentation as selective search for object recognition. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 1879–1886
34. Arbeláez, P., Pont-Tuset, J., Barron, J.T., Marques, F., Malik, J.: Multiscale combinatorial grouping. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2014) 328–335

35. Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: European Conference on Computer Vision, Springer (2014) 391–405
36. Liao, M., Shi, B., Bai, X.: Textboxes++: A single-shot oriented scene text detector. arXiv preprint arXiv:1801.02765 (2018)
37. Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., Liang, J.: East: an efficient and accurate scene text detector. arXiv preprint arXiv:1704.03155 (2017)
38. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
39. Park, E., Han, X., Berg, T.L., Berg, A.C.: Combining multiple sources of knowledge in deep cnns for action recognition. In: Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on, IEEE (2016) 1–8
40. Yang, J., Liu, Q., Zhang, K.: Stacked hourglass network for robust facial landmark localisation. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on, IEEE (2017) 2025–2033
41. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: European Conference on Computer Vision, Springer (2016) 483–499
42. Ma, J., Shao, W., Ye, H., Wang, L., Wang, H., Zheng, Y., Xue, X.: Arbitrary-oriented scene text detection via rotation proposals. arXiv preprint arXiv:1703.01086 (2017)
43. Veit, A., Matera, T., Neumann, L., Matas, J., Belongie, S.: Coco-text: Dataset and benchmark for text detection and recognition in natural images. arXiv preprint arXiv:1601.07140 (2016)
44. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision, Springer (2014) 740–755
45. Larsson, F., Felsberg, M., et al.: Using fourier descriptors and spatial models for traffic sign recognition. In: SCIA. Volume 11., Springer (2011) 238–249
46. Lyu, P., Yao, C., Wu, W., Yan, S., Bai, X.: Multi-oriented scene text detection via corner localization and region segmentation. arXiv preprint arXiv:1802.08948 (2018)
47. Kang, L., Li, Y., Doermann, D.: Orientation robust text line detection in natural images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 4034–4041
48. Yao, C., Bai, X., Liu, W., Ma, Y., Tu, Z.: Detecting texts of arbitrary orientations in natural images. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 1083–1090
49. Yin, X.C., Yin, X., Huang, K., Hao, H.W.: Robust text detection in natural scene images. IEEE transactions on pattern analysis and machine intelligence **36**(5) (2014) 970–983
50. Yao, C., Bai, X., Sang, N., Zhou, X., Zhou, S., Cao, Z.: Scene text detection via holistic, multi-channel prediction. arXiv preprint arXiv:1606.09002 (2016)
51. Shi, B., Bai, X., Belongie, S.: Detecting oriented text in natural images by linking segments. arXiv preprint arXiv:1703.06520 (2017)
52. Mao, J., Li, H., Zhou, W., Yan, S., Tian, Q.: Scale based region growing for scene text detection. In: Proceedings of the 21st ACM international conference on Multimedia, ACM (2013) 1007–1016
53. Zhang, Z., Shen, W., Yao, C., Bai, X.: Symmetry-based text line detection in natural scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 2558–2567
54. Bušta, M., Neumann, L., Matas, J.: Deep textspotter: An end-to-end trainable scene text localization and recognition framework. (2017)

55. He, P., Huang, W., He, T., Zhu, Q., Qiao, Y., Li, X.: Single shot text detector with regional attention. In: The IEEE International Conference on Computer Vision (ICCV). (2017)