Crimson Publishers
Wings to the Research

Research Article

Text Detection using Object Recognition Techniques

Siba Haidar[1]*, Ihab Sbeity[1] and Marwa Ayyoub[1]

[1]Faculty of Science, Beirut, Lebanon

Abstract

In this paper we propose an approach in text detection using object detection technique. Our approach is to deal with letters as objects. We use an object detection method, Retina Net deep leaning, to detect letters and recognize the text in natural scene images. The goal is to achieve high accuracy in text spotting, especially for curved text where the state-of-art methods fail. The Retina Net model architecture was used and modified in different ways to find the best performing model. Retina Net is implemented using Keras with Tensor flow backend. We prepare a model for training using the total-text dataset. The system manages to detect the letters as objects in the images. Then we perform letter candidate grouping to detect text based on distances between neighboring letters. The results exceeded our expectations. The training data was barely sufficient for some letters due to the differences in the frequency of their appearance in the dataset. Moreover, some specific letters, in specific orientations, were confusingly identified as others with a similar pattern. Nevertheless, we obtained a good performance on the test data for some classes, with a mAP of 40%. In order to further develop this method and to improve performance, more training data is needed, containing letters in different fonts. We also consider adding a dictionary to help correct or complete missing letters.

Keywords: Text detection; Object detection; Deep learning; Retina net; Total text

For HTML Version scan this QR code:

***[1]Corresponding author:** Siba Haidar, Faculty of Science, Beirut, Lebanon

Submission: February 19, 2019
Published: March 12, 2019
Volume 1 - Issue 1

How to cite this article: Siba H, Ihab S,Marwa A. Text Detection using Object Recognition Techniques. COJ Rob Artificial Intel. 1(1).COJRA.000502.2019.

Introduction

Recently, text detection and recognition became pragmatic research topics with quick advancement and growth in the world of computer vision. The widespread of smartphones and digital cameras provide us with an enormous number of images that contains text. Text in natural images gives precious information about the content of the image that helps in many vision-based applications. For example, protecting the pedestrians using smart transportation. Also, in blind navigation assistance systems, it is considered a useful component of navigation devices when it successfully recognizes the text on the street signs and shows directions for blind people.

At first, text recognition starts with documents of black text and white backgrounds where traditional techniques such as Optical Character Recognition (OCR) was used. OCR works well and gives high accuracy when dealing with regular fonts, standard sizes, and colors. Texts of natural scene images suffer from an extensive variety in backgrounds, fonts and sizes, low resolution and different typing directions which vary from vertical to horizontal, multi-oriented and curved text. These difficulties can affect both text detection and recognition stages as explained in [1].

State-of-the-art methods use convolutional neural networks (CNNs), recurrent neural networks with long short-term memory (RNN-LSTM) or the combination of them [1-9]. Many approaches presented rousing proposals and persuasive work that increase the performance and accuracy of text detection. However, challenges are still present and there is no one method that can solve all the problems together, specifically the curved text problem where most algorithms deal with horizontal text. And this limitation weakens these algorithms since the text in natural scene images is designed in different directions and orientations.

Object detection and recognition is one of the areas of computer vision that is growing quickly. Mostly object detection is related to detection of instances of semantic objects in computer vision and image processing. Object recognition allows robots and AI programs

to pick out and identify objects from inputs like video and camera images. Today, there is a plenty of pre-trained models for object detection such as YOLO, Fast R-CNN, and Retina Net that achieve high accuracy. We propose detecting text in natural scene images by applying one of the object detection techniques which is Retina Net dealing with letters as objects to get high accuracy comparing to the state-of-the-art of text detection techniques.

Related work

There are two main thoughts concerning the detection stage, character region and sliding window approaches. Then we will present character-based methods and word-based methods that are works done in the recognition stage.

Detection methods: In the pipeline of text spotting, the first step is to detect text. Character region-based detection which produces a segmentation of the detected text; and sliding window-based detection which results in bounding boxes fitting the text region are the two existing methods. Character-regions methods comprise of distinguishing associated pixels forming characters to deliver at that point character segmentation that is helpful for the recognition step. Many character-based strategies have been proposed in the writing [10-17]. The most significant works in the character-area identification extend are the Stroke Width Transform (SWT) [18] and the Maximally Stable Extremal Regions MSERs [19]. On the other hand, sliding window-based methods are a typical choice for character-region detection. They comprise sliding a classifier window through the image with a specific end goal to anticipate regardless of whether text exists. The sliding window is either considered at various scales or the image is down-examined to various scales in order to fit the text. Some relevant sliding window works [20-23] used boosted windows' classifiers of different scales [21], or hand-crafted features for a sliding random forest classifier [20]. Then, the sliding window produces a text saliency map. Then again in different approaches [22], the sliding window is trained on HOG features for characters detection. The characters are, at that point, assembled into words from a limited dictionary considering the spatial relationship between characters. High performance was shown in experiments with different standard datasets. CNNs were vastly used in different text detection stages [23-28] when Krizhevsky et al. [26] won the ILSVRC competition with a high margin. At a first stage, they extract text or character regions proposals, then improve those proposals to remove false positives and indicate the text components. While Wang et al. [23] trained a CNN to detect text regions, Huang et al. [17] utilized a CNN for the refinement stage and left the detection to a MSER [19] operator.

Recognition methods: Text recognition is the exact determination of the series of characters represented in a word image. Works at this stage can be categorized as character-based recognition or word-based recognition. Given segmented or bounded characters, character-based recognition depends on per-character classifiers. Several works have adopted this approach [25,29-34]. Wang et al. [22] managed the recognition stage in a per-character modeling fashion in their conclusion about text spotting system. Then, the word with the greater outcome in a list is selected, which keeps the recognition to a limited lexicon under control. Word-based recognition comprises of anticipating the image word among a list of a potential dictionary. This excludes predicting unclear words. In predicting word classification image features are directly considered. This approach is broadly embraced [16,28,30-33].

Contribution

We propose the use of the well-known method Retina Net, in the text recognition, considering text in text scenes as objects. We will experiment out proposal and compare it with different object detection techniques.

Retina net

Retina Net is a single, unified network composed of a backbone network and two task-specific subnetworks. The backbone is responsible for computing a conv feature map over an entire input image and is an off-the-self convolution network. The first subnet performs classification on the backbones output; the second subnet performs convolution bounding box regression. The two subnetworks feature a simple design that we propose specifically for one-stage, dense detection [8]. The one-stage network Retina Net was chosen as the model architecture since a state-of-the-art network was to be examined and since low prediction time is of interest. Since it generally has a lower prediction time than a two-stage detector. The model was implemented in Keras using Tensor flow as backend software. A GitHub repository was used and modified in the implementation [13]. According to [9], it is very hard to have a fair comparison among different object detectors. There is no straight answer on which model is the best. For real-life applications, choices can balance accuracy and speed. Besides the detector types, they considered other choices that impact the performance, such as the training dataset, localization loss function, boundary box encoding.

The technology advances so quick in a way that makes any comparison ends up out of date rapidly. Here, in [9] they condense the outcomes from several papers so it can be seen together. By showing different perspectives in a single setting, so that the execution scene better can be understandable. Retina Net builds on top of the FPN using Res Net. So, the high mAP achieved by Retina Net is the combined effect of pyramid features, the feature extractor complexity, and the focal loss. Yet, this is not an apple-to-apple comparison.

Dataset and training

We used the Total-Text Dataset, a dataset with horizontal, oriented, and curved text. Later we will show the preparation for the training and present the testing results. The curved text is commonly seen in real-world scenery such as business logos, signs, and entrances, but it has close to zero existence in available datasets. As a result, text detection algorithms with curved oriented text in consideration is rarely seen. So the introduction of total text dataset spurs an interest in the community of scene text understanding. The dataset features 3 different orientations: horizontal, multi-oriented, and curve. It has 1555 images in total,

with 11459 annotated text instances. It is available on the GitHub repository [14]. Familiar datasets such as ICDARs and MSRA TD500 have assumed a significance part in starting the impulse of scene text related research. The common point in all ICDARs versions is that the text is horizontal (Figure 1).

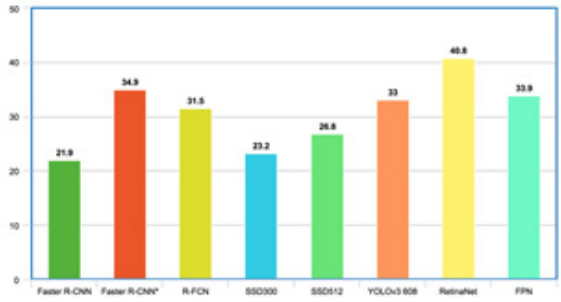

Figure1: Comparison between object detection methods in terms of Accuracy (mAP) [9].

In 2012, Yao et al. [6] introduced a new scene text dataset, namely MSRA TD500, that challenged the community with texts arranged in multiple orientations. MSRA's fame thus defined the tradition of 'multi-oriented' texts. Notwithstanding, a more intensive investigate the MSRA-TD500 dataset uncovered that most, if not all of the text are found in a straight line manner as in ICDARs. But still curved-oriented texts, in spite of its regularity, is absent from the case of study. So as not to deny, there is only one available dataset, CUTE80 [7] as scene text dataset to-date with curved text.

Datasets	No. of images	No. of Text Instances	Text instances per image	Horizontal Text	Multi-Oriented Text	Curve Text
ICDAR 2013	462	1943	4.2	✓		
ICDAR 2015	1670	11886	7.12	✓		
MSRA-TD500	500	1719	3.4		✓	
COCO-Text	**63686**	***173589***	2.73	✓	✓	
Total-Text	1555	11459	***7.37***	✓	✓	✓

Figure 2: Well-known scene text datasets compared to the total-text dataset.

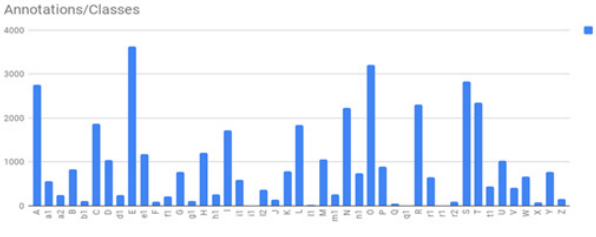

Figure 3: Number of annotations for each class.

However, its scale is too small with only 80 images and it has very minimal scene diversity. Following, in Figure 2 & 3; [14] shows a comparison between the well-known datasets already mentioned, and the Total-Text Dataset, according to the number of total images, text images and their orientations. Figure 4 presents some examples of images found in those datasets.

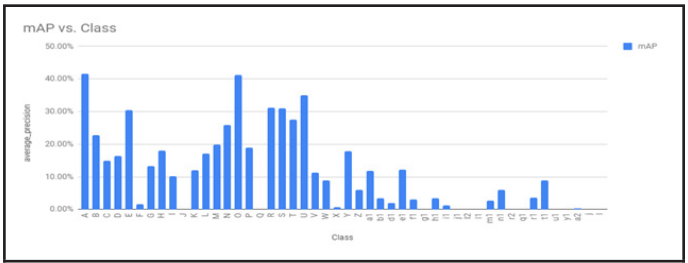

Figure 4: mAP Accuracy for classes.

Our training

As we mentioned above, since the Total-text dataset is plentiful with images in different directions and orientations and especially curved ones, we rely on it and use it in a different way. And since we are former to the thought of applying object detection tool to detect text in scene images, there is no pre-trained model to use it. So, we prepare our own model.

Model preparation

Training your own dataset in Retina Net requires two CSV files: data.csv and classes.csv. The data file is about the annotations we created and the classes file about a class name to ID mapping. The classes for our situation are the English Alphabets. We separate between capitalized letters, lowercase letters, and distinctive text styles for a few letters as they go in the images. Also, we join some letters capital ones and small ones for some letters considering they have the same pattern such as letter o, letters s, letters c. The final view for the classes file contains 43 classes and their mapping Id.

Dataset image labeling

After specifying the classes, we start in localization for the letters. In this step, we are responsible for putting a bounding box or drawing a red rectangle around the position of the letter in the image. Where the term localization refers to figuring out where in the picture is the specified letter we've detect. There might be multiple letters in the picture and we have to detect them all and localized them all. The classification of localization problem is when you're trying to recognize and localize objects, it has one big object in the middle of the image. In contrast, in the case of text localization and detection, the problem is that text is a set of words that contain many characters, so in the same image, we have to localize all letters that are related to different classes. So, we implement labeling tool that perform the mission of localizing the letters and form the data csv file that we need by auto saving the annotations.

Discussion of an ascertainment

When text localization finishes, the data file contains 40,926 records for all classes. One notices a numerical contrast between the instances of the classes; the classes of capitalized letters dominate because the images in the Total-text dataset are a collection of logos, brands, street signs, etc. which are almost all written in

capitalized manner. Also, as we know, vowel letters (A, E, O, U, I and Y) are naturally very frequent in the English Language. So according to this, the high numbers of some letters compared with others that are rarely used can be understood and justified.

Training procedure

The dataset is divided into two groups: training and testing. The network is trained on the training data. When satisfactory results are achieved by both the training data, the network is evaluated on the test data to get a final result of the performance of the network. The division of the dataset will typically be 80% training data and 20% test. All the networks were trained on Google Cloud Nvidia Tesla K80 GPU (Graphics processing unit) to improve processing power for deep learning tasks. The training was run for 50 epochs with 5000 steps per epoch. During training, the loss function, after several epochs, drops gradually to reach classification loss, of the order of ⟦6e⟧^(-4) and regression loss equal to 0.048 that is considered fairly satisfactory records.

Results and Discussion

In this section, we present our results and the mAP we obtain. Then we discuss our findings. The graph below presents the mAP of the trained model on testing images. The outcomes were evaluated using Retina Net evaluation protocol. The evaluation was built on the comparison of the obtained results with the labeled testing images.

Neighborhood function

Letter detection is the first step we obtain in the testing stage. Then we move to text recognition. We implement the neighborhood function that collects all detected letters and check the distance between them, so that character siblings at adjacent positions form a word. This method allows us to determine the text of the scene images. We calculate the centroid of each labeled box. Then we sort the detected letters on x-axis according to the x-coordinate of the centroid. After that, for each image, we calculate the distance between all letters in this image using the Euclidean distance:

$$\sqrt{(x2+x1)^2+(y2+y1)^2} \quad (1)$$

We then take the minimum distance as a threshold. Two adjacent characters should not be too apart from each other despite the variations of width, so the distance between two connected components should not be greater than the threshold plus the width of the character. The characters that fall within this range belong to the same word. This being said, let S the set of all predicted letters. Let a, b ∈ S be two predicted letters. Let C_(a) and C_(b)be the centers of the bounding boxes of the letters a & b respectively. We have:

$$N(a,b)=\{1 \text{ if } d(c_a,c_b)<\varepsilon\}$$
$$0 \textit{ otherwise} \quad (2)$$

Discussion

Figure 5: Examples of the testing results in the total-text dataset.

As we see in Figure 5, the most trained classes gave mAp accuracy about 40%. This is the same result obtained by Retina Net when applied on COCO dataset which surpass other object detection tools. For example, the mAp of letter A and letter O was about 40%, whereas mAp of letter E was about 30% since it is much complicated letter. Although the mAP of some classes was not high since it as rarely found in the training data such as letter J and letter Q almost 0% accuracy. Also, all classes of small letters was under 10% mAp due to dominance of capital letters in training data. Thus, the classifier failed in detecting some letters correctly, but we consider the results are satisfactory. Since there is no prepared dataset with focused text images in natural scenes that covers all fonts and designed to fit such work, not even the total-text dataset that we used. In order to be more accurate, the classifier ought to be trained on around 1000 images for each class. However, in the total-text dataset, we did not find such number for each class for the reasons already discussed earlier. Consider the following example, we test this image using our model and compare the result with [28,29].

As we have seen the results obtained by Retina Net shows high accuracy compared to other tools. There is few gaps, which can be filled, in future works, with the use of a dictionary for example. A smart dictionary that can check if the word is correct and give suggestions in case of misspelled words. It can also complete a missing letter or replace a wrong detected letter as in Figure 6. Moreover, the dictionary can solve the problem of similarity between some letters such as M and W, since they have same patterns in different orientations, and this may cause failure in classification. Furthermore, detecting letters is more complicated than objects due to similar patterns between letters and especially

in different rotations. Training the classifier so that it can recognize objects as humans see it and give us better results require much work (Table 1).

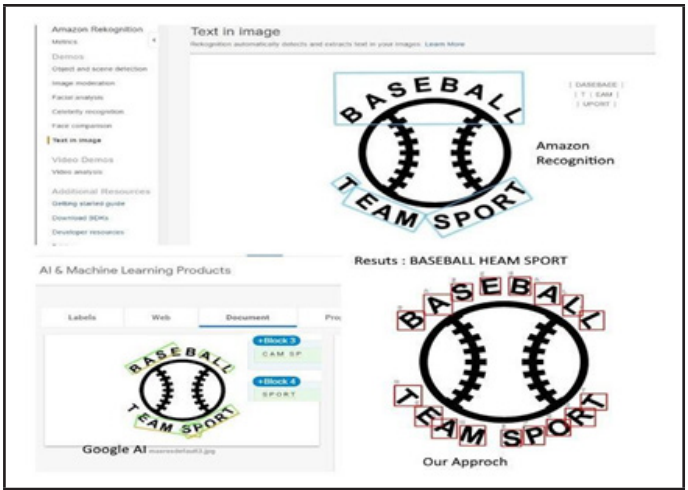

Figure 6: Comparing detecting curved text using our model and other known tools.

Table 1: Comparison between results of text detection.

Retina Net	Google AI	Amazon AI
Baseball	Cam SP	Dasebaee
Heam Sport	Sport	Team
		Uport

Conclusion

Text detection and recognition out of scene images are by inverse to it from images of a printed document, such as books, contracts, and magazines, a hard work. In this case, text may appear in any orientation and have more complex backgrounds. In this thesis, we proposed an innovative method, to detect text in natural images based on object detection and recognition method named Retina Net. Among several object detection techniques, Retina Net gave a high accuracy (mAP) in a relatively short time. We used the total-text dataset which is a dataset with horizontal, multi-oriented and curved text to prepare a training model. After training, we tested and evaluated our work and the results were very satisfactory and promising. The most trained letters got high accuracy compared to the others, an obvious and natural observation. This shows that if we train the letters more, the results will be more accurate.

Subsequently, we performed letters grouping in order to combine the candidate text letters into text strings. Text strings contain letter members that fall in the same neighborhood. Our text line grouping was able to extract text strings with arbitrary orientations. The combination of letter localization and text detection by neighborhood grouping gave the best performance, which outperformed the algorithms presented in Google AI and Amazon AI. This work will be extended by finding a new dataset covering most fonts with enough annotations for each class. Also, we will consider the integration of language dictionary to help predict and correct recognized strings. Finally, we can say that when searching for the state-of-the-art text detecting network one realizes that this area is under great development with new ideas, papers and improvements being released almost every day. It is a challenge just to keep up with all progress. The future of text detection and image analysis using machine learning will be an interesting topic to follow.

References

1. Yingying Z, Cong Y, Xiang B (2016) Scene text detection and recognition: recent advances and future text trends. Frontiers of Computer Science 10(1): 19-36.
2. Bolan S, Shijian L (2015) Accurate scene text recognition based on recurrent neural network Asian conference on computer vision connexis, Singapore, pp. 21-01.
3. Suman K, Ernest V, Andrew (2017) Visual attention models for scene text recognition. Media Integration and Communication Center (MICC) University di Firenze, Firenze, 1: 943-948.
4. Fei Y, Yi C, Xu Y, Cheng L (2017) Scene Text Recognition with Sliding Convolutional Character Models. National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, China.
5. Chng CK, Chan CS (2017) Total-text: A comprehensive dataset for scene text detection and recognition, 14th IAPR International Conference on Document Analysis and Recognition ICDAR, Japan.
6. Yao C, Bai X, Liu W, Ma Y, Tu Z (2012) Detecting texts of arbitrary orientations in natural images. CVPR, USA.
7. Risnumawan A, Shivakumara P, Chan CS, Tan CL (2014) A robust arbitrary text detection system for natural scene images. Expert Systems with Applications 41(18): 8027-8048.
8. Tsung L, Priya G, Ross B, Kaiming H, Piotr D (2017) Focal loss for dense object detection. ICCV.
9. Medium-a place to read and write big ideas and important stories.
10. Yao C, Bai X, Liu W, Ma Y, Tu Z (2012) Detecting texts of arbitrary orientations in natural images. IEEE Conference on Computer Vision and Pattern Recognition, USA, pp. 1083-1090.
11. Yao C, Bai X, Liu W (2014) A unified framework for multi oriented text detection text detection and recognition. IEEE Transactions on Image Processing 23(11): 4737-4749.
12. Huang W, Lin Z, Yang J C, Wang J (2013) Text localization in natural images using stroke feature transform and text covariance descriptors. IEEE International Conference on Computer Vision, Doi: 10.1109/ICCV.2013.157.
13. Tsung Y, Priya G, Ross G, Kaiming H, Piotr D (2017) Implementation of retina net object detection as described in focal loss for dense object detection.
14. (2017) Total text dataset-ICDAR.
15. Neumann L, Matas J (2012) Real-time scene text localization and recognition. IEEE Conference on Computer Vision and Pattern Recognition, pp. 3538-3545.
16. Novikova T, Barinova O, Kohli P, Lempitsky V (2012) Large-lexicon attribute-consistent text recognition in natural images. 12th European Conference on Computer Vision, pp. 752-765.
17. Huang W, Qiao Y, Tang X (2014) Robust scene text detection with convolution neural network induced Mser trees. European Conference on Computer Vision, pp. 497-511.
18. Epshtein B, Ofek E, Wexler Y (2010) Detecting text in natural scenes with stroke width transform. IEEE Conference on Computer Vision and Pattern Recognition, pp. 2963-2970.

19. Neumann L, Matas J (2011) A method for text localization and recognition in real-world images. Lecture Notes in Computer Science. Asian Conference on Computer Vision, pp.770-783.

20. Anthimopoulos M, Gatos B, Pratikakis I (2013) Detection of artificial and scene text in images and video frames. Pattern Analysis and Applications 16(3): 431-446.

21. Posner I, Corke P, Newman P (2010) Using text-spotting to query the world. IROS, Taiwan.

22. Wang K, Babenko B, Belongie S (2011) End-to-end scene text recognition. International Conference on Computer Vision, IEEE, Spain, pp. 1457-1464.

23. Wang T, Wu DJ, Coates A (2012) End-to-end text recognition with convolutional neural networks. IEEE, pp. 3304-3308.

24. Zhang Z, Shen W, Yao C, Bai X (2015) Symmetry-based text line detection in natural scenes. IEEE Conference on Computer Vision and Pattern Recognition, China, pp. 2558-2567.

25. Jaderberg M, Vedaldi A, Zisserman A (2014) Deep features for text spotting. European conference on computer vision, Europe.

26. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. NIPS, pp. 1106-1114.

27. Qin S, Manduchi R (2016) A fast and robust text spotter. IEEE Winter Conference on Applications of Computer Vision, USA.

28. Jaderberg M, Simonyan K, Vedaldi A (2016) Reading text in the wild with convolutional neural networks. International Journal on Computer Vision, USA.

29. Yao C, Bai X, Shi B, Liu W (2014) Strokelets: A learned multi-scale representation for scene text recognition. IEEE Conference on Computer Vision and Pattern Recognition, USA, pp. 4042-4049.

30. Mishra A, Alahari K, Jawahar C V (2012) Top-down and bottom-up cues for scene text recognition. IEEE Conference on Computer Vision and Pattern Recognition, USA.

31. Jaderberg M, Simonyan K, Vedaldi A, Zisserman A (2014) Synthetic data and artificial neural networks for natural scene text recognition. ArXiv, USA.

32. Alsharif O, Pineau J (2014) End-to-end text recognition with hybrid HMM maxout models. International conference on learning representations, USA.

33. Mishra A, Alahari K, Jawahar CV (2012) Scene text recognition using higher order language priors. 23rd British Machine Vision Conference, USA.

34. Goodfellow IJ, Bulatov Y, Ibarz S, Shet V (2014) Multi-digit number recognition from street view imagery using deep convolutional neural networks, USA.