Docscanner: document location and enhancement based on image segmentation

Ziqi Shan, Yuying Wang*, Shunzhong Wei, Xiangmin Li, Haowen Pang and Xinmei Zhou
School of computer and information security,
Guilin University of Electronic Technology, Guilin 541004, China;
**Corresponding author's e-mail: getwyy@guet.edu.cn*

***Abstract*—Document scanning aims to transfer the captured photographs documents into scanned document files. However, current methods based on traditional or key point detection have the problem of low detection accuracy. In this paper, we were the first to propose a document processing system based on semantic segmentation. Our system uses OCRNet to segment documents. Then, perspective transformation and other post-processing algorithms are used to obtain well-scanned documents based on the segmentation result. Meanwhile, we optimized OCRNet's loss function and reached 97.25 MIoU on the test dataset.**

***Keywords*-Document Scanner, Document Processing, Semantic Segmentation.**

I. Introduction

Document scanning systems have been widely used in many fields such as official business, administration, etc. With the demand for on-device document scanning increasing, document scanning systems are proposed to provide the office crowd with convenience.

The current document scanning system is mainly based on traditional algorithms or keypoints detection. [1] proposed a advanced hough-based method for document localization. Javed et al. [2] proposed a document keypoints detector based on cnn. However, for most of those works, the quality of scanned images still can be improved to make a more accurate and robust document capturer.

We propose a document capturer based on semantic segmentation in this work and use advanced post-processing algorithms to gain well-scanned documents. The system uses the OCRNet [3] to segment documents. OCRNet is based on the object-contextual representation scheme and can achieve a good document segmentation effect. According to the segmentation result, we can fit the four corners of the document. Then we use perspective transformation [4] and other post-processing methods to make the well-scanned document.

II. System Design

Figure 1 shows the system procedure. First, we use OCRNet [3] to get the document mask from the input image. Second, we extract the four key points of the documents. Then we apply a perspective transformation to convert the document image from a 3D image to a 2D scanned image. Finally, we use binarization and other post-processing algorithms to enhance the document image.

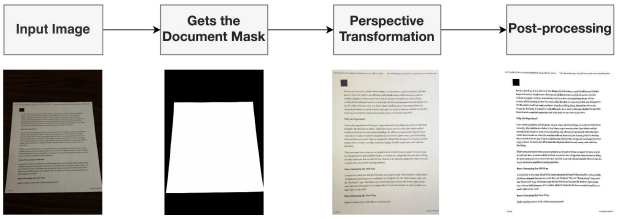

Figure 1. System Procedure

III. Implementation

A. Image Segmentation

Semantic segmentation has been widely used in many fields. We use OCRNet [3], a semantic segmentation network, to segment documents in this system. OCRNet solves the context aggregation problem in semantic segmentation by using target regional representations to enhance its pixel representations, which improves the quality of segmentation results. We also use the Ohem cross-entropy [5] loss to solve the class imbalance problem.

1) Network Architecture: The architecture of OCRNet is shown in Figure 2. OCRNet in our system choose HRNet-W48 [6] as the backbone. In the following stage, OCRNet first divides the contextual pixels into a set of soft object regions based on the output of the backbone. Each region corresponds to a class. Based on soft object regions and pixel representations of the backbone, OCRNet calculates a group vector called Object Region Representations. Each of these vectors corresponds to a semantic category's feature. Next, OCRNet will calculate the relations between pixel object regions. The OCR is the weighted aggregation of all the object region representations with the weights calculated according to the relations between pixels and object regions [3]. Finally, OCR and deep network features are combined as augmented representations.

2) Loss Function: The loss function of OCRNet in our system contains two parts: the Ohem cross-entropy loss [5] and the Dice loss [7]. Ohem cross-entropy loss can help to solve the class imbalance problem. Dice loss is used to measure whether the predicted mask is close to the groundtruth. Our loss function is defined as follows:

$$L = L_{Ohem} + wL_{Dice} \tag{1}$$

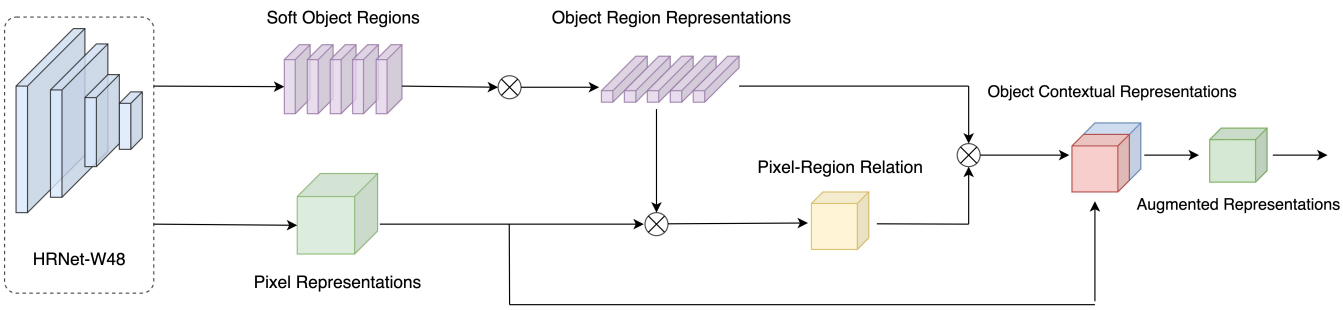

Figure 2. The Architecture of OCRNet

w is weight coefficient. We choose $w = 0.2$ in this work. L_{Ohem} is the Ohem cross-entropy loss and L_{Dice} is the Dice loss. Ohem cross-entropy loss, known as online hard example mining cross-entropy loss, is used to pay close attention to hard examples and set a higher weight. Let $\mathcal{M}$ be the predicted mask. $\mathcal{G}$ is the groundtruth mask of the image. L_{Ohem} is defined as follows:

$$L_{Ohem} = \sum_{i=1}^{N} \max_{i=1}^{N} \mathcal{M}_i log(\mathcal{G}_i) \tag{2}$$

L_{Ohem} first calculates each pixel's loss, then sums the top k loss items. It selects hard examples as training samples to improve the effect of network parameters. L_{Dice} is derived from the dice coefficient and is often used as the metric function to estimate the comparability of two samples. L_{Dice} is defined as follows:

$$L_{Dice} = 1 - \frac{2|\mathcal{M} \cap \mathcal{G}|}{|\mathcal{M}| + |\mathcal{G}|} \tag{3}$$

B. Perspective Transformation

After we obtain the predicted mask, we can carry the perspective transformation to the image according to the mask. The process is shown in the Figure 3. First, we use the canny [8] algorithm to extract the edge of the mask. Then we use the Douglas-Peucker algorithm [9] to find the four corners of the edge. According to the key points, we can calculate the perspective transformation matrix. Then we apply the perspective transformation to the image according to the matrix to convert it from a 3D image to a 2D image.

Figure 3. The process of perspective transformation

C. Post Processing

At the last step, we obtain the 2D view document image. In order to get the better quality of the document image, we use post-processing algorithms to enhance the document image. The post-processing algorithms mainly include separating the background and foreground colours.

Table I
EXPERIMENT ENVIRONMENT

Equipment	Configuration Information
CPU	Intel(R) Xeon(R) Gold CPU @ 3.00GHz
RAM	64G
GPU	Tesla V100-PCIE-32GB
OS	Ubuntu 16.04.3 LTS
Programming Language	Python
DL Framework	PaddlePaddle

1) Separate Background Colors: The usual method to separate background colours is by analyzing the number of pixel values. However, even the largest number pixel value of an 8-bit colour image is a tiny percentage (under 10%) of all pixels. Therefore, we convert an 8-bit channel colour image to a 6-bit channel colour image. Reducing the pixel type will help determine the background colour. By analyzing the 6-bit colour image's pixel value number, we can determine the background colour. Document images are enhanced by replacing the background colour with white.

2) Separate Foreground Colors: Once the background colour has been identified, we can calculate the foreground colour by calculating how similar each pixel is to the background colour. We first convert the image to HSV color space. Then we calculate the European distance between each pixel and the background colour. To make the image more colourful, we cluster the foreground pixels into eight groups using k-means [10]. We will assign every foreground colour to one of the eight groups.

IV. EXPERIMENTS

This section will introduce the experiment setup, image segmentation, and post-processing result. We will also compare different image segmentation methods and post-processing methods.

A. Experiment Setup

Our experiment environment is shown in Table 1. We use Baidu open source dataset containing 3,000 images. We choose the OCRNet [3] and U-net [11] for image segmentation to compare the result. For the post-processing algorithm, we compare the adaptive binarization method and

Figure 4. Comparition between OCRNet and UNet

our method described in Section III-C. For our experiments, we use Baidu's open-source document dataset.

B. Experiment Results

1) Image Segmentation: We compare three groups of image segmentation methods: OCRNet(without the Ohem loss), U-Net, and OCRNet(Ours). The experiment result is shown in Table II. OCRNet(Ours) achieve the best result (94.14% miou) on the test dataset. By comparing OCRNet with U-Net, we can find that U-Net's prediction masks have some holes, as shown in Figure IV-B. OCRNet, because of its context aggregation strategy, performs excellently in document segmentation.

Table II
THE COMPARISON OF DIFFERENT MODELS

Model	MIoU(%)	Mean Acc(%)
OCRNet(Origin)	94.14	96.15
U-Net	94.46	96.64
OCRNet(Ours)	97.26	99.36

2) Image Post Processing: We compare two image post processing method: adaptive binarization and our method. Adaptive binarization is widely used in image enhancement. Our method is based on the idea of separation of foreground and background and is described at Section III-C2. Figure IV-B2 show the comparition of the two methods. Adaptive binarization performs better when the image is not noisy. However, when the image is noisy, the adaptive binarization method is unsuitable, as shown in the third row of Figure IV-B2. In the meantime, adaptive binarization will result in grayscale images. Our method is better when the image is noisy. And we can make the scanned document colorful.

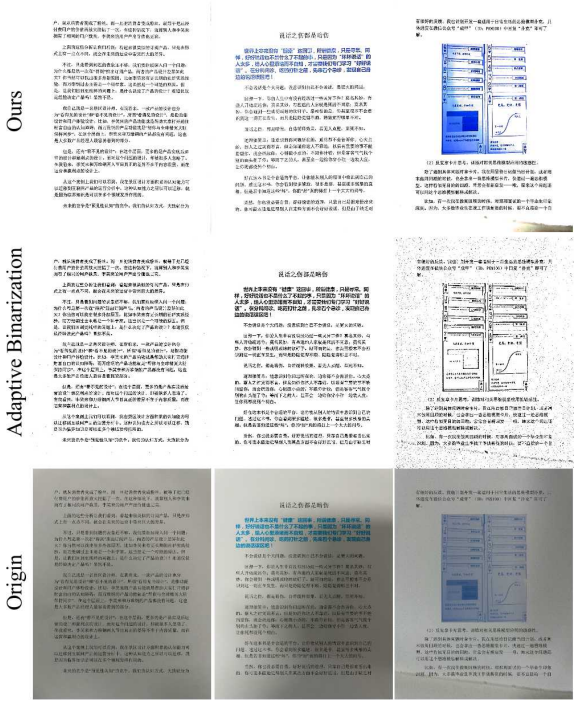

Figure 5. Post-processing Method Comparition

V. CONCLUSION

In this paper, we build a document scanning system to generate precise and colorful results. The system is based on image segmentation. We use OCRNet to produce the document segmentation mask. Meanwhile, we use a mixed loss function combining Ohem cross entropy and dice loss to improve the model. We also use an advanced post-processing method to enhance the scanned image.

ACKNOWLEDGMENT

This work is supported by Student's Platform for Innovation and Entrepreneurship Training Program under Grant (202110595025).

REFERENCES

[1] D. V. Tropin, A. M. Ershov, D. P. Nikolaev, and V. V. Arlazarov, "Advanced hough-based method for on-device document localization," *ArXiv*, vol. abs/2106.09987, 2021.

[2] K. Javed and F. Shafait, "Real-time document localization in natural images by recursive application of a cnn," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 105–110.

[3] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *European conference on computer vision*. Springer, 2020, pp. 173–190.

[4] H. Lin, P. Du, W. Zhao, L. Zhang, and H. Sun, "Image registration based on corner detection and affine transformation," in *2010 3rd International Congress on Image and Signal Processing*, vol. 5. IEEE, 2010, pp. 2184–2188.

[5] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 761–769.

[6] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3349–3364, 2020.

[7] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 fourth international conference on 3D vision (3DV)*. IEEE, 2016, pp. 565–571.

[8] P. Bao, L. Zhang, and X. Wu, "Canny edge detection enhancement by scale multiplication," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 9, pp. 1485–1490, 2005.

[9] S.-T. Wu, A. C. da Silva, and M. R. Márquez, "The douglas-peucker algorithm: sufficiency conditions for non-self-intersections," *Journal of the Brazilian Computer Society*, vol. 9, pp. 67–84, 2004.

[10] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the royal statistical society. series c (applied statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.