Optical Braille Recognition Based on Semantic Segmentation Network with Auxiliary Learning Strategy

Renqiang Li[1] Hong Liu[1*] Xiangdong Wang[1] Jianxing Xu[2] Yueliang Qian[1]
[1] Beijing Key Laboratory of Mobile Computing and Pervasive Device,
Institute of Computing Technology, Chinese Academy of Sciences
[2]Northwestern Polytechnical University
[1]{lirenqiang,hliu,xdwang,ylqian}@ict.ac.cn
[2]xjx_david@outlook.com

Abstract

Optical Braille Recognition methods usually use many designed steps, such as image de-skewing, Braille dots detection, Braille cell grids construction and Braille character recognition, which are less robust for complex Braille scenes. This paper proposes an optimal semantic segmentation framework BraUNet to directly detect and recognize Braille characters in the whole original Braille images. BraUNet adds extra auxiliary learning strategy to UNet network, which uses long-range connections of feature maps between encoder and decoder to get more low-level features. And auxiliary learning strategy can combine multi-class Braille characters segmentation with Braille foreground extraction, which can improve the feature learning ability and the Braille segmentation performance. Then morphological post-processing is used on semantic segmentation results to get the final individual Braille character regions. Experimental results show the proposed framework is robust, effective and fast for Braille characters segmentation and recognition on both complex double sided Braille image dataset and handwritten Braille image dataset.

1. Introduction

According to the latest WHO survey [1], there are about 1.3 billion people with some degree of vision impairment in the world. Braille is a basic writing language for the visual impaired to learn knowledge and communicate with each other. Braille documents are constructed by Braille characters or Braille cells, which are lied in Braille cell rows and Braille cell columns according to detailed Braille arrangement rules. Each Braille cell is made up of six raised or flat Braille dots arranged in three rows and two columns. So there are 64 different Braille character classes including the empty Braille cell.

Recently, many Optical Braille Recognition (OBR) systems are proposed, which focus on detecting Braille cells from Braille document images and converting them into corresponding natural language characters [2]. OBR systems are useful and meaningful to protect and republish early precious Braille books, recognize handwriting Braille documents and automatically evaluate examination papers in the special education fields, which are now mainly processed manually.

Braille cell detection and recognition is the basic technology in OBR systems. Existing methods are mainly based on traditional image processing techniques [3, 4] and machine learning methods [5, 6]. These methods are usually based on several complex steps [2], such as image de-skewing, Braille dots detection, Braille cell grid construction, and Braille character recognition. These steps are designed carefully to achieve satisfactory performance according to Braille appearance and arrangement rules.

While in some complex situations, such as the irregular arrangement of Braille cells suffered from Braille printing noise, image acquisition noise, and complex handwritten Braille images. It's difficult to design appropriate rules to convert the Braille dots to Braille characters in above complex situations. Especially for handwritten Braille images, Braille rows or paragraphs may have different skew angles. These different skew angles make it difficult to perform image de-skewing using traditional arrangement rule based methods.

With the great success of deep learning on ImageNet in 2012 [7], deep learning has made impressive progress on many difficult tasks such as image classification, object detection and semantic segmentation. However, there are very few applications in optical Braille recognition. Some existing methods mainly focus on classification of the cropped Braille character images using CNN networks. These methods cannot directly process the whole Braille image, which is difficult to apply in real applications.

This paper focuses on general and effective Braille characters detection and recognition on the whole original Braille images. We directly consider Braille characters or Braille cells as targets to detect and segment instead of Braille dots. For this task, we propose a robust OBR framework based on semantic segmentation network BraUNet and morphological post-processing method. The framework can process the original Braille images by end-to-end way without traditional complex steps.

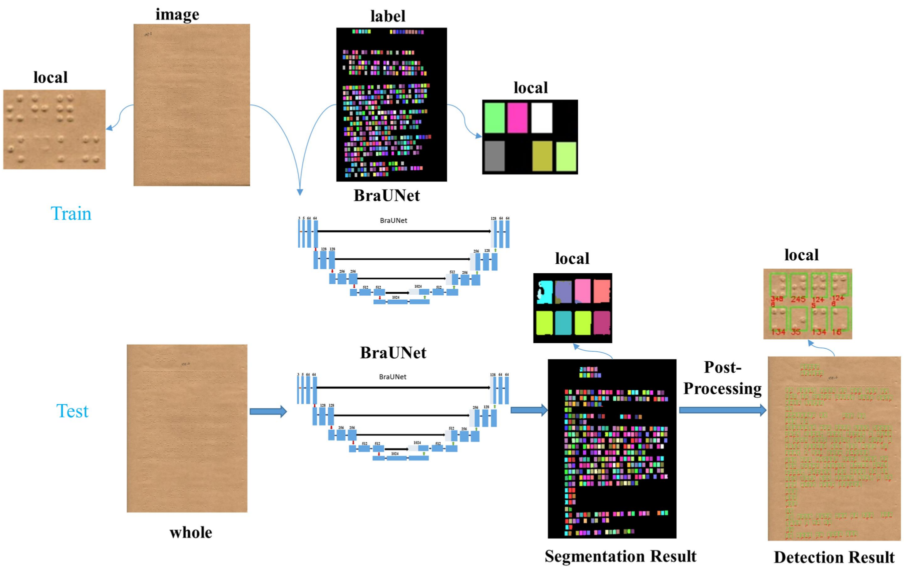


Figure 1: Our proposed framework for Braille characters detection and recognition.

The proposed BraUNet combines multi-class Braille characters segmentation task and auxiliary Braille cell foreground segmentation task to supervise the network learning process. And U-Net structure [8] uses long-range connections of feature maps between encoder and decoder to get more low-level features. These strategies can improve the feature learning ability and the whole Braille character segmentation performance. We further use morphological post-processing on semantic segmentation results to get the final Braille character regions.

The experimental results show the proposed framework is more general, robust and effective for Braille characters recognition on the public double-sided Braille image dataset and collected handwritten Braille document images.

2. Related work

Optical Braille Recognition systems are developed from 1990s [9], which can be grouped into three categories.

Traditional image processing techniques are widely used [2, 3, 4]. Antonacopoulos et al introduced a local adaptive thresholding method to segment the Braille image into three parts including shadows, light and background, and then identify Braille recto dots and verso dots by the combination rules of these parts [3]. They also constructed a Braille grid based on Braille arrangement rules to convert the Braille dots to Braille characters. However, these methods are sensitive to segmentation thresholds and designed rules.

In order to overcome the above shortcomings, some methods used machine learning techniques to recognize Braille dots. Li et al [6] adopted the cascaded classifier with Haar to quickly detect the Braille dots. Li and Yan [5] used SVM with a sliding window strategy to recognize Braille dots. Recently, Li et al [6] proposed a two-stage learning framework for double-sided Braille images recognition. A cascaded classifier with Haar is used to quickly detect the Braille dots in the first stage. Then the detected Braille dots are used for images de-skewing and constructing Braille cell girds. They also used multiple SVM classifiers with HOG or LBP features to further classify each intersection on the grid in the second stage. Namba and Zhang [11] used cellular neural network to only classify 10 Braille numbers on the cropped Braille character images.

Recently, some researchers use deep learning methods to classify the cropped Braille characters. Li et al [12] used stacked denoising autoencoder to classify 10 Braille numbers. Kawabe et al [13] trained a CNN network AlexNet to classify Braille recto dots, verso dots and background.

Above image segmentation based methods and traditional machine learning based methods for OBR may contain several steps, such as image preprocessing, de-skewing, Braille dots detection, Braille grid construction, and Braille characters recognition. These methods are limited by multi-stage processing and designed rules, which are difficult to apply to complex Braille images. Some neural network and deep learning based methods only classify cropped Braille characters with few classes or Braille dots. So far, there is a lack of research on the recognition of 64 braille characters on the whole Braille image, especially for complex double-sided Braille images and handwritten Braille images.

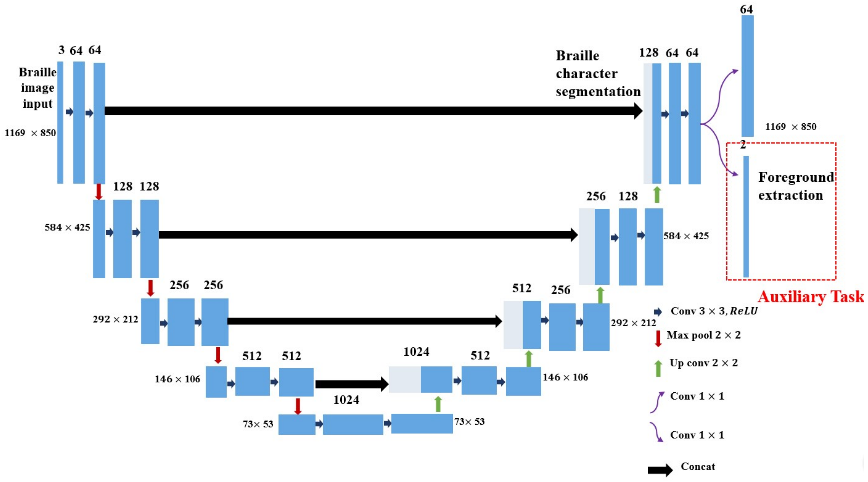


Figure 2: The architecture of our network BraUNet.

3. Our proposed work

This paper applies the semantic segmentation method in natural image analysis into the task of Optical Braille Recognition. A general and robust OBR framework is proposed based on the semantic segmentation network BraUNet and morphological post-processing techniques. Compared with the existing methods, the proposed framework can directly detect and recognize Braille characters in the whole original Braille images without relying on several steps and complicated rules.

Fig. 1 shows our proposed framework. Original Braille images and the corresponding pixel-level annotations of Braille characters with 64 classes are input into the semantic segmentation network BraUNet for training and testing. Then morphological post-processing techniques are used to refine segmentation results and get the final Braille character bounding box regions. We will introduce our framework in detail in following sections.

3.1. Network architecture of BraUNet

We propose an optimal network BraUNet for our OBR task, which is based on U-Net [8] and adopts extra foreground extraction based auxiliary task to improve the feature learning ability. U-Net structure uses long-range connections between the encoder and decoder to connect the feature maps with corresponding size, which can enhance the accuracy of low-level edge prediction and get the refined pixel labels with size of original image. U-Net is widely used in medical image segmentation due to its simple structure and good performance.

Fig. 2 shows the architecture of our network BraUNet. The left part is a contracting path used to extract high-level semantics features and the right part is an expansive path used to recover the original size gradually. The skip connection to recover the details of the segmentation result from the higher level is also adopted in our network.

This paper inputs the whole Braille images and their labeled images with Braille characters to train semantic segmentation network as Fig. 1 shows. While pixel-level segmentation for 64 Braille characters is not robust in some complex scenes due to acquisition noise, background disturbance and irregular Braille arrangement. Some segmentation errors may influence the shape of Braille characters and make wrong recognition for Braille characters. If we regard each Braille character or Braille cell as the single foreground object, the task to extract foreground from background may be simpler than multi-class semantic segmentation.

So inspired by the idea of [14], we add foreground extraction of the Braille cell as an auxiliary task at the end of the framework in our task. We train the multi-class Braille characters segmentation and the auxiliary Braille cell foreground segmentation task in the same network and calculate the corresponding loss. As shown in Fig.2, we simultaneously output the results of the foreground extraction and Braille character segmentation at the end of the network. The result of foreground extraction is used to improve the feature learning ability and supervise accurate generation of Braille cell boundary during training.

3.2. Annotations for Braille Images

The semantic segmentation network requires pixel-level annotations for model training. To alleviate the labelling workload, we simply use the bounding boxes to annotate the Braille characters on the original whole Braille images and directly convert the bounding boxes into the pixel-level annotation results. We create an empty annotation image $\mathrm{L(i,j)}$ with the same height and weight of the original image. For each of the annotated Braille characters, we can get a bounding box $R_k$and its class type c. For all position $\mathrm{i,j}$ in the R_k, we assign each pixel $\mathrm{p(i,j)}$ in image $\mathrm{L(i,j)}$ with the value c. Here c is an integer from 0 to 63 representing 64 types of Braille character. And the background and empty Braille characters are all assigned as 0. In this way, we can easily get pixel level annotations for Braille characters in Braille images. For auxiliary learning task, we just remain the background pixel as 0 value and change the rest pixel with 1 value as Braille cell foreground.

3.3. Network training

Different from the existing methods using the high-resolution Braille image such as 200dpi [6, 10] or even 600dpi [5], we only use 100dpi Braille images as the network input, which can greatly reduce the data storage, transmission and inference time. In the data preprocessing stage, we firstly down sampling the input 200dpi RGB Braille color image in DSBI dataset from 2338×1700 to 1169×850 pixels, which is the original size of the image scanned by 100 dpi.

For 64 classes Braille characters semantic segmentation task, we use the combined Dice loss [15] and Cross Entropy (CE) loss as our loss function. They are defined as follows:

$$L_{Dice} = \frac{1}{C-1}\sum_{c=1}^{C-1}(1-\frac{2\sum_i^N p^c(i)g^c(i)+\gamma}{\sum_i^N p^c(i)+\sum_i^N g^c(i)+\gamma}) \quad (1)$$

$$L_{CE} = -\frac{1}{N\times \mathrm{C}}\sum_{c=0}^{C-1}\sum_{i}^{N} g^c(i)logp^c(i) \quad (2)$$

$$L = L_{Dice} + \alpha L_{CE} \quad (3)$$

Where $p^c(i)$ and $g^c(i)$ denote the predicted value and the ground truth respectively at the position i of the whole image, C denotes the class number, which is 64 in our paper, and N denotes the total number of pixels of the whole Braille image. For smoothing purposes, we add a γ factor to both the nominator and the denominator to avoid denominator is 0. In our experiments, α is set to 1.

For the auxiliary foreground extraction task, we also use the combined Dice loss and CE loss, except that C is set to 2. Finally, the total loss is defined as follows:

$$L_{total} = L_{multi\_class} + \beta L_{auxiliary} \quad (4)$$

The weight β is set to 1 in our experiments. We train the network for 70 epochs in our experiments with the optimizer Adam and learning rate 1e-4. And the best model is selected according to the Dice value on the validation set during training process.

3.4. Post processing

In the test stage, we input the original whole Braille image into our trained BraUNet and get the initial pixel-level semantic segmentation result for Braille characters. Due to the segmentation noise and the small spacing between adjacent Braille characters, some segmented Braille characters areas are connected each other. We further use morphological processing methods to get the final bounding box of each Braille character. We binarize each class type of Braille character based on segmentation results. For each binary image, an erosion operation is used to reduce the adhesion between adjacent Braille characters, and the connected component analysis is used to extract the contour of each Braille character. To reduce noise, some small areas are removed. Finally, we take the bounding box of each contour as the detected and recognized results of Braille characters.

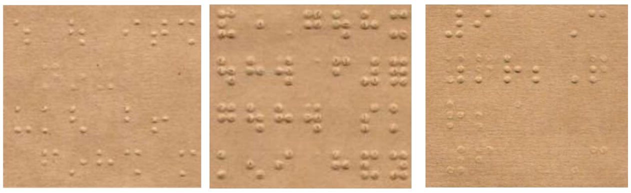
4. Experiments and analysis

4.1. Dataset

Double Sided Braille Image dataset DSBI. We adopt public Braille image dataset DSBI [10] to evaluate our method, which contains 114 double-sided Braille images from several Braille books. Li et al [6] proposed a two stage learning framework TS-OBR for Braille character recognition based on Braille dots detection on DSBI with 200dpi resolution. They used 26 Braille images for training, which is sufficient for detecting and classifying Braille dots. But for training deep learning models, such as the semantic segmentation model with 64 classes of Braille characters, 26 images are not enough for high performance. In our paper, we divide DSBI into three subsets including training, validation and test set with 74, 10 and 30 images respectively. Images from each Braille book in DSBI are proportionally sampled to construct the three subsets. And recto Braille characters and verso Braille characters detection and recognition are all evaluated in our experiments. We use 200dpi resolution of Braille image for TS-OBR and 100dpi for U-Net and BraUNet methods which are obtained by down sampling images directly.

Braille Answer Sheet Dataset BAS. To evaluate our method for handwritten Braille recognition, we collect some Braille answer sheets from a special education school. These answer sheets are all written by different students using some certain Braille boards. It is challenge for handwritten Braille recognition.

The biggest problem in handwritten Braille is that Braille characters in different lines or paragraphs in the same page may have different skew angles. This is usually caused by the location change of the Braille writing board, which makes rule-based methods fail. On the other hand, different students may have different writing habits which leads to various appearance of Braille dots as shown in



(a) Tiny Braille dots (b) Thick dots (c) Erased dots

Figure 3: Some handwritten Braille dots and Braille characters.

Fig.3(a) and Fig.3(b). Another difficulty is that many students modify certain Braille characters by directly erasing one or more Braille dots in one Braille character, which is more difficult to distinguish whether these Braille dots are erased or not by visual information as Fig.3(c). We collect total 50 Braille answer sheets images from 32 students. These handwriting Braille images are single sided with recto Braille dots.

4.2. Metrics

Dice value [8] is adopted to evaluate the performance of semantic segmentation results with or without auxiliary task of U-Net model. For Braille characters detection and recognition performance, we also use the Precision, Recall and F1 values in [10] as the evaluation metrics. In addition, we use the Intersection over Union (IOU) to evaluate the degree of overlap of two Braille character boxes. For each predicted Braille character box, we use all the ground truth of Braille character boxes to calculate the IOU value. If the maximum IOU value is greater than threshold T, we assume that the predicted Braille character box is correct. In our experiments, T is set to 0.5. We denote the number of Braille characters correctly classified as TP, the number of Braille characters misclassified but with the correct position as FP, the number of Braille characters with the wrong position as WP, and the number of Braille characters missed as TN. The Precision, Recall and F1 values can be defined as follows:

$$\text{Pre} = \frac{TP}{TP + FP + WP} \tag{5}$$

$$\text{Rec} = \frac{TP}{TP + FN} \tag{6}$$

$$\text{F1} = \frac{2 \times Pre \times Rec}{Pre + Rec} \tag{7}$$

4.3. Experimental results

4.3.1. Experiment setting

To evaluate our proposed method, we conduct two experiments to compare semantic segmentation network BraUNet with U-Net, and existing method TS-OBR [6]. U-Net means the original network [8] and BraUNet means our U-Net with auxiliary foreground segmentation task. U-Net and BraUNet are all deep learning based methods. We also compare our method with the recent method TS-OBR [6] based on traditional machine learning method. TS-OBR uses a cascaded classifier to quickly detect the Braille dots and then de-skews image to construct a Braille cells grid, and further adopts SVMs to classify each intersection on the grid for Braille dots. This method relies on the construction of Braille cells grid, which is sensitive to noise and irregular arrangement. We retrain and test TS-OBR method on our newly divided sets of DSBI with 200dpi resolution of Braille images.

All the models of U-Net, BraUNet and TS-OBR are trained only on the public double-sided Braille dataset DSBI. We evaluate the semantic segmentation and detection performance including recto Braille characters and verso Braille characters on DSBI.

We further use single-sided Braille answer sheet dataset BAS as test set to evaluate the generalization ability of Braille character detection and segmentation methods, which are trained on DSBI dataset.

4.3.2. Results of semantic segmentation

Table 1 shows the results of pixel-level semantic segmentation performance of Braille characters for U-Net and BraUNet on both DSBI and BAS datasets. Fig.4 shows the Dice columns figure on DSBI and BAS datasets. Dice value is used to evaluate the result of semantic segmentation performance without any post-processing. On DSBI dataset, the Dice value of our BraUNet improves 5.48% for recto Braille characters than U-Net, and improves 3.17% for verso Braille characters. From table 1, we can also find the Dice value of BraUNet drop from 0.9508 on DSBI to 0.9048 on BAS, which maybe some data distribution of single-sided handwritten Braille characters are different from those of double-sided printed Braille documents. This problem can be resolved by adding training sample from BAS for fine tuning model. And on BAS dataset, the Dice value of BraUNet improves 5.43% compared with U-Net for recto Braille characters segmentation. Fig.4 also shows above conclusion.

Table:1 Semantic segmentation results of Braille characters for U-Net and BraUNet.

Dataset	Type	Method	Dice
DSBI	Recto Braille character	U-Net	0.8960
		BraUNet	**0.9508**
	Verso Braille character	U-Net	0.9040
		BraUNet	**0.9357**
BAS	Recto Braille character	U-Net	0.8505
		BraUNet	**0.9048**

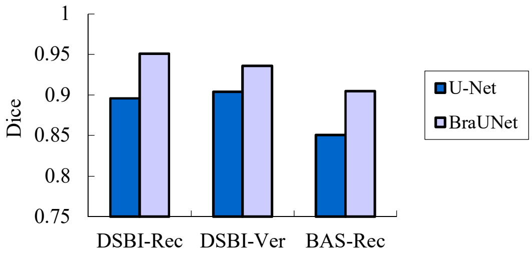
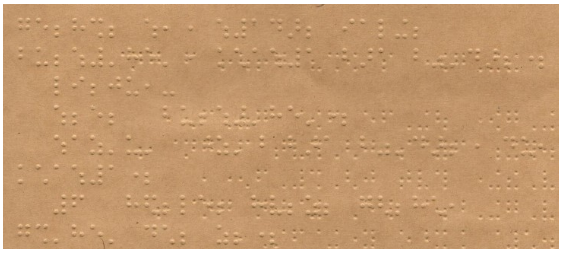
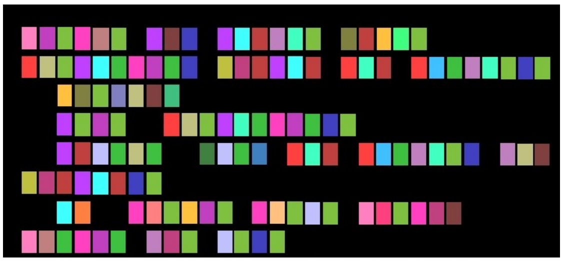


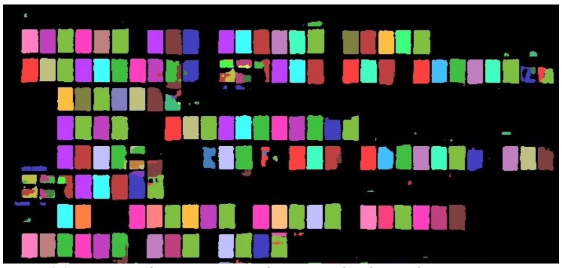
Figure 4: Braille character segmentation performance.



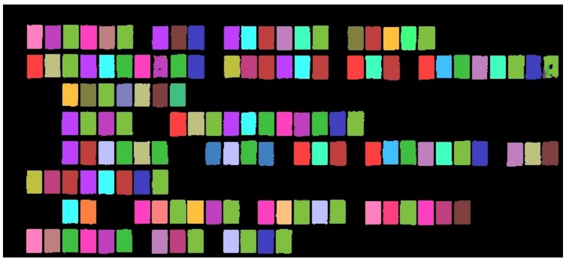
(a) Local region of source Braille image in DSBI.



(b) Manual labels with 64 classes of Braille characters.



(c) Semantic segmentation results based on U-Net.



(d) Semantic segmentation results based on BraUNet.

Figure 5: Semantic segmentation results from a local regions of Braille images in DSBI. Different color means different class of Braille characters.

The above results show the effectiveness and generation performance of our optimal for both recto and verso Braille character segmentation.

Fig.5 shows some semantic segmentation results of recto Braille characters from a local part of one Braille image on DSBI dataset. Fig.5(a) is a local region of an original double-sided Braille image and Fig.5(b) is the ground truth label of recto Braille character with 64 different colors. Fig.5(c) is the semantic segmentation results of U-Net model, which shows many noise and wrong segmentation pixels especially on the edge region of each Braille character. Fig.5(d) is the results of BraUNet, which adds auxiliary foreground segmentation task to U-Net. It's clear that the outline of Braille characters are improved compared with the results of U-Net in Fig.5(c). And the segmentation and recognition noises and errors of some Braille characters in Fig.5(c) are improved in Fig.5(d).

4.3.3. Results of Braille characters detection

Table 2 shows the detection performance of Braille characters for U-Net, BraUNet and TS-OBR on both DSBI and BAS datasets. The metrics of precision, recall and F1 are used to evaluate detection performance. Based on above semantic segmentation results, we further use morphological post-processing methods in section 3.4 to get the final bounding box of each Braille character.

On DSBI dataset, despite the disturbance of the Braille dots on the back page and the insufficient amount of training data, original U-Net network still can achieve 0.9751 and 0.9768 F1 values for the recto and verso Braille character recognition. These results show that the end-to-end semantic segmentation model is useful for Optical Braille Recognition. With the help of auxiliary task, our BraUNet network finally gets 0.9966 and 0.9893 F1 values for recto and verso Braille characters respectively.

On BAS dataset, we get 0.9425 and 0.9638 F1 value for recto Braille characters detection for U-Net and BraUNet respectively. Nearly 2% improvements of F1 value shows our auxiliary task is effective, which improves the feature learning ability and segmentation performance for complex double-sided Braille and handwritten Braille images.

Table 2 also shows the comparative results of our deep learning method and existing TS-OBR method [6]. The F1 values are similar for BraUNet and TS-OBR on DSBI dataset including recto and verso Braille character detection. While TS-OBR used 200dpi resolution of Braille images which are double size of width and height compared with our 100dpi for BraUNet.

On BAS dataset, TS-OBR just gets the 0.9408 F1 value, which is about 2.3% lower than BraUNet. This maybe there are many irregular arrangements of Braille characters in handwritten Braille documents, which will bring error to construct the accurate Braille cell grid using TS-OBR, so some Braille characters could not be obtained correctly.

Table 2: Comparative results of Braille characters detection.

Dataset	Type	Method	Dpi	Pre-%	Rec-%	F1
DSBI	Recto Braille character	U-Net	100	98.35	96.69	0.9751
		BraUNet	100	99.43	99.88	**0.9966**
		TS-OBR	200	99.28	99.96	0.9962
	Verso Braille character	U-Net	100	98.31	97.06	0.9768
		BraUNet	100	98.81	99.05	0.9893
		TS-OBR	200	98.44	99.70	**0.9906**
BAS	Recto Braille character	U-Net	100	92.90	95.64	0.9425
		BraUNet	100	93.50	99.44	**0.9638**
		TS-OBR	200	89.47	99.19	0.9408

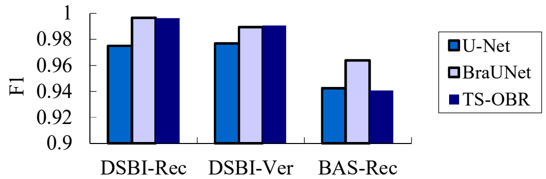


Figure 6: Braille character detection performance.

While our proposed BraUNet is more robust and can end-to-end get the Braille character recognition results without multiple complex steps and designed rules in TS-OBR. Fig.6 shows the F1 columns figure of U-Net, BraUNet and TS-OBR for Braille characters detection on DSBI and BAS datasets, which also shows the effectiveness of our optimal BraUNet compared with U-Net and TS-OBR methods.

We implement our BraUNet framework using the deep learning framework PyTorch with one GPU 1080Ti. The average processing time of Braille character segmentation and recognition is about 0.25s for one Braille image.

5. Conclusion

This paper introduces an effective Braille segmentation and recognition framework for whole Braille images. We propose an optimal semantic segmentation network BraUNet with auxiliary learning task by end-to-end way. This auxiliary learning task can combine multi-class Braille characters segmentation and Braille cell foreground extraction, which can improve the feature learning ability and segmentation performance of Braille characters. The morphological processing algorithms are used to get the final Braille detection results. The proposed framework is effective and general, which can directly detect and recognize Braille characters in the original Braille images without relying on Braille dots detection, image de-skewing and Braille arrangement rules. The experimental results on public double-sided Braille image dataset and collected Braille answer sheet dataset show the robustness and effectiveness of BraUNet. In the future, we will collect more complex Braille images to test and improve OBR performance.

Acknowledgement

This work is supported in part by Beijing Haidian Original Innovation Joint Foundation (L182054).

References

[1] Visual impairment and blindness, World Health Organization, 2018.

[2] Samer Isayed and Radwan Tahboub. A review of optical braille recognition. In WSWAN, pages 1–6. IEEE, 2015.

[3] Apostolos Antonacopoulos and David Bridson A robust braille recognition system. InDAS, pages 533–545, 2004.

[4] Majid Yoosefi Babadi and Shahram Jafari. Novel grid-based optical braille conversion: from scanning to wording. International Journal of Electronics,98(12):1659–1671,2011.

[5] Lie Li and Xiaoguang Yan. Optical braille character recognition with support-vector machine classifier. In ICCASM2010, volume 12, pages V12–219. IEEE, 2010.

[6] Renqiang Li, Hong Liu, Xiangdong Wang, and Yueliang Qian.Effective optical braille recognition based on two-stage learning for double-sided braille image. In PRICAI, 2019.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural net-works. In NIPS, pages 1106–1114, 2012.

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In MICCAI, pages 234–241, 2015.

[9] Jan Mennens, Luc Van Tichelen, Guido Francois, and Jan J.Engelen. Optical recognition of braille writing. In ICDAR, pages 428–431, 1993.

[10] Renqiang Li, Hong Liu, Xiangdong Wang, and Yueliang Qian. DSBI: Double-sided braille image dataset and algorithm evaluation for braille dots detection. In ICVIP, pages 65–69. ACM, 2018.

[11] Michihiro Namba and Zhong Zhang. Cellular neural network for associative memory and its application to braille image recognition. In IJCNN, pages 2409–2414, 2006.

[12] Ting Li, Xiaoqin Zeng, and Shoujing Xu. A deep learning method for braille recognition. International Conference on Computational Intelligence and Communication Networks, pages 1092–1095. IEEE, 2014.

[13] Hiroyuki Kawabe, Yuko Shimomura, Hidetaka Nambo, andShuichi Seto. Application of deep learning to classification of braille dot for restoration of old braille books. In ICMSEM, pages 913–926. Springer, 2018.

[14] Wenjia Wang, Junxuan Chen, Jie Zhao, Ying Chi, XuansongXie, Li Zhang, and Xian-Sheng Hua. Automated segmentation of pulmonary lobes using coordination guided deep neural networks. ISBI 2019, pages 1353–1357, 2019.

[15] Carole H Sudre, Wenqi Li, Tom Vercauteren, et al. Generalized dice overlap as a deep learning loss function for highly unbalanced segmentations. In Deep learning in medical image analysis and multimodal learning for clinical decision support, pages 240–248, 2017.