An automatic extraction system for screen-shot documents based on deep learning

Shouming Hou*[a], Kai Li[a], Yabing Wang[a]
[a] College of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, 454000, China
[*] Corresponding author: housm@163.com

ABSTRACT

The use of mobile phone cameras to capture and save reports shared on the screen at the meeting has become a major way for researchers to obtain information. However, a large number of screen-shot images obtained in this way have a lot of redundant information, and it takes a lot of time and energy to organize and save them later, therefore, it has become a realistic requirement to develop a software tool that can quickly extract the subject content of screen-shot images and realize automatic batch segmentation and compression storage. We use the self-made screen-shot image dataset SSD (Screen-Shot Document Dataset) composed of more than 1000 images for training, based on the improved U^2-Net network model to achieve automatic segmentation of screen-shot image subject area, combined with OTSU binarization, Canny edge detection and Hough Transform to extract the quadrilateral boundary of the subject area, and implement an Android-based screen-shot document automatic extraction system. The system can be used to automatically extract and save screen-shot document to PDF format in real time or at a later stage, significantly improving the efficiency of information collection and storage for researchers, reducing the financial and time loss caused by researchers not being able to find backups when they need information and discuss key content for meetings, and saving storage space on mobile phones.

Keywords: screen-shot image, deep learning, U^2-Net Network model, SSD dataset, image segmentation

1. INTRODUCTION

With the advent of the era of information intelligence, the application of large-screen display devices in life is becoming more and more extensive. The use of projected Led splicing display devices used in various conferences and academic exchanges makes information sharing and communication more vivid. For most researchers, one of the purposes of participating in offline conferences is to understand the research status and collect relevant data through the speaker's report, if the speaker is willing to share the content of his report, it will be a boon for the researchers participating in the conference. However, in most cases, with the acquiescence of the organizer and the speaker, the participants need to record the content of the report shared by the speaker by taking screen images (referred to as screen-shot images) through their mobile phones, to collect and organize the data for the research of interest. Recording and sharing content of interest through screen-shot images has become a spontaneous and convenient way for researchers and the public to exchange information and keep records between different media platforms[1].

According to incomplete statistics, at present, the display equipment objects captured by the screen-shot images mainly include projector screens, LED splicing large screens, large-screen LCD TVs, etc, and the displayed documents mainly include PPT (Power Point), PDF (Portable Document Format), Word and so on. Usually, the screen-shot images obtained by participants often include the background of the conference site in addition to the content of the theme report displayed in the center, the outline of the document will also be geometrically deformed, and sometimes it will be partially occluded. All of these bring great problems to the post-processing and editing of screen-shot images, and participants often need to spend more time to organize a speaker's report into a document that meets the requirements for reporting and sharing.

Currently, existing document segmentation and extraction tools often use human-computer interaction to automatically identify quadrilateral outlines in images through artificial intelligence image segmentation algorithms, and save them after confirmation by users. For example, apps such as Office Lens[2] and PDF-scanner[3] are mostly used as scanners on

Third International Symposium on Computer Engineering and Intelligent Communications (ISCEIC 2022),
edited by Xianye Ben, Proc. of SPIE Vol. 12462, 124620F · © The Authors. Published under a
Creative Commons Attribution CC-BY 3.0 License · doi: 10.1117/12.2660983

the mobile side. This method has no problem with dividing and saving a single or a small number of pictures, but for screen-shot images of conference reports, which are often in the hundreds of thousands, researchers get bored or overwhelmed. Therefore, it has become a realistic requirement to develop a tool software that can quickly extract the subject content of screen-shot images and realize automatic batch segmentation and compression storage.

In order to achieve the above goals, we use a method based on deep learning to segment and extract the subject content of the conference screen-shot images, combine the traditional Canny edge extraction algorithm to obtain the content boundary, extract the complete and standardized documents through perspective transformation, and an automatic extraction system for screen-shot images has been implemented based on Android mobile.

2. BACKGROUND

2.1 Image segmentation

Image segmentation can be formulated as the problem of classifying pixels with semantic labels (semantic segmentation), or partitioning of individual objects (instance segmentation), or both (panoptic segmentation)[4]. The research on image segmentation algorithm first started in 1979 when Nobuyuki Otsu used the method of selecting threshold from grayscale histogram[5], Then to the later k-means clustering[6], the watershed method[7], the active contours[8], the method based on sparsity[9]. With the development of deep learning (DL) in recent years, many new generation image segmentation models have been produced, which have significantly improved the performance of image segmentation.

For the extraction of the screen-shot document in this paper, the screen-shot image needs to be separated from the foreground and the background, that is, the foreground object and the background object in the image are segmented. The foreground and background being the parts of interest and disinterest to us, which in this paper are the screen-shot documents and non-document noise respectively. With the development of deep learning, image segmentation can achieve pixel-level object separation, the vision community has made great strides in instance segmentation, in part by leveraging strong similarities in the well-established field of object detection[10]. The representative segmentation methods based on the deep science department are: (1) VGGNet[11], ResNet[12], etc, based on feature coding; (2) R-CNN [13], Mask R-CNN[14], etc, based on region selection; (3) FCN[15], U-Net[16], U^2-Net[17] and other methods based on upsampling/deconvolution. Although, so far, there has not been a general and perfect image segmentation method, but the general laws of image segmentation have almost reached a consensus and have produced quite a few search results and methods.

Screen-shot images are rich in colors and features, and there are many factors that affect document extraction. The results obtained by using the U^2-Net model are prone to graying of the edges, which makes the position of the final vertex deviate from the real value. In this paper, the feature map output by the U^2-Net model further uses a single-channel U-Net to extract more accurate edge features to solve the grayscale situation. And use deep learning and traditional methods to eliminate factors such as angle and occlusion.

2.2 Perspective transformation

When shooting during the meeting, the participants are distributed in different directions, so the screen-shot document in the last saved image will have a certain angle, which is not convenient to view. As shown in Fig.1, PT(perspective transformation) can keep a certain area straight, that is, the straight lines in the original image are still straight after PT. It is a method of projecting an image from one geometric plane to another geometric plane, after three-dimensional transformation, and then mapping it to two-dimensional space.

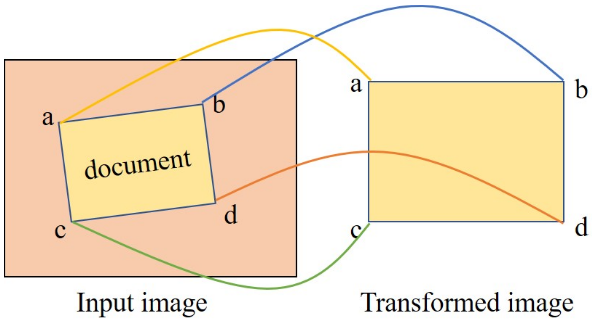

Figure 1. Perspective transformation

3. SYSTEM DESIGN AND IMPLEMENTATION

The screen capture document automatic extraction system mainly includes four parts: conference image acquisition, image segmentation, document extraction, and PDF generation. The overall structure of the system is shown in Fig.2.

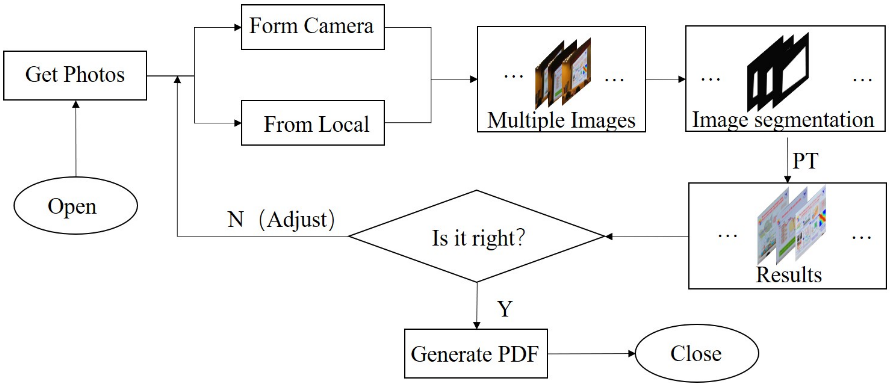

Figure 2. System overall structure diagram

The system uses on-site shooting and local uploading to obtain screen-shot images to ensure that documents can be obtained during and after the meeting. First, the acquired image is segmented by the deep learning model, and the result of image segmentation is represented as a binary image, that is, the foreground document area is set to white, and the background area is set to black, so as to facilitate the acquisition of the vertexes of PT. Further use traditional digital image processing to obtain document vertexes, and finally use the vertexes to perform PT to obtain the document. If the document content is correct and orderly, a PDF is generated and saved, otherwise the image is adjusted until it meets the requirements.

3.1 Make SSD dataset

In order to make the image segmentation model have high accuracy and generalization, this paper produces the SSD data set shown in Fig.3 for the screen-shot image, and the RGB image in the Fig.3 is the 'train_img' dataset, and the binary image is the 'train_mask' dataset.

This dataset captures multiple sets of screen-shot images from different angles in conference rooms, classrooms, and other places where screen-shot documents appear as much as possible. Meanwhile, in order to obtain the corresponding MASK set, the coordinates of the four vertexes of the screen-shot document are manually recorded, and the quadrilateral determined by the vertex is generated through batch processing. If the coordinates are within the quadrilateral, the value of the point is changed to white, otherwise modify it to black, so that the binarized MASK set can be obtained.

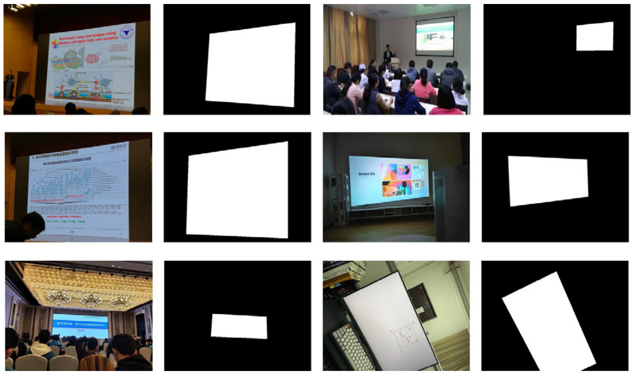

Figure 3. Examples of SSD dataset

3.2 Train an image segmentation model

or static background segmentation algorithms, traditional methods currently do not have a good segmentation threshold. However, compared with other methods, U^2-Net[16] based on deep learning has higher performance and smaller model size, which is suitable for running on Android and can effectively control APK (Android application package) size. But, this paper performs foreground and background segmentation for screen-shot documents. If the model trained by U^2-Net is used directly, the degree of generalization is low and the pertinence is weak. Therefore, this paper modifies the U^2-Net network model, the U-Net[15] model is added after the feature map to enhance feature reusability, and uses the SSD data set for training.

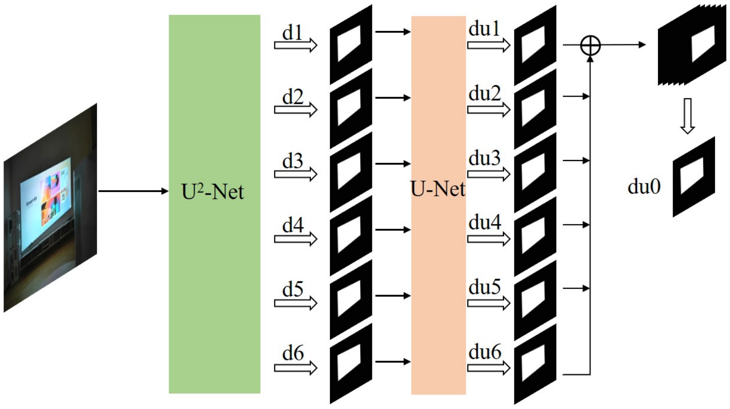

Figure 4. Modified model

As shown in Fig.4, ‘d1~d6’ are the six feature maps output by the U^2-Net model, and use U-Net for feature extraction for each feature map to obtain six feature maps: ‘du1~du6’. then, splicing these six feature maps to get the final result ‘du0’, through the test. Under the same training conditions, the modified model performs better than the U^2-Net model (see Fig. 5).

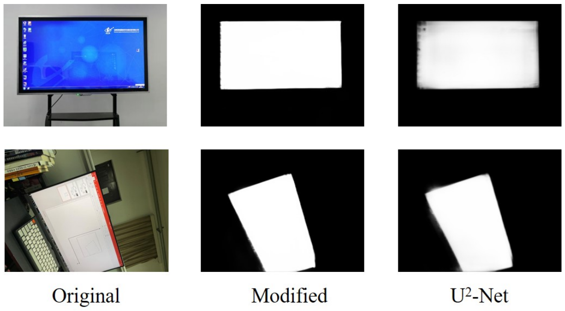

Figure 5. Comparison of model result

3.3 Get vertexes for perspective transform

After the image uploaded by the user is processed by the model, a binary image with front and background separation is obtained. If the screen-shot document is extracted from the binary image, a series of digital image processing methods are also required, as shown in Fig.6.

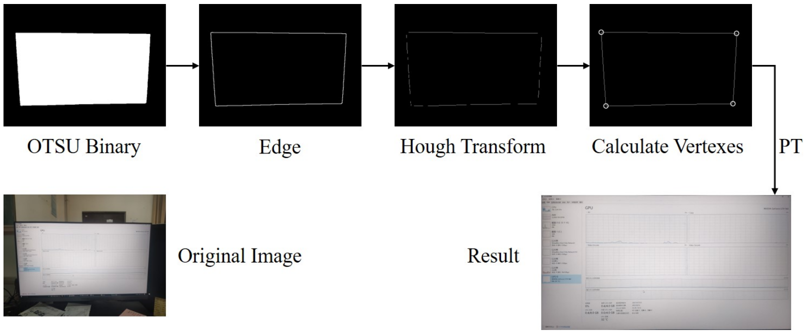

Figure 6. Document extraction process

Firstly, OTSU binarization is performed on the obtained feature map to eliminate the gray-scaled edge; Secondly, the edge image of the document is obtained by Canny edge detection algorithm; Then, the Hough Transform is used to find each line segment that matches the set length of the edges, and divide these line segments into four sides LEFT, TOP, RIGHT, BOTTOM according to the coordinates of the center point of the document, then integrate these edges into straight lines, and calculate the four vertexes of the document through the intersection of the straight lines; Finally, perform PT on the original image with the obtained vertexes to get the result.

3.4 Design the interface

3.4.1 System interface

As a simple and convenient meeting document extraction tool, our system can be used directly without registering users. As shown in Fig.5, the system has a total of three bottom navigation buttons, which are the ‘pictures’ page for selecting images, the ‘processed’ page for the task of performing the extraction and generating PDF documents, the ‘pdfs’ page for managing and viewing the extracted meeting documents.

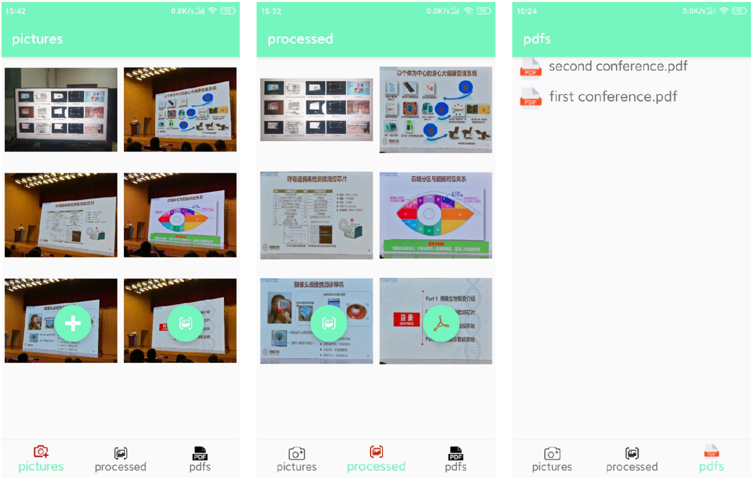

Figure 7. System interface

3.4.2 Select image module

On the ‘pictures’ page(see Fig.7(left)) of the system, users can click the ‘add button’ on the left to call out a floating window for selecting image sources. There are two ways to obtain images: camera shooting and local uploading. Selecting the camera to shoot requires the user’s authorization to open the camera. Selecting the camera to shoot requires the user's authorization to open the camera, returns captured images based on timestamps when the camera was turned on and off.

If the user chooses to upload locally, they need to authorize the album permission. Develop a new image selection component, realize clicking local upload to jump to multiple image selection pages, and return to the selected images according to the click order. Display multiple selected images on the 'pictures' page for preview, click the image to display the original image, and long press the image to add, replace, and delete images at the current location. Click the Process button on the right side of the page to send the currently selected image to the Process page(see Fig.7(middle)).

3.4.3 Document extraction module

In the document extraction module, click the 'process button' on the left to load multiple images sent by 'pictures' page via the pytorch API to load the trained model for image segmentation, and then utilize multi-threading mechanism for document extraction. After the extraction is successful, the result will be displayed in the 'processed' page, and it has the same function of previewing and modifying a single image as the image selection module. Click the 'generate pdf button' on the right to choose from a variety of ways to generate PDF documents: (1) Delete the selected image and the extraction result image, only generate a PDF document; (2) Save the selected image and the extraction result image and generate a PDF document. After the user confirms the saving method, use the iText pdf API to generate a PDF document, which can be viewed and deleted on the 'pdfs' page(see Fig.7 (right)).

4. TEST AND OPTIMIZE

After completing the development of the app that automatically extracts the screen-shot document, the overall performance of the app was tested with different models of Android phones and compared with other apps. The results are shown in Fig 8.

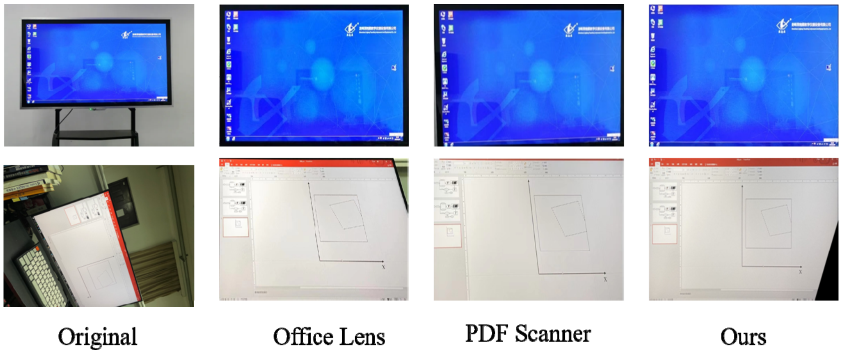

Figure 8. Example of comparative results

The results show that the system performs well in the extraction of screen-shot documents, It can accurately extract images with geometric deformation or lack of vertices, and the distortion correction effect is better. When the image resolution is large, the image preview interface of some models will appear slightly stuck. Our solution is to compress the image while ensuring the image clarity, and use the original image to save the image.

5. CONCLUSION AND OUTLOOK

We designed and implemented an automatic screen-shot document extraction system for meeting attendees' requirements for obtaining conference documents. The system uses a combination of deep learning and traditional methods to obtain more accurate documents, and adopts a multi-threading mechanism and compressed image preview to improve extraction efficiency and robustness. Through the test results of the actual scene and the analysis of the overall system performance, the system has good extraction effect, simple operation, and good real-time performance and environmental adaptability. Through the test results of the actual scene and the analysis of the overall system performance, Through the test results of the actual scene and the analysis of the overall system performance, our system has simple operation, good extraction effect, and good real-time performance and environmental adaptability. In the future, we will augment our dataset to make the system have better generalization ability.

ACKONWLEDGMENT

This work was supported by the Key Scientific Research of the Higher Education Institutions of Henan Province, China 22B520012, and in part by the Education Science "14th 5-year Plan" 2021 of Shanxi Province, China SGH21Y0398.

REFERENCES

[1] Liu, B., Shu, X., & Wu, X. (2018). Demoiréing of Camera-Captured Screen Images Using Deep Convolutional Neural Network. ArXiv, abs/1804.03809.

[2] https://office-lens.en.softonic.com/ office lens- microsoft

[3] https://app.mi.com/details?id=com.szyy2106.pdfscanner

[4] Minaee, S., Boykov, Y., Porikli, F.M., Plaza, A.J., Kehtarnavaz, N., & Terzopoulos, D. (2022). Image Segmentation Using Deep Learning: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44, 3523-3542.

[5] N. Otsu, "A threshold selection method from gray-level histograms," IEEE Trans. Syst. Man Cybern., vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[6] N. Dhanachandra, K. Manglem, and Y. J. Chanu, "Image segmentation using K-means clustering algorithm and subtractive clustering algorithm," Procedia Comput. Sci., vol. 54, pp. 764–771,2015.

[7] L. Najman and M. Schmitt, "Watershed of a continuous function," Signal Process., vol. 38, no. 1, pp. 99–112, 1994.

[8] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," Int. J. Comput. Vis., vol. 1, no. 4, pp. 321–331, 1988.

[9] J.-L. Starck, M. Elad, and D. L. Donoho, "Image decomposition via the combination of sparse representations and a variational approach," IEEE Trans. Image Process., vol. 14, no. 10, pp. 1570–1582, Oct. 2005.

[10] Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019). Yolact: Real-time instance segmentation. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 9157-9166).

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, arXiv:1409.1556.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

[13] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2015). Region-based convolutional networks for accurate object detection and segmentation. IEEE transactions on pattern analysis and machine intelligence, 38(1), 142-158.

[14] K. He, G. Gkioxari, P. Doll?ar, and R. Girshick, "Mask R-CNN," in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 2961–2969.

[15] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in Proc. IEEE Conf. Comput.Vis. Pattern Recognit., 2015, pp. 3431–3440.

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervention, 2015, pp. 234–241.

[17] Qin, X., Zhang, Z., Huang, C., Dehghan, M., Zaiane, O. R., & Jagersand, M. (2020). U^2-Net: Going deeper with nested U-structure for salient object detection. Pattern recognition, 106, 107404.