A Human Eye-based Text Color Scheme Generation Method for Image Synthesis

Shaowei Wang
College of Computer Science and Technology, Huaqiao University, Xiamen, China
jem2rsc@gmail.com

Xiangyu Luo∗
College of Computer Science and Technology, Huaqiao University, Xiamen, China
luoxy@hqu.edu.cn

Guanjie Huang
College of Computer Science and Technology, Huaqiao University, Xiamen, China
mdd@stu.hqu.edu.cn

ABSTRACT
Synthetic data used for scene text detection and recognition tasks have proven effective. However, there are still two problems: (1) The limitation and confusion of the existed color pairs learning from the real dataset; (2) The restriction of the text position which can only have the same color depth. In this paper, we design a novel method to generate color schemes, which is consistent with the characteristics of the human eyes to observe things. The advantages of our method are as follows: (1) It overcomes the problem of color confusion between the text and background caused by dirty data; (2) The generated texts are allowed to appear in most locations of any image, even across depths; (3) The speed of generating images is fast, nearly one image generated per three milliseconds; (4) It exceeds the state-of-the-art methods on several public datasets.

CCS CONCEPTS
• Computing methodologies → Artificial intelligence.

KEYWORDS
Image synthesis, Text color scheme, Data augmentation, Scene text recognition

ACM Reference Format:
Shaowei Wang, Xiangyu Luo, and Guanjie Huang. 2022. A Human Eye-based Text Color Scheme Generation Method for Image Synthesis. In 2022 6th International Conference on Electronic Information Technology and Computer Engineering (EITCCE 2022), October 21–23, 2022, Xiamen, China. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3573428.3573704

1 INTRODUCTION
The emergence of synthetic data effectively solves the bottleneck problem caused by model data-thirsty. Extensive researchers employ synthetic data to train models, which needs less effort and time, especially for the free label generation. The various generating methods [1-4] significantly improved the progress of data synthesis. [4] and [5] illustrate the process of solving difficult areas of text recognition. [6-10] were dedicated to solving text detection.

However, two issues are generally neglected in the existing methods: First, the color scheme used for coloring text learns from the real dataset which contains some inevitable dirty data. The dirty data means that the synthesized text and the background are tough to separate. For example, [7] utilizes the color pairs generated from the IIIT5K [11] dataset. As shown in (a) to (d) of Figure 1, four images are generated by [7]. The texts are framed in the Figure, and the color of the generated text is similar to the background. This instance also proves the unreasonableness of the existing color schemes. Second, existing methods ignored the situation that the text can appear across depths in the image. Two instances are displayed in Figure 1 ((e) to (h)), where (e) and (g) are real scene pictures while (f) and (h) are the depth maps corresponding to these two pictures.

This article has designed an original color scheme generation method. The proposed approach is inspired by the features of the human eyes to observe things. Specifically, our visual system is insensitive to slight changes in the grayscale. When the difference between the gray value of the text and the background is weak, it will produce a problem that the human eyes cannot recognize. Our approach takes advantage of this characteristic to render words. Even when the text spans depths, the algorithm can analyze the suitable color for the current background. Some related examples have been shown in Figure 1, ((i) to (l)). To have an impartial comparison, we take the identical picture mentioned in the above examples as the background. The position of the text is randomly selected. Since we did not make any collision detection, there are some overlaps between texts. However, the details in Figure 1 like (i) show that the colors among the texts cannot interfere with each other.

2 RELATED WORK
Humans utilize the visual system to record and transmit information. The visual system of our person is hard to perceive the weak changes in the grayscale. In a grayscale image, when the gray value changes by one or two pixels, the human eyes cannot distinguish it. Based on [12], the value scope of pixels in the grayscale picture is from 0 to 255. The recognition accuracy of human vision is about 93.16%, 68.75%, and 45.31% when the gray level is 8, 16 and 32 separately. The gray level denotes the difference between bright pixels and dark pixels. When the gray level is higher, the degree of subdivision of gray pixels is greater, and on the contrary, the accuracy of human eye recognition is lower. In addition, it also mentioned the view that there is no significant difference in the resolution of human eyes for grayscale images and RGB images.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
EITCCE 2022, October 21–23, 2022, Xiamen, China © 2022 Association for Computing Machinery. ACM ISBN 978-1-4503-9714-8/22/10 ... $15.00 https://doi.org/10.1145/3573428.3573704
3 METHODOLOGY

3.1 Overall Pipeline

The architecture of our method is displayed in Figure 2. In the preparation phase, we sampled some text fonts and background images. The font is extracted from Google fonts, which contains 3099 available Latin fonts. To verify the effectiveness of our method, 100 images collected from the dataset [7] are randomly selected as our background images. In Figure 2, the first column is a randomly selected image, and then we randomly select a position in the image to synthesize the text, the size of which is identical to the text. There are two parts in the second column: The first part is to crop the selected position and convert it to the corresponding grayscale image by averaging it (if it is originally a grayscale image, this step can be omitted); The objective of the second part is to get the text border. First, we sample a text arbitrarily and add some operations like rotation transformation. Then, we expand the text by 2 pixels. Finally, we subtract the original text from the expanded text to get the text border. The third column presents the essential component of our algorithm. Both the text border collected in the former stream and the translated gray background information are used to generate the appropriate background color. Further, we analyze all the gray values of the background contained in the text border through our algorithm to obtain the most suitable text color. The details of this part will be depicted in the subsequent paragraphs. The final column denotes the text rendered by our algorithm and the synthesized picture. It can be seen from the color analysis chart (the third column) that the background pixels are very complex, but our algorithm avoids generating dirty data, and the generated results are visible.

3.2 Thresholds

This module is the main idea of our method and contains two parts. The first one (vertical threshold) is to determine some valid pixels that could be used for text coloring. Both the text border collected in the former stream and the translated gray background information are used to generate the appropriate background color. Further, we analyze all the gray values of the background contained in the text border through our algorithm to obtain the most suitable text color. The details of this part will be depicted in the subsequent paragraphs. The final column denotes the text rendered by our algorithm and the synthesized picture. It can be seen from the color analysis chart (the third column) that the background pixels are very complex, but our algorithm avoids generating dirty data, and the generated results are visible.
threshold) aims to select some available colors as candidate text colors from the first part.

3.2.1 Valid & Invalid Pixels (Vertical Threshold). As shown in Figure 1, ((a) to (d)), the pixels used for text coloring have a negative impact on the results of the generated image. We call these types of pixels invalid pixels. This section aims to obtain valid pixels from any input image by the first threshold we set in the method. To get the best threshold, we conduct a comparison test. More specifically, we used 50,000 data and trained 20 cycles, and then used it to verify the ICDAR 2003 test set [13]. As is shown in Figure 3, we arrange the abscissa according to the percentage of the usage frequency of pixels. The ordinate is the experimental evaluation which involves two parts: "Accuracy" and "Loss". The result shows that the two curves of "Accuracy" and "Loss" both have a smooth trend, and finally achieve the best results at "0". Therefore, our first threshold is set to "0", and the meaning is to list all unused pixels.

3.2.2 Suitable Candidate Color (Horizontal Threshold). This section is based on the first threshold and aims to select the available values from the obtained pixels as a candidate for the final text coloring. The setting of the second threshold references the features of the human eyes to observe things. For the purpose of showing the characteristics more clearly, we have prepared an example, as shown in Figure 4. It clearly describes the influence of gray resolution. Here we use black as the background and choose "Hello" as the text content, which will serve as a carrier to show us the differences among different text colors. The range of gray pixels we used is from 0 to 255, and it is separated into 32 parts, which means that each text color in the box increased by the 8-pixel interval. As mentioned above, the recognition accuracy of the human eyes corresponding to the 32-level gray value is about 45.31%. From the Figure, we can find that the text under similar pixels is difficult to distinguish. As shown in Figure 5, we conducted experiments on ICDAR 2003 to determine the final set of the second threshold.

4 EVALUATION

4.1 Implementation Details

We use CRNN [14] as our recognition model, which uses CTC [15] as a loss function and completes the line-level text recognition.
as input. More importantly, it can handle sequence recognition problems of any length, with no need for predefined lexicons. In the evaluation part, we have made two tests. The primary one is text recognition. We compare our algorithm with some state-of-the-art synthesis approaches and some recognition algorithms. To create a fair comparison environment, we all use the same amount of training data. In the second experiment, we compare the existing color scheme (IIIT5K) with ours and make the ablation studies on analyzing the best setting of the two key thresholds. The above two analyses are implemented on an Ubuntu workstation with an 8-core Intel CPU, an NVIDIA TITAN RTX GPU, and 16G RAM.

4.2 Experiments on Text Recognition

The proposed approach is evaluated over three public datasets including ICDAR 2003, ICDAR 2013, and Street View Text. Table 1 shows the comparison among some recognition algorithms, CRNN training with the three existing synthesis methods separately, and CRNN with ours. For a fair comparison with other synthesis methods, we choose Google Fonts as these methods [5][7][10] do, and the same text corpus used in [5]. We all adopt the CRNN method and use 5 million (5M) data to train the model. Compared with some recognition algorithms, ours has achieved the most advanced scene recognition accuracy on the ICDAR 2003 and ICDAR 2013 datasets. The poor performance of other synthesis methods is largely due to the unreasonable text color scheme. However, on arbitrary shape recognition data such as SVT, CRNN training with our data still performs less well than "CRNN+VISD". The reason is that we only make some simple operations to generate curve text, which is not enough for model learning.

4.3 Ablation Studies

In this section, we first evaluate the impact of the color scheme by comparing our generated data set with a public data set, and then we conduct ablation studies to validate the two key thresholds of our algorithm.

We take IIIT5K as the public dataset and compare it with our approach under different thresholds. The experimental setting includes 1 million training data, 60 training epochs, CRNN as the recognizer, and ICDAR 2003 as the verification dataset. Moreover, we use 0.9 as the curve smoothing coefficient. Here we use less training data (1 million), which causes the performance in this Figure is worse than shown in Table 1.

As shown in Figure 5, the pink dotted line indicates the performance of IIIT5K, and other lines indicate our method. Whether it is an accuracy or loss graph, the performance of IIIT5K is worse than ours. The reason is that the dirty data in the dataset can lead to the problem of color confusion. The results in Figure 5 also prove that the combing of "0vt" with "16px" performs the best.

4.4 Speed Evaluation

In this subsection, we will validate the superiority of our method’s speed. In Figure 6, the abscissa denotes time, and the ordinate denotes the number of generating pictures. We use different colors of lines to indicate the speed of each algorithm to synthesize pictures. The two dashed lines with pink and red colors represent the speed of the algorithm [9] and the algorithm [7], and the other two lines represent our method.

As shown in the Figure, [9] produces about 50 pictures per minute, and [7] produces nearly 120 pictures per minute. After a rough calculation, the speed of our algorithm to generate a picture containing only one character is about 3 milliseconds. Here we assume a picture containing 10 words, which means that the generated speed is nearly 30 milliseconds to produce an image (which represents the "speed_ours (30ms)" in the figure). Similarly, the speed of generating an image with 40 words equals 120 milliseconds. It is obvious that the steepness of the slope of our algorithm is larger and this also verifies the advantage of the speed of our method.

5 CONCLUSIONS

This paper introduces a new approach to data synthesis. We mainly start with the low-level feature, the text color, and take into account the limited grayscale that the human eye can distinguish. With the human visual feature, we designed a generation method of color schemes. Our color schemes prevent text from being confused by complex backgrounds or cross-depth areas, and the throughput of our method is also fast. More importantly, our method allows texts to appear in the image across depths. Consequently, our method is helpful to improve the performance of the text recognition models. Nevertheless, there is still much work worthy of in-depth study. For example, in this paper, we only make the recognition experiments, and the performance of the detection tasks can also be improved through our method in theory. In the next step, we will conduct experiments for detection tasks.
Figure 5: The comparison of color schemes between the IIIT5K dataset and ours on accuracy and loss. We uniformly utilize 1 million training data and 60 epochs. “0vt” denotes the first threshold is set to 0. “0,8,16,24px” denotes the second.

Figure 6: The speed comparison between different methods, “speed 16” refers to [7], and “speed 20” refers to [9]. “speed ours(30ms)” and “speed ours(120ms)” represent different costs of 30 and 40 words using our method.

REFERENCES

[1] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. 2015. Flownet: Learning optical flow with convolutional networks. In Proceedings of the IEEE international conference on computer vision. 2758–2766. https://doi.org/10.1109/ICCV.2015.316

[2] Illker Yildirim, Tejas D Kulkarni, Winrich A Freiwald, and Joshua B Tenenbaum. 2015. Efficient and robust analysis-by-synthesis in vision: A computational framework, behavioral tests, and modeling neuronal representations. In Annual conference of the cognitive science society, Vol. 1.

[3] Stefan Hinterstoisser, Olivier Pauly, Hauke Herbel, Martina Marek, and Martin Bokeloh. 2019. An annotation saved is an annotation earned: Using fully synthetic training for object instance detection. arXiv preprint arXiv:1902.09967 (2019).

[4] Tao Wang, David J Wu, Adam Coates, and Andrew Y Ng. 2012. End-to-end text recognition with convolutional neural networks. In Proceedings of the 21st international conference on pattern recognition (ICPR2012). IEEE, 3304–3308.

[5] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Synthetic data and artificial neural networks for natural scene text recognition. arXiv preprint arXiv:1406.2227 (2014). https://doi.org/10.48550/arXiv.1406.2227

[6] Fangneng Zhan, Hongyuan Zhu, and Shijian Lu. 2019. Scene Text Synthesis for Efficient and Effective Deep Network Training. arXiv preprint arXiv:1901.09193 (2019).

[7] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. 2016. Synthetic data for text localisation in natural images. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2315–2324.

[8] Minghui Liao, Boyu Song, Shangbang Long, Minghang He, Cong Yao, and Xiang Bai. 2020. SynthText3D: synthesizing scene text images from 3D virtual worlds. Science China Information Sciences 63, 2 (2020), 120105. https://doi.org/10.1007/S11432-019-2737-0

[9] Shangbang Long and Cong Yao. 2020. UnrealText: Synthesizing realistic scene text images from the unreal world. In Proceedings of the IEEE/CVF Conference
