Influence of quality of pixel level annotations on text detection performance in natural images

I. Dorkić*, M. Brisinello*, R.Grbić** and M. Herceg**
* TTTech Auto Central and Eastern Europe, Osijek, Croatia
** Faculty of Electrical Engineering, Computer Science and Information Technology Osijek, Osijek, Croatia
ivan.dorkic@tttech-auto.com, matteo.brisinello@tttech-auto.com, ratko.grbic@ferit.hr, marijan.herceg@ferit.hr

***Abstract* - Text detection in natural images is a task that arises in many computer vision applications. State-of-the-art text detection methods are mainly based on deep neural networks designed for instance segmentation task. However, most of the available datasets for text detection do not have fine annotations at the pixel level which are required during supervised learning of such networks. Usually, a whole or reduced text bounding box is used as a segmentation mask. In this paper, a method that generates a synthetic dataset with precise annotations at the pixel level is proposed. The method is based on the available Synthtext script for generating synthetic datasets with text instances. By creating synthetic datasets with precise and coarse annotations at the pixel level we explore the efficiency of the state-of-the-art text detector TextFuseNet.**

Keywords* - *object detection; text detection; instance segmentation; Synthtext; TextFuseNet

I. INTRODUCTION

The text represents a set of words that conveys a certain meaning to the reader. In our daily life, we often encounter text while reading newspapers, emails, information and traffic signs, street names, and so on. Text detection and recognition are for an adult person relatively simple task. Reading a text with a computer is successfully solved within domains where the structured text appears like in digitalized books, official documents, license plates, and so on. Such text is usually on a legible background, with a standard font and line spacing and horizontally aligned with no occlusions. However, text can appear in so-called unstructured form - we find it in random places in natural images. Typical examples are scenes of the streets or shops where text appears in posters, advertisements, cafe names, products, and so on. Such text instances can have non-standardized font and spacing, can be in mixed colors, multi-oriented, occluded, curved, and with a colorful background. Detection of such text in natural images is still quite challenging for computer algorithms.

Nowadays, text detection in natural images is based on deep Artificial Neural Networks (ANNs). The most effective methods [1] are based on deep Convolutional Neural Networks (CNNs) that perform instance segmentation task. To successfully learn such a network in a supervised manner, an appropriately annotated dataset is required which means that each text instance in the image is annotated at a pixel level. However, most of the available datasets provide only a bounding region of a text instance in form of a rectangle or polygon, and usually, every pixel within such bounding region is considered a part of the text by the applied method for text detection [2]. In the rest of this paper, we call such pixel-level annotations coarse. We believe that more precise text instance annotations on a pixel level of the training dataset can potentially boost text detector efficiency.

To explore whether more precise annotations at the pixel level of the training dataset can boost text detector efficiency, we propose an approach for obtaining precisely annotated synthetic datasets for text detection. The available code from [3,4] is used to generate synthetic images where the text appears in natural images. The code is further modified to produce precise annotations on a pixel level in form of a segmentation mask or polygon for each character and whole text instance. Such datasets are then used to train state-of-the-art text detector TextFuseNet [1] to quantify the importance of precise annotating of text instances in the training dataset.

The paper is structured as follows. In Section II, a problem of text detection and text annotation is presented, followed by an overview of existing datasets for text detection. After that, an overview of the most recent solutions for text detection is given. A process of creating datasets with precise and coarse annotations on a pixel level is given in Section III along with the process of text detector training. The obtained results on the created datasets and the accompanying discussion are given in Section IV. At the very end of the paper, a conclusion is given.

II. RELATED WORK

Most text detection methods, at least the most effective ones, are based on deep neural networks for instance segmentation task like the well-known Mask RCNN [5]. These networks are further modified to take into account text appearance in natural images and are trained in a supervised manner based on available text detection datasets. Most of the datasets do not provide the fine annotations of each character or the whole text instance at the pixel level (in form of a mask or a polygon). In theory, the most precise way of specifying a text mask would result in a mask consisting of pixels within the bounding box that exactly belong to the text in the image. An example of an image with several text instances is shown in Fig. 1 a) while the difference between the commonly

used coarse approach (Fig.1 b)) and the precise labeling approach (Fig.1 c)) is clearly visible. Most of the instance segmentation methods based on Mask R-CNN solve this problem by treating all pixels within the bounding box as a segmentation mask during training.

A. A brief overview of available datasets for text detection

The ICDAR 2013 dataset [2] consists of 229 training images and 233 testing images. All images are of different resolutions. It is a set in which all the images are photographed with a camera and the text on them is natural, i.e. it is not in focus. It is a standard reference dataset for horizontal text detection.

Synthtext in the wild [3] is a synthetically generated dataset in which words are placed in images of natural scenes. The text instances are placed in an image by considering the outline of the scene and the depth map of the scene. In this way, the text was added to sufficiently large areas. It consists of 800,000 images with approximately 8 million words artificially added to the images. The images are of different resolutions. Each text instance is annotated with word-level and character-level bounding boxes.

The COCO_Text dataset [6] contains 63,686 images with 145,859 text instances, where words are annotated by using polygons. It is the largest real-world dataset of images with text appearance in a natural form. It consists mostly of horizontal and multi-oriented text instances which are rotated by a certain angle. A small part of the images also contains curved text.

The Multi-Lingual scene Text (MLT-2019) dataset [7] contains 20,000 images. The images are of different resolutions. The text in the images is displayed in its natural form. Annotation of the dataset contains word level and character level text bounding boxes along with the corresponding transcription and language class. It contains text in 10 different languages.

The Incidental Scene Text (ICDAR 2015) dataset [8] contains 1,670 images and 17,548 annotated regions. This dataset contains images in which the text is not in the focus. Therefore, most of the text instances in the images are out of focus and blurry. Text instances are annotated with word-level bounding boxes in form of quadrilaterals.

B. A brief overview of text detection methods

In [9], a method called Mask TextSpotter is proposed, which is an end-to-end neural network for spotting text with arbitrary shapes. Mask TextSpotter uses a simple end-to-end learning procedure that can achieve the detection and recognition of text directly from two-dimensional space using semantic segmentation. This method is efficient in detecting instances of irregular shape text, such as curved text. It is evaluated on four English datasets and one multi-language dataset, achieved results are much better than other similar methods.

Most existing solutions for text detection in images are based on neural networks primarily intended for instance segmentation. In [1], a method called TextFuseNet is proposed, which is mainly based on the architecture of Mask R-CNN and TextSpotter, which uses richer fused features to detect text. TextFuseNet collects and fuses text features from different levels using a multi-path fusion architecture that can efficiently match and fuse different representations. TextFuseNet detects text at the word and character level. During training, TextFuseNet uses a more precise polygon segmentation mask. The text instance annotation is a "narrowed" bounding box compared to the approach of most other methods, where everything inside the bounding box is considered a segmentation mask. The detection success measured with F1 score on the ICDAR2013 dataset is 94.3%, and on the ICDAR2015 dataset it is 92.1%

It appears that one problem in building efficient object detectors is the lack of large data sets that have precisely annotated text at the pixel level. In [10], a method for annotating text on images at the level of pixels is proposed. The previously mentioned COCO_Text data set was used. From the images in that set, 1,000,000 text images per border of the rectangular bounding box are cropped. These images were then used to train a deep neural network that is used for text segmentation, that is, the background is separated from the text itself using this network. After the training phase, two thresholds were determined. If the probability is less than 0.3, that part of the image is considered the background, if the probability is greater than 0.7, the image contains text, and if the probability is between these two thresholds, the network is unsure of its decision. Using this method, 14,690 images were generated where text is precisely annotated at the pixel level.

a) b) c)

Figure 1. Examples of annotated images: a) input image with text bounding boxes, (b) coarse segmentation mask, (c) precise segmentation mask.

III. CREATING PRECISELY ANNOTATED SYNTHETIC DATASETS

This section describes the proposed approach for the precise annotating of text instances in synthetic natural images. For this purpose, we used an existing script available at [4] which was used for developing a famous Synthtext in the wild dataset. Based on created synthetic datasets we trained and tested TextFusenet text detector.

The code from [4] produces natural images where text instances are placed at suitable locations in the image. The script for each text instance provides coordinates of the text bounding box and information which image pixels correspond to the characters of the text. The example of generated text instance is shown in Fig. 2 where annotation at a pixel level provided by the script [4] is overlaid in red color. It can be noticed that generated annotation is quite imprecise. Certain pixels that belong to the background are annotated as text, so the actual shape of each character is not clearly visible from the generated mask.

To annotate the text instance more precisely at the pixel level we first extract the text instance from the image based on the provided text bounding box. Then *k*-means clustering algorithm [11] is applied to the extracted image to perform color segmentation. By doing so the segmentation of the image is performed based on the color similarity of the pixels. *K*-means clustering is performed in RGB color space. The resulting centers of *k*-means algorithm represent the dominant colors in the image. Since every character of each generated text instance is always in the same color, the assumption is that the pixels that belong to the text will be assigned to a single group and all other pixels (e.g. background) to other groups. In this paper, the optimal number of clusters was determined experimentally, and it is four. If less than four clusters are used, the text wouldn't always be segmented properly. Segmentation of the image shown in Fig. 2 in four clusters by *k*-means algorithm is shown in Fig. 3. In this case the first cluster represents pixels belonging to text (Fig. 3 a)) while the remaining clusters represent the background and text border. Obviously, cluster representing text can be used as a more precise text mask than the annotation produced by the script [4] which is shown in Fig. 2. Furthermore, since script [4] provides rectangle bounding boxes on a character level also, we can easily extract precise mask for each character of a text instance from the obtained cluster representing text.

Since the result of the k-means algorithm depends on the initial initialization of the centers and the appearance of the image itself, it is not possible to know in advance in which group the pixels belonging to the text will be assigned. For this purpose, we developed CNN that performs binary classification whether the image contains text or not. In that way, we can determine which cluster actually contains pixels belonging to the text, i.e. required precise mask of the text. The CNN structure is shown in Table I. It is based on convolutional and max pooling layers. An input image is scaled to the size of 64x64 pixels. Dropout layer is used to prevent overfitting. To train such CNN, 812 different images containing text instances (like Fig. 4) were segmented with *k* equal to four

Figure 2. Example of generated text instance with overlaid annotation in red color produced by script [4].

Figure 3. Segmentation of the image shown in Fig. 2 using *k*-means algorithm in four clusters.

resulting in 3,248 images (similar to Fig. 3). The resulting images were manually labeled to class 1 or 0. It is necessary to do so because images containing text are considered positive examples, while images without text are considered negative examples. Several examples of such images are shown in Fig. 4. The images were divided into a training set, a validation set, and a test set. The training dataset contains 512 images with text and 1536 images without text. The validation set contains 50 images with text and 150 without text. The test set contains 250 images with text and 750 without text. The number of training epochs was 40 and the batch size was 32. During learning online augmentation was used such as rotation, horizontal flip, vertical flip, zoom, etc.

The evaluation of the built CNN is shown in Table II using the confusion matrix on the test dataset with 0.5 threshold applied to network output. The built CNN shows accuracy on the test of over 99%, with precision equal to 97.6% and recall equal to 98.4%.

The built CNN was implemented along with *k*-means algorithm in the script [4] to produce precise annotation at the pixel level for each text instance during the generation process (on-the-fly). Based on the resulting mask, a polygon or RLE record is determined for each text instance and saved to the JSON file containing annotations in COCO style [12]. Although the RLE notation is somewhat more precise, in this paper we used the polygon annotations for pixel-level segmentation since the default implementation of TextFuseNet requires segmentation annotation in form of polygons. Since the script from [4] provides a bounding box for each character of the text instance, we simply annotated each character at pixel level by extracting only part of the text mask that intersects with the character bounding box.

The code from [4] script uses a database of 8000 images but during a single run over 7000 images with

TABLE I. CNN STRUCTURE FOR BINARY CLASSIFICATION OF IMAGES CONTAINING TEXT OR NOT .

Layer	Input size	Output size
InputLayer	(64,64,3)	(64,64,3)
Conv2D	(64,64,3)	(64,64,32)
Conv2D	(64,64,32)	(62,62,32)
MaxPooling2D	(62,62,32)	(31,31,32)
Droput	(31,31,32)	(31,31,32)
Conv2D	(31,31,32)	(31,31,64)
Conv2D	(31,31,64)	(29,29,64)
MaxPooling2D	(29,29,64)	(14,14,64)
Droput	(14,14,64)	(14,14,64)
Flatten	(14,14,64)	(12544)
Dense	(12544)	(512)
Droput	(512)	(512)
Dense	(512)	(2)

TABLE II. CONFUSION MATRIX OBTAINED ON TEST DATASET

		Predicted	
		Text	*Not text*
True	*Text*	244	6
	Not text	4	746

a) b) c) d)

Figure 4. Examples of training images obtained by *k*-means clustering, a) - b) images containing text, c) - d) images without text.

random text instances are generated because some images do not always provide a convenient place to add text instance of a certain size and length. A typical example of the generated image is visible in Fig. 5 and a single text instance with precise and coarse annotations at the pixel level is shown in Fig. 6 a) and b).

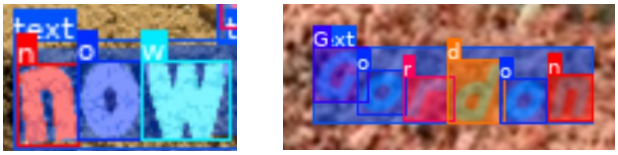
In the end, six different training datasets were generated with varying numbers of training images (around 1000, 3000, and 5000) and with different types of annotations at the pixel level (precise or coarse). The generated datasets are summarized in Table III.

IV. RESULTS AND DISCUSSION

This section presents the results of TextFuseNet detectors training and testing on the generated datasets. A PC with a Linux operating system (Ubuntu 20.04. LTS), NVIDIA GeForce RTX 3080Ti graphics card, and Intel Core i9-11900F processor with 32 GB of RAM was used for training and testing. The training comprised of 60 epochs and the model with the lowest validation loss was selected for further evaluation.

All built detectors were evaluated on the same test data set with respect to text instances detection. COCO evaluation metrics are used for built detectors evaluation: average precision over 10 IoU thresholds from 0.5 to 0.95

Figure 5. Examples of the generated image with precise annotations using modified *Syntext* script.



a) b)

Figure 6. Examples of text instance with a) precise annotation, b) coarse annotation.

TABLE III. GENERATED SYNTHETIC DATASETS

Dataset name	Number of images in each dataset
Precise_1k	1029
Precise_3k	3265
Precise_5k	5099
Coarse_1k	1029
Coarse_3k	3265
Coarse_5k	5099
Validation	1038
Test	1032

with a step size of 0.05 - **AP**, average precision at IoU equal to 0.5 – $\mathbf{AP^{50}}$, average precision at IoU equal to 0.75 – $\mathbf{AP^{75}}$, average precision for small objects with an area less than 32^2 – $\mathbf{AP^S}$, average precision for medium objects with an area between 32^2 and 96^2 – $\mathbf{AP^M}$, and average precision for large objects with an area greater than 96^2 – $\mathbf{AP^L}$.

The evaluation results of TextFuseNet detectors which were trained on precisely and coarsely annotated datasets with a varying number of training images are presented in Table IV and Table V. In both tables, it can be observed that metrics slightly increase with the increase of the number of training images which was expected. AP^{50} values are only slightly lower than AP^{75} indicating good localization properties of all detectors. All models have high values of average precision for medium and large objects but struggle with small ones. If we compare AP^{50} value of detectors trained on precisely and coarsely annotated datasets, then it can be noticed that the obtained values are only slightly in favor of more precise

TABLE IV. TEST RESULTS OF TEXTFUSENET DETECTORS TRAINED ON PRECISELY ANNOTATED DATASETS

Training dataset	AP	AP^{50}	AP^{75}	AP^{S}	AP^{M}	AP^{L}
Precise_1k	73.02	87.05	82.77	66.92	81.65	86.82
Precise_3k	74.75	88.04	83.85	68.58	83.18	88.23
Precise_5k	75.49	88.03	84.86	69.40	83.92	89.75

TABLE V. TEST RESULTS OF TEXTFUSENET DETECTORS TRAINED ON COARSELY ANNOTATED DATASETS

Training dataset	AP	AP^{50}	AP^{75}	AP^{S}	AP^{M}	AP^{L}
Coarse_1k	73.32	87.04	82.71	63.55	81.12	89.04
Coarse_3k	74.25	87.08	83.79	64.49	82.43	89.22
Coarse_5k	74.96	87.08	83.92	65.09	83.04	89.79

Figure 7. TextFuseNet output for a single test image.

annotating process. However, further analysis of average precision for different size of text instances shows certain benefits when annotating instances more precisely in the case of small and medium-sized objects. It is especially pronounced for small objects where AP^{S} is around 6% higher for detectors trained on precisely annotated datasets. Interestingly, AP^{L} is slightly decreased when the detectors are trained on precisely annotated datasets. This suggests that different strategies can be used when data annotating concerning the object size for the best TextFuseNet detector performance. In Fig. 7. the detector output for one test image is displayed.

V. CONCLUSION

In this paper, text detection in natural images is studied in more detail regarding the quality of annotations in the training dataset. A method that produces fine annotations on a pixel level is proposed and implemented within available Synthext in the wild script to obtain precisely annotated synthetic datasets for text detection. Precisely and imprecisely datasets are used to train text detectors based on TextFuseNet. The obtained results suggest that average precision on IoU equal to 0.75 does not significantly increase if precise annotations at pixel level are used. However, average precision for small and medium-sized objects increases, while average precision for large objects decreases.

Future work will include pretraining of TextFuseNet with precisely annotated synthetic datasets and then training on available real-world datasets. Additionally, modifications to TextFuseNet will be performed to use masks as annotations rather than polygons annotations. Furthermore, we believe that the recognition process that should be performed after text detection can benefit when detectors are trained with precisely annotated text instances, and this will be explored in more detail.

REFERENCES

[1] J. Ye, Z. Chen, J. Liu, and B. Du, "TextFuseNet: Scene Text Detection with Richer Fused Features," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, Yokohama, Japan, Jul. 2020, pp. 516–522. doi: 10.24963/ijcai.2020/72.

[2] D. Karatzas *et al.*, "ICDAR 2013 Robust Reading Competition," in *2013 12th International Conference on Document Analysis and Recognition*, Washington, DC, USA, Aug. 2013, pp. 1484–1493. doi: 10.1109/ICDAR.2013.221.

[3] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic Data for Text Localisation in Natural Images," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2315–2324. doi: 10.1109/CVPR.2016.254.

[4] A. Gupta, "SynthText." Aug. 23, 2022. Accessed: Aug. 23, 2022. [Online]. Available: https://github.com/ankush-me/SynthText

[5] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Oct. 2017, pp. 2980–2988. doi: 10.1109/ICCV.2017.322.

[6] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie, "COCO-Text: Dataset and Benchmark for Text Detection and Recognition in Natural Images." arXiv, Jun. 19, 2016. Accessed: Aug. 23, 2022. [Online]. Available: http://arxiv.org/abs/1601.07140

[7] N. Nayef *et al.*, "ICDAR2019 Robust Reading Challenge on Multi-lingual Scene Text Detection and Recognition -- RRC-MLT-2019." arXiv, Jrul. 01, 2019. Accessed: Nov. 10, 2022. [Online]. Available: http://arxiv.org/abs/1907.00945

[8] D. Karatzas *et al.*, "ICDAR 2015 competition on Robust Reading," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, Aug. 2015, pp. 1156–1160. doi: 10.1109/ICDAR.2015.7333942.

[9] M. Liao, P. Lyu, M. He, C. Yao, W. Wu, and X. Bai, "Mask TextSpotter: An End-to-End Trainable Neural Network for Spotting Text with Arbitrary Shapes." arXiv, Aug. 22, 2019. Accessed: Dec. 13, 2022. [Online]. Available: http://arxiv.org/abs/1908.08207

[10] S. Bonechi, M. Bianchini, F. Scarselli, and P. Andreini, "Weak supervision for generating pixel–level annotations in scene text segmentation," *Pattern Recognit. Lett.*, vol. 138, pp. 1–7, Oct. 2020, doi: 10.1016/j.patrec.2020.06.023.

[11] "What is K-Means Clustering? - Definition from Techopedia," *Techopedia.com*. http://www.techopedia.com/definition/32057/k-means-clustering (accessed Aug. 23, 2022).

[12] S. Bonechi, P. Andreini, M. Bianchini, and F. Scarselli, "COCO_TS Dataset: Pixel-level Annotations Based on Weak Supervision for Scene Text Segmentation," vol. 11730, 2019, pp. 313–325. doi: 10.1007/978-3-030-30490-4_26.