Synthetic Images Generation for Text Detection and Recognition in the Wild

Natalia Khanzhina, Natalia Slepkova, Andrey Filchenkov
Machine Learning Lab, ITMO University, 49 Kronverksky Pr., St. Petersburg 197101, Russia

ABSTRACT

Deep neural networks help solving different images related tasks very efficiently, though their cost is high whereas a lot of data are required for training. While there is a great demand to build neural network models for optical character detection and recognition for different languages, such as, for mobile real-time applications, datasets collecting and labeling are quite expensive. In this paper, we propose the fully automated approach for synthetic images with text generation based on deep learning and projective geometry methods. For evaluation, we trained two neural networks on the dataset generated by our algorithm. Our approach enables to decrease the false negative rate on real images from SVT and SVT-50 datasets in comparison with training on SynthText dataset, giving ~1% of F_1-measure increasing.

Keywords: Image generation, neural networks, optical character recognition, semantic segmentation, text detection, text localization.

1. INTRODUCTION

Optical character recognition is an important task in many application areas including street scenes processing such as recognition of store signs, caution plates for further machine translation, license plates etc. To solve this problem, there are many classical old approaches, such as character recognition using the curvature scale space [1, 2], text recognition based on cellular automata [3], using Voronoi diagrams [4] etc. However, deep neural networks became one of the most powerful approaches for solving this problem due to growth of machine learning. According to contests reports, neural networks achieved 89.84% of F-measure for text localization and 65.33% of F-measure for end-to-end recognition [5]. Nevertheless, deep neural networks training requires large amounts of data, which may be quite difficult to find especially for a particular recognition task related to low-resource language. Artificial generation of realistic images with text can solve such data shortage problem.

One possible algorithm for this goal was proposed in article [6] called SynthText, which is commonly used these days to increase the number of images in training set. It allows generating images with text based on background source images (without text). The disadvantage of this approach is that the text can be projected anywhere, on any objects, such as the sky, grass, animals. Because of this, the resulting images do not look realistic. As a rule, the neural network model trained on such data can mistakenly find a text on these objects, thus increasing the false positive rate.

In this paper, we present an algorithm that is able to generate an image with text given and image and a text that should be placed there. There main advantage of this approach is that it avoids the generation of text on the semantically inappropriate parts of the image. This provides more realistic data and models trained in these images show less false positive rate on real images.

The rest of the paper is organized as follows. The related works are overviewed in Section 2. The approach steps are described in detail in Section 3. The experiments are described and their results are presented and discussed in Section 4. Section 5 contains the conclusion of the paper.

2. RELATED WORK

The synthetic image generation often addresses the lack of datasets problem for different tasks on images. For instance, authors [7, 8] tried to superimpose small objects to indoor scenes using masking, selective positioning and blending algorithms to improve the score of image detection task, while in work [9] authors performed similar approach for semantic segmentation task improvement. In [10, 11], basic computer vision techniques were used for text pasting

into random background image for text recognition task dataset augmentation. The background and text separation were also performed in [12] to generate the structured historical documents. In paper [13], its authors performed a successful attempt of using generative adversarial networks for vehicles license plate generation to solve the automatic license plate recognition task.

The work [6] is the only one that we found the most relevant to our research. The key point of this paper for our research is fully automatic pipeline for text placing on a wild scene image that can be used afterwards for both further text localization and detection.

The authors used Newsgroup20 dataset [14] for text extraction. Single words, sentences and whole paragraphs from this dataset were used as well. Images from Google Image Search (over 8000) were used as background for text projection. To avoid double superimposition of text, only images that do not contain text by default were picked.

Every image is divided into regions where the text is to be placed. The regions are extracted using the edge detection and the hierarchical image segmentation [15]. After obtaining a hierarchical segmentation, it is necessary to select those regions that are large enough for the text projection on them to be legible and visible.

Based on each pixel coordinates (x',y',z'), one can determine a plane that corresponds to each of the remaining regions. However, this computation is not accurate enough because of the average, not precise focal distance used, which leads to a number of outliers. To prevent them from having a significant impact on the resulting planes set, the RANSAC algorithm is used [16]. The planes that are at an angle close to $90°$ to the plane of the image itself are eliminated, since the text superimposed at such an angle is illegible and unrecognizable.

Next, two homography matrices H_1 and H_2 are calculated for the remaining planes, where H_1 is a transfer matrix to a plane parallel to the image, H_2 is inverse transfer matrix. After transferring the region to a plane parallel to the image, the smallest quadrilateral polygon containing all points of the area is calculated. Then the text is superimposed along the largest side of each one of these picked polygons. To make the text look more realistic on new background, the Poisson method is used [17]. This method takes into account the background image texture and changes the text color according to it. Then, using the H_2 transformation, the text area is transferred back to the image plane. Finally, the text is projected into one segment of the original image.

The problem with this algorithm is that the text can be projected even on those areas that in real life can never contain the text, for example, on the faces of people, the sky, the grass, on the buildings' windows, animals etc (Fig. 1). This can cause false positives, which can be avoided by generating data using semantics.

Figure 1. The example of the wrong text projection of the SynthText dataset. On the right part of the image the text is projected on the sky.

For example on Fig. 2, the classification of the text on the image by a neural network trained on a dataset obtained after application of the algorithm [6] is presented. One can see that the area on the right part of the image is falsely classified as a text.

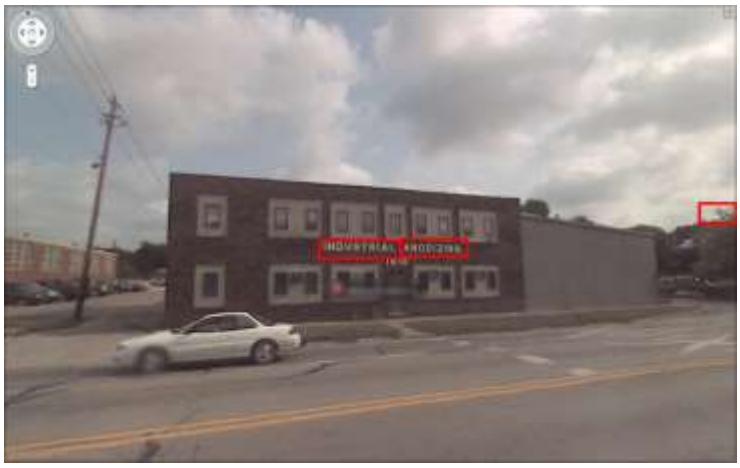

Figure 2. The example of false positive error. On the right part of the image, the sky and tree are detected as text.

3. THE APPROACH

To address the problem of the approach [5], we propose following pipeline of our method:

a. Semantic segmentation of the image for determining of areas suitable for text occurrence (Section 3.1);
b. Depth map prediction using *monoDepth* neural network for picked after semantic segmentation areas [18];
c. Image coordinates transformation using obtained depth map (Section 3.2);
d. Plane calculating for each area (Section 3.4);
e. Filtering of remaining areas by its aspect ratio etc. (Section 3.3);
f. Random selection of text color;
g. Text projection using projective geometry methods (Section 3.4);
h. Text blending using Poisson method [17].

3.1 Semantic segmentation

Image semantic segmentation is the first step of our approach. The input image is divided into areas where the text would look less natural and more natural: pavement, car bodies, walls of building, road signs etc. The base dataset for this task is Cityscapes dataset [19]. This dataset contains images of streets taken with the car digital video recorder (DVR). We chose it due to the labelled classes that include the ones where the text usually occurs: road, sidewalk, parking area etc.

Cityscapes are originally labelled on approximately 35 classes that could most likely be found on the streets of the city (see Table 1). Therefore, to use this dataset for our goal it is necessary to relabel it, that is, to merge the classes inappropriate for the text projection in one class and leave the rest unchanged. Table 1 shows all the classes that occur in the Cityscapes dataset. The classes where the text can be projected, and it would look realistic are marked with bold italics. The remaining classes after relabeling represent one large class.

Table 1. The Cityscapes dataset classes and its categories

Category	Classes
flat	***road***, ***sidewalk***, ***parking***, rail track
human	person, rider
vehicle	car, truck, bus, caravan, trailer, train, motorcycle, bicycle, license plate

construction	***building***, ***wall***, fence, guard rail, bridge, tunnel
object	pole, pole fender, traffic light, traffic sign
nature	vegetation, terrain
sky	sky
void	ground, unlabeled

To perform the segmentation the neural network with the U-Net architecture was used [20], trained on the modified Cityscapes dataset. Thus on the next steps the text is projected only on the semantically suitable areas.

3.2 Image coordinates transformation

Along with the segmentation of the image the depth map is also computed, using which the new (x,y,z) coordinates are extracted for every point. These new coordinates are calculated based on the camera model described in [21].

Figure 3 shows a side-view of the simple camera model. C point corresponds to the center of the camera; p is the plane of the image; f is the focal distance, Z is z-coordinate value of the point, which corresponds to the value of the depth map; z axis, containing the point C is optical axis of the camera.

Coordinates transformation is considered as follows. Let (x,) be the coordinates of a point A with depth Z belonging to the image. The origin of coordinates is normally located at the (0,0) point of the image. If one turns from the image coordinate system to the standard camera coordinate system, then the optical axis crosses the image in the center at the point $\left(\frac{H}{2},\frac{W}{2}\right)$, where H and W are the height and the width of the image respectively. Then the coordinates of the point of the image in the real world can be calculated according formulae 1–3 due to the similarity.

$$x' = (x - W) \cdot \frac{z}{f}, \tag{1}$$

$$y' = (y - H) \cdot \frac{z}{f}, \tag{2}$$

$$z' = Z, \tag{3}$$

For depth map extraction, we use the pretrained *monoDepth* [18] neural network. After applying this model using the resulting depth map we obtain the coordinates (x',y',z') for each pixel of the image in the real world. Since the exact focal distance for random source image is unknown and can vary, it is taken as average, f = 520.

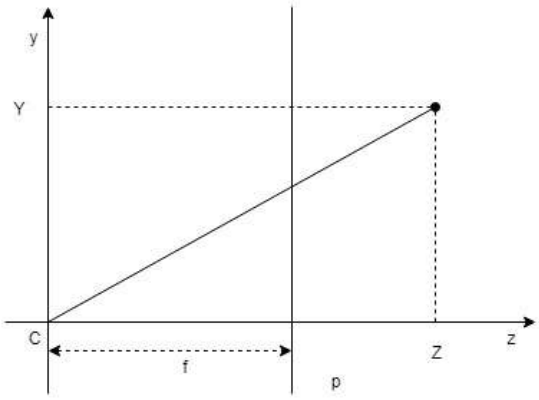

Figure 3. Simple camera model

3.3 Areas filtering

As mentioned above, the segmentation step is performed for the selection of suitable areas. However, not all the areas left after the segmentation are appropriate for text projection because of their size and location. This step of filtering areas is taken from [6]. First, we eliminate areas if the number of belonging pixels is lower than some threshold. Then, for each of the remaining areas, we compute the largest quadrilateral polygon containing all the pixels in the area. If at least one of its four sides is too small, this polygon is also eliminated.

3.4 Text projection

The method of projecting text is based on the projective geometry. Initially, the text is projected on the mask, the mask is oriented in the frontal-parallel. Fig.4 shows an example of a mask with text.

Figure 4. The example of a mask with text

Next, each mask is projected onto a black image that is equal in size to the input image. This produces a text mask for the entire image.

To project the mask onto the particular plane to which the considered segment belongs, it is necessary to build the homography matrix. To build it, we need to know the correspondence of at least four points of the mask in the frontal-parallel plane and the points in the plane of the segment. Without losing generality, within this work, the number of corner points selected is four. Therefore, one must find the four corresponding points in the plane of the segment. To do this, first it is necessary to find this plane. The algorithm for its search is the same as in article [6], described in more detail in Section 2. Briefly, it can be formulated as follows:

a. The depth map is calculated for the original image using *monoDepth* neural network [18];

b. The assumed coordinates of the points are restored in three-dimensional space using the depth map and standard camera model (Fig. 3), see details in Section 3.2;

c. For every segment using its points in three-dimensional space the normal vector $\vec{n}$ to its plane is calculated.

Pay attention that for the images from the Cityscapes dataset, the usage of a standard camera model is acceptable since all the images were obtained from the car DVR, thus the optical axis is located mostly in the center of image.

After finding the plane for a particular segment, we need to find four points on the plane to make the projection. For this purpose, let us consider a quadrilateral of the smallest square containing all points of the particular segment plane. The corner points of the quadrilateral do not fit, because, first, can be arbitrarily far from the most of the points of the segment, if it has a complex shape, and second, there is no guarantee that these corner points lie on the same plane with the segment. To address these two drawbacks, first a set of segment points that are closest to each of the quadrilateral's corner points is selected. From each of these four sets one point that belongs to the given plane is picked. Thus, the initial quadrilateral “shrinks” to the current segment. Figure 5 shows the example of depth map estimation and text projection.

Figure 5. The example of monoDepth neural network output and text projection using given output

After that, we need to set the correspondence between the corner points of the mask and the two-dimensional coordinates of the found polygon and build a homography matrix. Now for each of the mask points its projection on the original image is known.

Pay attention that it was still not taken into account that the segment does not have to be continuous: inside it there may also be other segments that can belong to another plane, so without any modifications the text will be projected on them too. To avoid this, after projection, for each text pixel on the mask it checks its belonging to the considered segment. If the segment is changed, the entire word containing this pixel is removed from the mask. This condition works due to the fact that the mask initially contains several areas with text, the square of bounding boxes of which is much smaller than the square of the mask itself.

4. EXPERIMENTS AND RESULTS

4.1 Experiments design

To evaluate the results, several models for detection and recognition were trained on generated by our algorithm images. First is Fully-Convolutional Regression Network (*FCRN*), proposed and used for evaluation in [6], as detector, in pair with *CRNN* [22] for recognition. The next model is the neural network described in [23], we call it *Jaderberg*, that performs both text detection and recognition end-to-end, i.e. the image in the wild is passed as the input for the inference without known locations of words. These networks were trained on the datasets obtained with the base of the Cityscapes dataset and two image generation algorithms: the baseline [6] and the method described in this paper. The number of images in each of the generated datasets is 7000, while for the evaluation of the baseline models 800,000 images were used. For our method 70,000 masks with text were built to be sure of covering each suitable area on every image. The base text was taken from the Newsgroup20 dataset [14].

To increase the number of images, the text was generated on one building or wall per one image. Images with more than one suitable building for text projection were cloned in the dataset by the number of containing buildings.

The results were tested and compared on Street-View Text (SVT) [24] and SVT-50 datasets. SVT contains 647 images obtained from Google Street View dataset. Notice that SVT is a rather noisy dataset containing a lot of unlabeled words. The SVT-50 dataset is based on the SVT augmented with a dictionary of 50 words.

For evaluating detection results, we use F_1-measure according ICDAR evaluation protocol [25]. For evaluating semantic segmentation results (the first step of our generation pipeline) we use intersection over unit (IoU) measure. We use 5-fold cross-validation. The experiments were conducted on a computer with a GeForce GTX 1080 Ti GPU with 128 GB of RAM.

4.2 Results comparison

For the semantic segmentation step we used U-Net [20] trained on modified Cityscapes dataset [19]. The IoU measure on the test set is 0.95.

Table 2 shows the comparison of FCRN + CRNN model trained on two synthetic datasets (SynthText and our) and tested on SVT and SVT-50 datasets by F_1-measure. The F_1-measure in the table is average value by 5 folds of experiments. In addition, Table 3 presents the comparison with the results of the *Jaderberg* model [23], trained also on ICDAR datasets: ICDAR 2003, ICDAR 2011, ICDAR 2013.

Table 2. Recognition results comparison by F_1-measure of models trained on synthetic and not synthetic datasets with FCRN + CRNN model.

Model (dataset)	**SVT**	**SVT-50**
FCRN + CRNN (SynthText)	52.7	75.7
FCRN + CRNN (our)	**53.6**	**76.8**

Table 3. Recognition results comparison by F_1-measure of models trained on synthetic and not synthetic datasets with Jaderberg model.

Model (dataset)	SVT	SVT-50
Jaderberg (SynthText)	53.1	76.1
Jaderberg (ICDAR)	53.0	76.0
Jaderberg (our)	**54.2**	**77.0**

The Tables shows that the models trained on our generated dataset allow higher score on both real datasets then the same models trained on SynthText and ICDAR datasets. In all cases, comparison of our dataset provides a slightly higher F_1-measure with the standard deviation equal to 0.05. This proves that our improvement of baseline approach [6] is statistically significant and decreases the false positives. The examples of the generated by our method images could be found at the website [26].

5. CONCLUSION

In this paper, we propose an approach for the synthetic image with text datasets generation, which is the improvement of algorithm [6]. The image generation is useful in different tasks related to text, for instance, text detection and recognition in the wild scenes for any languages for which no labeled dataset is available. The developed approach allows generating images with text, which look natural. With our approach, we achieved decreasing of false positives for text recognition models trained on the obtained synthetic dataset. These models showed better results than the same models trained on the dataset obtained by the baseline algorithm [6].

In the future work, we plan to improve this approach, by restoring exact camera parameters instead of using the average parameters. We also plan to try conditional generative adversarial networks [27] for solving the task.

ACKNOWLEDGMENTS

The authors would like to thank Arseny Nerinovsky for useful conversations.

This work was financially supported by the Government of the Russian Federation (Grant 08-08).

REFERENCES

[1] S. Kopf, T. Haenselmann and W. Effelsberg, "Enhancing curvature scale space features for robust shape classification," *In 2005 IEEE International Conference on Multimedia and Expo,* p. 4, 2005.

[2] J. Wang and J. Jean, "Segmentation of merged characters by neural networks and shortest-path," *Pattern Recognition,* pp. 649-658, 1994.

[3] C. C. Oliveira and P. P. de Oliveira, "An approach to searching for two-dimensional cellular automata for recognition of handwritten digits," *In Mexican International Conference on Artificial Intelligence* , pp. 462-471, 2008.

[4] K. Kise, A. Sato and M. Iwata, "Segmentation of page images using the area voronoi diagram," *Computer Vision and Image Understanding,* vol. 3, no. 70, p. 370–382, 1998.

[5] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao and J. Yan, "Fots: Fast oriented text spotting with a unified network," *In Proceedings of the IEEE conference on computer vision and pattern recognition,* pp. 5676-5685, 2018.

[6] A. Gupta, A. Vedaldi and A. Zisserman, "Synthetic data for text localisation in natural images," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 2315-2324, 2016.

[7] D. Debidatta, I. Misra and M. Hebert, ""Cut, paste and learn: Surprisingly easy synthesis for instance detection."," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1301-1310, 2017.

[8] G. Georgakis, A. Mousavian, A. C. Berg and Kosec, "Synthesizing training data for object detection in indoor scenes.," 2017.

[9] M. Goyal, P. Rajpura, H. Bojinov and R. Hegde, "Dataset augmentation with synthetic images improves semantic segmentation," *In Proceedings ofNational Conference on Computer Vision, Pattern Recognition, Image Processing, and Graphics*, pp. 348-359.

[10] M. Jaderberg et al., Synthetic data and artificial neural networks for natural scene text recognition, 2014.

[11] K. Wang, Word spotting in the wild, 2013.

[12] S. Capobianco and S. Marinai, ""DocEmul: a Toolkit to Generate Structured Historical Documents."," *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1, 2017.

[13] X. Wang, Z. Man, M. You and C. Shen, "Adversarial generation of training examples: Applications to moving vehicle license plate recognition," 2017.

[14] K. Lang and T. Mitchell, "Newsgroup 20 dataset 1999," 1999.

[15] P. Arbelaez , M. Maire, C. Fowlkes and J. Malik, "Contour detection and hierarchical image segmentation," no. 33, p. 898–916, 2011.

[16] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM.*, vol. 24, no. 6, pp. 381-395, 1981.

[17] P. Perez, M. Gangnet and A. Blake, "Poisson image editing," *ACM Transactions on Graphics*, vol. 3, no. 22, p. 313–318, 2003.

[18] C. Godard, O. Mac Aodha and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 270-279, 2017.

[19] S. B. Cordts, M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213-3223, 2016.

[20] O. Ronneberger, P. Fischer and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Proc. Med. Image Comput. Comput.-Assisted Intervention*, p. 234–241, 2015.

[21] R. Hartley and A. Zisserman, Multiple view of geometry in computer vision, 2004, pp. 153-166.

[22] B. Shi, X. Bai and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 11, pp. 2298-2304, 2016.

[23] M. Jaderberg, K. Simonyan, A. Vedaldi and A. Zisserman, "Reading text in the wild with convolutional neural networks," *International Journal of Computer Vision*, vol. 1, no. 116, pp. 1-20, 2016.

[24] K. Wang, "The Street View Text Dataset (SVT)".

[25] S. M. Lucas, A. Panaretos, L. Sosa and A. W. Tang, "ICDAR 2003 robust reading competitions," *In Seventh International Conference on Document Analysis and Recognition, Proceedings*, pp. 682-687, 2003.

[26] [Online]. Available: http://genome.ifmo.ru/files/papers_files/ICMV/.

[27] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014.