ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/335360748

# Effective Optical Braille Recognition Based on Two-Stage Learning for Double-Sided Braille Image

**Chapter** *in* Lecture Notes in Computer Science · August 2019

DOI: 10.1007/978-3-030-29894-4_12

CITATIONS
11

READS
655

**4 authors**, including:

Hong Liu
Institute of Computing Technology, Chinese Academy of Sciences

**109** PUBLICATIONS **1,045** CITATIONS

SEE PROFILE

All content following this page was uploaded by Hong Liu on 22 March 2021.

The user has requested enhancement of the downloaded file.

# Effective Optical Braille Recognition Based on Two-stage Learning for Double-sided Braille Image

Renqiang Li, Hong Liu *, Xiangdong Wang, and Yueliang Qian

Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

**Abstract.** This paper proposes a novel two-stage learning framework TS-OBR for double-sided Braille images recognition. In the first stage, a Haar cascaded classifier with the sliding window strategy is adopted to quickly detect Braille recto dots with high confidence. Then a coarse-to-fine de-skewing method is proposed to correct original skewed Braille images, which maximizes the variance of horizontal and vertical projection at different angles. And an adaptive Braille cells grid construction method based on statistical analysis is proposed, which can dynamically generate the Braille cells grid for each Braille image. In the second stage, a decision-level SVM classifier with four classifiers recognition results is used to get recto dots detection results only on intersections of the Braille cells grid. Experimental results on the public double-sided Braille dataset and our Braille exam answer paper dataset show the proposed framework TS-OBR is effective,robust and fast for Braille dots detection and Braille characters recognition.

**Keywords:** Optical Braille recognition · De-skewing · Braille dots detection · Braille cell location · Double-sided Braille.

## 1 INTRODUCTION

There are about 1.3 billion visual impaired and 36 million blind in the world according to the WHO [13]. Braille is a tactile writing system for the visually impaired to learn knowledge and obtain information, which is designed by Frenchman Louis Braille. Automatic recognizing Braille document images into Braille characters can be called Optical Braille Recognition (OBR) [8]. OBR system is meaningful and important for republication of numerous early and valuable Braille books, translation of Braille exam papers in the special education field, and communication with others.

The Braille document consists of Braille characters based on rectangular cells, which contain six dots arranged in three rows and two columns. Many Braille books are double-sided to save papers, which may contain recto dots and verso dots in one Braille document. Double-sided OBR is a challenging task for the diversity of Braille papers, disturbance of complex arrangement of recto and verso dots, deformation and skewness of Braille images by acquisition noise.

* Corresponding author (Email: hliu@ict.ac.cn)

Generally, the OBR system contains several steps including Braille image acquisition, image de-skewing, Braille dots recognition and Braille cells location and recognition [8]. Image segmentation based OBR methods are widely used, which segment the Braille dots from the background and design some rules to identify them.

Antonacopoulos et al. [5] segmented the Braille image into shadow, highlight and background regions, and then identified them as recto dots or verso dots according to different combinations of highlight and shadow regions. Al-Shamma et al. [4] used canny method to detect dot edges and used holes filling and image filtering to detect Braille dots. They tested on several scanned single-sided Braille documents with the average time of 32.6 seconds for one document. Above segmentation based methods are simple and affected by designed rules and segmentation threshold values, and not robust for complex Braille images.

Some work used statistical learning methods to detect Braille dots and recognize Braille cells. M.Namba et al. [12] applied the neural network based on associative memory to classify Braille cell images into ten classes and obtained 87.9% recognition rate on Braille cell images. This method cannot deal the whole Braille image which is not suitable in real applications. Li et al. [9] used SVM with the sliding window strategy to recognize Braille characters, which needed 20 minutes to process one Braille image with the classification error of 5%. This method is time-consuming with low performance.

Recently, Li et al. [10] released the first public Braille images dataset DSBI [1]. They also proposed a Haar+Adaboost with the sliding window strategy to detect recto dots. Their strategy got 0.970 F1 value for recto dots detection on the DSBI dataset. However, they only evaluated on de-skewing images for recto dots detection. And the performance is not good enough, since even little error rate may lead dozens of Braille dots recognized wrong for a Braille page with over one thousand dots averagely.

Besides Braille dots detection, in real applications, one OBR system also should process the original Braille images with certain degree of skewness and deformation from acquisition noise or human errors. Most of existing OBR methods didn't mention this issue, some de-skewing methods are mentioned lightly and time-consuming in practice.

Braille cells locating is also a crucial stage of OBR systems. Most of existing methods are based on the standard arrangement of Braille dots, which are not robust for some complex situations with sparse Braille characters, incomplete Braille row or column groups and image deformation.

This paper focuses on the effective and quick double-sided Braille image recognition task for original images. The main contributions are as followings.

We propose a novel Two-Stage learning framework for double-sided Braille image recognition called TS-OBR.There are several advantages of our proposed framework TS-OBR, which contains the whole processing including image de-skewing, location of Braille cells and Braille dots detection and Braille characters recognition. We can directly process the original Braille images with skewness and deformation. Our system can recognize recto and verso Braille characters

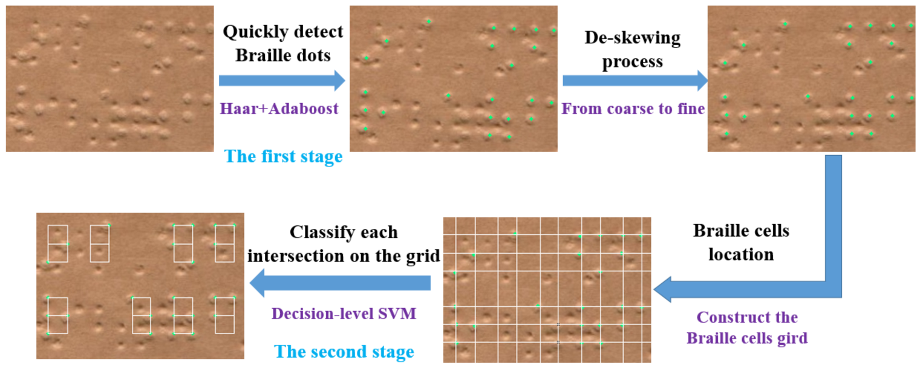


**Fig. 1.** Our proposed framework TS-OBR.

together so that we can only scan and process one side of the double-sided Braille image in real applications. Experimental results on the DSBI dataset and our Braille exam answer paper dataset show the effectiveness of our proposed method. The average F1 value is over 0.997 for both recto and verso dots detection and over 0.994 for Braille characters recognition on the DSBI. We also achieve an excellent performance with the average F1 value 0.992 for recto dots detection on our Braille exam answer paper dataset BEP directly using the model only trained on the DSBI. Unlike the DSBI, which is from printed Braille books, our dataset BEP comes from manually stabbed paper by blind student.The whole system TS-OBR only costs about 1.5 seconds for recto and verso dots detection respectively with high recognition rate on the two datasets.

## 2 THE PROPOSED FRAMEWORK

The proposed framework TS-OBR is shown in Fig.1. The DSBI dataset provides the original color images and de-skewing color images. We just use original images to process. Firstly, we convert the original color image to the gray image. In our first stage, different from other methods, we use Haar+Adaboost with the sliding window strategy to quickly detect Braille dots rather than performing the de-skewing process. Then, the position of detected Braille dots will be used by a coarse-to-fine method to find an appropriate angle to correct the skewed image. A flexible Braille cells grid is then constructed to locate the Braille cells based on detected Braille dots and the statistical information of Braille cells arrangement. In our second stage, we further classify each intersection of the Braille cells grid using a decision-level SVM classifier to improve the performance of OBR.

## 3 OUR MAIN WORK

### 3.1 The first stage for Braille dots initial detection

Braille dots detection is the most crucial task in OBR systems. The existing segmentation based methods are not robust for complex double-sided Braille

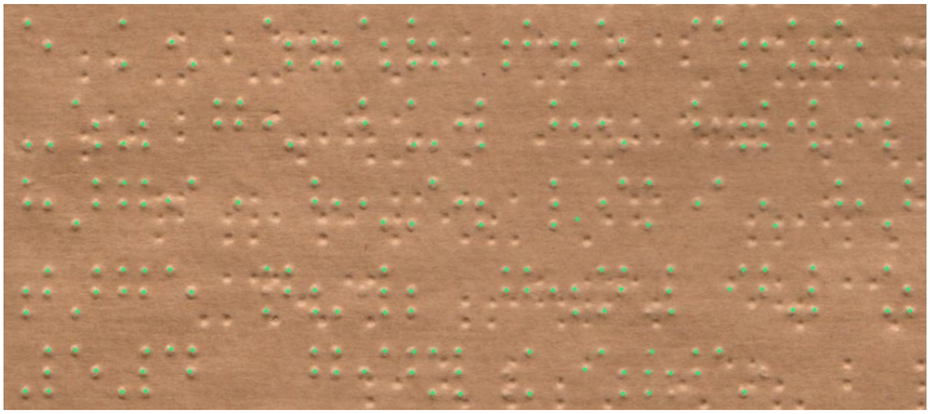

**Fig. 2.** Braille recto dots detection results with Haar+Adaboost.

images. This paper regards Braille dots detection as general object detection in nature images. So we also adopt the classical object detection method, which uses Haar feature and Adaboost classifier [10, 14] to detect Braille dots.

Different from [10], our first stage is only used for the initial Braille dots detection. Fig.2 shows one local region of recto dots detection with Haar+Adaboost and most of recto dots are detected successfully and reliably. Although some recto dots are missing, this problem can be resolved in our second stage with a higher performance classifier. Experimental results show that our first stage only takes average 0.29 seconds for each image with 200dpi in the DSBI dataset.

### 3.2 De-skewing by a coarse-to-fine strategy

The standard arrangement of Braille dots and cells should be regular in both horizontal and vertical directions, while the acquired Braille images have much noise. So de-skewing process is necessary and important. Most of original Braille images have a slight skewness in the DSBI dataset and our dataset, which will make the Braille cells location and Braille characters recognition more difficult.

Many existing de-skewing methods, including Linear regression [6] and Hough Transform [5], are not robust and fast for complex double-sided Braille images. Mennens et al. [11] used the deviation over the sum of the rows to calculate the image rotation angle. The image was slanted one pixel in vertical direction each time.

This paper proposes a coarse-to-fine de-skewing method with two levels angel interval by maximizing the variance of horizontal and vertical projection. And the angle with the maximum variance is the skewed angle. The details are described in algorithm 1.

In our experiments, the parameters are set $s = 5$, $angle_1 = 5$, $angle_2 = 0.5$, $ls = 1^\circ$, $ss = 0.02^\circ$ .

We use the coarse-to-fine projection strategy to quickly get the skewed angle with average 0.57 seconds for each image.

**Algorithm 1: De-skewing by coarse-to-fine strategy**

1. Create a white binary image $b(x, y)$ with the same size of the original image $oi(x, y)$.
2. Replace each detected dot $d(x, y)$ at the same position $(x, y)$ in $b(x, y)$ with a black square with size of $s \times s$ .
3. In the coarse de-skewing stage, rotate the image $b(x, y)$ from $-angle_1^\circ$ to $+angle_1^\circ$ with a large step $ls$.
4.For each angle $angle^\circ$, project the rotated binary image $b_a(x, y)$ in horizontal and vertical directions, and obtain the horizontal frequencies $h$ and vertical frequencies $v$.
5. Calculate the projection variance $var$ for $angle^\circ$ :

$$\overline{h} = \frac{1}{n}\sum_{i}^{n} h_i, \overline{v} = \frac{1}{n}\sum_{i}^{n} v_i \tag{1}$$

$$var_h = \frac{1}{n}\sum_{i}^{n} (h_i - \overline{h})^2, var_v = \frac{1}{n}\sum_{i}^{n} (v_i - \overline{v})^2 \tag{2}$$

$$var = var_h + var_v \tag{3}$$

6. Select the rotated angle with the maximum $var$ as $a1^\circ$.
7. In the fine stage, rotate the binary image $b(x, y)$ from $(a1 - angle_2)^\circ$ to $(a1 + angle_2)^\circ$ with a small step $ss$ to find a more accurate rotated angle.
8. Select the rotated angle with the maximum $var$ as $a2^\circ$.
9. Rotate the original image $oi(x, y)$ with angle $-a2^\circ$, get the de-skewing image $dsi(x, y)$ and de-skewing dots $DSD$.

### 3.3 Dynamic Braille cells grid location

Each Braille document consists of hundreds of Braille characters and each Braille character has a rectangular block called a Braille cell, which contains six Braille dots arranged in three rows and two columns. These six dots can be tiny bump called recto dots or flat to represent a certain Braille character. In ideal situations, the Braille cells grid is a regular grid, and the distances among horizontal and vertical lines are regular and easy to estimate at a specific resolution of scanned Braille images.

Most of Braille cells grid location methods are based on arrangement rules of Braille characters. Some methods used a preset fixed grid and selected a Braille dot as the starting point to construct a regular grid [3, 9]. To enhance the robustness, Antonacopoulos et al. [5] used an adaptive method to form the Braille cells gird rather than the preset grid by calculating the distances of Braille rows and columns for each Braille document.

While in real applications, this grid is usually deformed due to acquired noise, thus the distances between lines are usually not fixed even in the same document. Besides these noise and deformation, Braille cells grid location is also difficult in some complex situations.

We introduce a robust and flexible method to construct the Braille cells grid by statistical information. The Braille cells grid can be dynamically generated by distribution of Braille dots. Our method contains four steps: detect Braille lines, group Braille lines, add missing Braille lines and construct the cells grid, which we will describe as followings.

**Detect Braille lines.** With the de-skewing image and Braille dots, we firstly sort all the detected dots in ascending order by y coordinates and generate the first horizontal line according to the first dot. Then the distance of subsequent dots to the line is calculated. If the distance is below the threshold $TH_1$, it will be added to this line and the position of this line is updated by the average of y coordinates on this line dynamically. Otherwise, a new horizontal line will be generated. In this way, some candidate horizontal lines are extracted. Then we remove some very close lines and only remain those with many dots to reduce the influence of wrong detected dots.

**Group Braille lines.** Based on the detected Braille lines, we select some reliable groups of three continuous horizontal lines from top to down, according to the arrangement rules of Braille cells. Set $\{hl_1, hl_2, hl_3\}$ is one group of lines, $d_{12}$ and $d_{23}$ are the distances of $hl_1$ and $hl_2$, $hl_2$ and $hl_3$ respectively, which should satisfies the following constraint:

$$d_{12}, d_{23} \leq TH_2 + \alpha \quad \&\& \quad d_{12}, d_{23} \geq TH_2 - \alpha \tag{4}$$

Where $TH_2$ is the distances of lines in one Braille cell, and $\alpha$ is a penalty factor. We will update the value of $TH_2$ according to the $d_{12}$ and $d_{23}$ in each line group dynamically. Some overlapped line groups will be removed by statistical analysis.

**Add missing Braille lines.** The above steps can form some reliable groups with three lines. Other line groups will be inserted according to the remaining lines and the regular distance to make sure that line groups are placed in the regular interval by y coordinates.

**Construct cell grid.** The above steps construct the whole horizontal lines and line groups of current Braille image. Then the similar process is applied for detecting and grouping vertical lines according to x coordinates while each group only contains two vertical lines.

The threshold values in our method change dynamically to adapt the complex situations and could be tolerate some errors. Braille images in the DSBI dataset are scanned with 200dpi, the initial parameters are:$TH_1 = 5$, $TH_2 = 21$, $\alpha = 3$ in our experiments. Experimental results show the effectiveness of our proposed Braille cells grid location method for complex double-sided Braille images.

### 3.4 The second stage for High-precision OBR

With above accurate Braille cells grid information, we can further adopt a relatively complex and high performance machine learning method for Braille dots detection and Braille characters recognition. This stage can only process on the vertexes of Braille cells, also the intersections of Braille cells grid, instead of the whole image. This strategy is not mentioned in existing methods for OBR.

This paper selects the feature of Histogram of Oriented Gradient (HOG) [7], Local Binary Pattern (LBP) [2] and fused HOG_LBP to train three SVM classifiers. We classify each intersection of the grid using them.

For better performance of Braille image recognition, the initial detection result will also be converted to the result based on the grid by assigning each detected dot to the nearest intersection. And then, we use a decision-level SVM

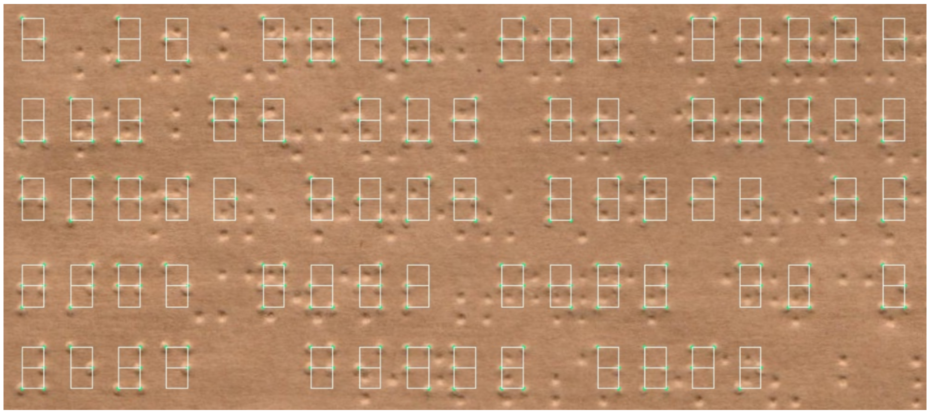

**Fig. 3.** Results of the second stage. The green dot is the detected recto dots and the white rectangle is the located Braille cells.

to fuse the converted result and the classified results by three SVMs. We take these results as a 4-dimension feature to train the decision-level SVM for fusion so that we don't have to be stuck on how to balance the weight of each result.

The processing time has been reduced greatly in this strategy compared with general sliding window strategy. Our method can also help avoid the wrong dots detection beyond the grid intersections and then improve the precision and recall rate. One region of the recto dots detection and cells location result by the second stage is shown in Fig.3.

## 4 EXPERIMENTS AND ANALYSIS

### 4.1 Dataset

Unlike most existing methods which are tested on their small datasets,we use two Braille datasets to evaluate our method. Firstly, we choose the double-sided dataset DSBI [10]. DSBI is the first public and only Braille image dataset available, which contains 114 double-sided Braille images from several Braille books. These Braille images are acquired by the flatbed scanner with 200dpi and the resolution is about $1700 \times 2338$ pixels. Some of Braille images have defects such as oil stains, distortion, cracks and abrasion Braille dots, which are complex and difficult for OBR.

The training set of DSBI contains 26 Braille images from 4 books and the test set contains remained 88 Braille images from all the books. This dataset also provides both the original Braille images and de-skewing images with detailed position annotation information of recto and verso dots, which can be used to evaluate the performance of dots and characters recognition.

In order to further verify the effectiveness of our proposed framework TS-OBR, we also constructed a Braille exam answer paper dataset called BEP. We

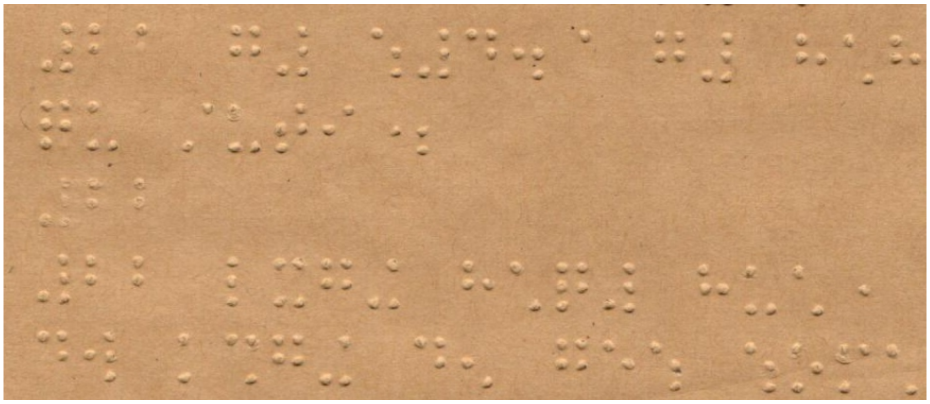

**Fig. 4.** One local region of the Braille exam paper with some erased and various appearance dots.

**Table 1.** Statistical information of Braille recto dots and Braille characters.

| Dataset | Recto dots | Braille characters | Total pages | Evaluated pages |
|---|---|---|---|---|
| DSBI | Average: 1071<br>ALL: 122117 | Average: 401<br>ALL: 45725 | 114 | 88 |
| BEP | Average: 1147<br>ALL: 109034 | Average: 406<br>ALL: 38659 | 95 | 95 |

have collected a total of 95 Braille exam answer papers from 28 blind students. Unlike Braille books, Braille papers are stabbed by blind students themselves using the special Braille writing tablet. Braille exam answer papers are usually one-sided, and the size of Braille dots stabbed varies greatly because of the difference of stab habits and strength.

What's more, many wrong-stabbed Braille dots are usually erased directly. But these erased dots are very difficult to distinguish from ordinary Braille dots visually, which are usually inferred by Braille teachers combined with context semantics. One local region of the Braille exam paper with some erased dots in our BEP is shown as Fig.4. Therefore it's more challenging than the recognition of printed Braille books for these cases. We use all the pages in dataset BEP to evaluate the performance and generalization ability of our framework. And the models are the same as DSBI and only trained on the training set of DSBI.

Unlike other datasets, each image contains only one or few objects, one Braille image contains averagely over 1,000 very small recto dots and hundreds of Braille characters. Table 1 gives us the statistical information of recto dots and Braille characters in both datasets. There are averagely over 1000 recto dots and around 400 Braille characters each Braille image no matter in the DSBI and BEP. The total number is very huge about over 100 thousands dots and around 40 thousands characters in each dataset. And we use 88 pages in DSBI and all 95 pages in BEP for evaluation.

### 4.2 Metrics

We adopt precision, recall and F1 value to evaluate our method, which is used in [10]. Besides Braille dots detection, we also evaluate the performance of Braille characters recognition for double-sided Braille images. These metrics can be defined as follows:

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

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

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

### 4.3 Experiment details

All the experiments in this paper are carried out by the ordinary computer with Intel i7-6700@3.40 GHz and 16G RAM without GPU. We also evaluate our framework on several laptops. It's very fast and easy to run without extra setup so that Braille teachers and others can easily make use of our work to save their valuable time.

**Haar+Adaboost training.** For recto dots training, we collect 9690 recto dots regions as positive samples and 28212 negative samples from the background and verso dots regions. The sample size is $20 \times 20$. We train a 7 cascaded classifier of Haar+Adaboost for recto dots using OpenCV. For verso dots training, we collect 10206 positive samples and 15016 negative samples to get a 9 cascaded classifier.

**SVMs training.** For recto dots training, we collect 26908 positive and 33853 negative samples with the size of $24 \times 24$ to train three SVMs. For verso dots training, we collect 26590 positive and 33806 negative samples. For HOG feature, we adopt the block size of $16 \times 16$, the cell size of $4 \times 4$, bin number of 9 and get 1296 dimension feature. For LBP feature, we take the Uniform Pattern LBP with the cell size of $8 \times 8$ and get the 522 dimension feature. Then the fused HOG_LBP feature has 1818 dimension. Then the final decision-level SVM is trained by the 4 dimension feature vector from the detection results of Haar+Adaboost and three SVMs.

### 4.4 Results and Analysis

To objectively evaluate the performance of our proposed framework TS-OBR, we compare our method with the Braille dots detection in [10] which gave the results of segmentation based method and Haar+Adaboost method for only recto dots. Our paper gives the results of dots detection and Braille characters recognition including recto and verso. In order to evaluate the performance and generalization ability of our proposed framework TS-OBR, we also evaluate on

**Table 2.** Results of Braille recto dots detection.

| Method | Images | Pre | Rec | F1 | Time |
|---|---|---|---|---|---|
| Segment [10] | De-skewing | 91.72% | 98.11% | 0.948 | / |
| Haar [10] | De-skewing | 97.65% | 96.38% | 0.970 | / |
| Haar(ours) | Original | 98.38% | 95.75% | 0.970 | 0.89s |
| HOG_LBP_SVM | De-skewing | 93.14% | 98.69% | 0.958 | 15.02s |
| SVM_Grid | Original | 99.31% | 99.97% | 0.996 | 1.22s |
| **TS-OBR** | **Original** | **99.65%** | **99.97%** | **0.998** | **1.45s** |

our constructed Braille exam answer paper dataset BEP. To analysis the proposed method in detail, we also summary processing time and other statistical information.

#### 4.4.1 Results on the public dataset DSBI

**Braille recto dots detection.**Table 2 gives the recto dots detection results. The first two lines are from [10], which are on the de-skewing Braille images. Our first stage of the Haar+Adaboost with sliding windows method got the same F1 value 0.970, and a higher precision rate 98.38% compared with 97.65% of Haar in [10]. Since we want to ensure the detected dots are more reliable, we remain a high precision rate in our first stage. We also test the method of the HOG_LBP with SVM called HOG_LBP_SVM using the sliding window strategy on the entire de-skewing Braille image. And it got the 0.958 F1 value, which is 0.01 higher than the image segmentation based method but 0.012 lower than the Haar method. The initial recto dots detection by our Haar only took average 0.89 second (including the image de-skewing time) for a Braille image. But the HOG_LBP_SVM took average 15.02 seconds, which is time-consuming and about 17 times that of our Haar method.

The method SVM_Grid means that we only apply a SVM classifier with HOG_LBP on the cells grid, which got the 0.996 F1 value just using 1.22 seconds. This F1 value is higher than above four methods and is 0.038 higher than HOG_LBP_SVM with the sliding window strategy method. But our SVM_Grid used much less time, which can reduce much wrong detection on the background and also greatly reduce the number of windows to extract features and recognize.

This last method in Table 2 is our proposed framework TS-OBR, which uses two-stage dots detection. It got the highest F1 value 0.998 for recto dots detection, which is 0.028 higher than Haar in [10] and 0.002 than SVM_Grid. Our framework TS-OBR just took average 1.45 seconds to process one Braille image in the DSBI dataset. There are mainly three time-consuming steps in our framework, and the average time of them are 0.29 seconds for Haar+Adaboost, 0.57 seconds for de-skewing process and 0.52 seconds for decision-level SVM.

**Statistical analysis of recto dots detection.** In order to more intuitively illustrate the effectiveness of our TS-OBR framework, the average correctly detected dots number TP, wrong detected dots number FP, and missing dots number FN for recto dots detection of our methods are given in Table 3. There are

**Table 3.** Statistical information of Braille recto dots detection.

| Method | Images | TP | FP | FN |
|---|---|---|---|---|
| Haar(ours) | Original | 1039.38 | 17.13 | 46.15 |
| **TS-OBR** | **Original** | **1085.16** | **4.03** | **0.34** |

**Table 4.** Results of Braille verso dots detection.

| Method | Images | Pre | Rec | F1 | Time |
|---|---|---|---|---|---|
| Haar(ours) | Original | 99.53% | 81.37% | 0.895 | 0.86s |
| **TS-OBR** | **Original** | **99.77%** | **99.74%** | **0.997** | **1.58s** |

averagely about 1085 recto dots on one double-sided Braille image in the DSBI dataset, which is much more than the general number of objects in natural image. As Table 3 shows, although in first stage, the Haar method had a high F1 value of 0.970, the average wrong detected dots number is 17.13 and the average missing dots number is 46.15.

**Braille verso dots detection.** The experimental result shows the F1 value of Haar method in our first stage has dropped sharply from 0.970 to 0.895, which is influenced by the lower recall rate of 81.37%. But in our framework, the first stage is just used to quickly get some reliable Braille verso dots for next accurate Braille cells grid construction. So we remain the high precision rate of 99.53% with a lower recall rate. Finally, our TS-OBR method still obtains high performance with the F1 value 0.997 for verso dots detection, which is similar as recto dots. The processing time is average 1.58 seconds for verso dots detection with our TS-OBR framework.

While our proposed framework TS-OBR with F1 value of 0.998, these can be reduced sharply to average 4.03 wrong detected dots and 0.34 missing dots for one Braille image. This statistical analysis results show the effectiveness of our framework for real applications.

**Braille characters recognition.** Based on the detection of Braille dots and cells grid, we can easily get the results of Braille characters recognition as Table 5 shows. The F1 values are 0.995 and 0.994 for recto Braille characters and verso Braille characters respectively on the DSBI dataset. The results are little lower than F1 value of Braille dots detection, which lies in the evaluation of six-doted Braille character is stricter than a single Braille dot.

### 4.4.2 Results on our dataset BEP

We use the models trained on the DSBI to directly evaluate all the Braille exam papers of BEP including recto dots detection and the Braille characters recognition. The detailed result is shown as Table 6.

For recto dots detection in the first stage, the method based on Haar+ Adaboost achieves a very high precision rate 99.25% on the BEP. Although the recall rate drops significantly from 95.75% to 90.50%, which means many dots are missing. However, the recall rate can be increased to 99.90% in the second stage. Compared with the results on DSBI, the precision rate on the BEP has a

**Table 5.** Results of Braille characters recognition.

| Method | Dots type | Pre | Rec | F1 |
|---|---|---|---|---|
| **TS-OBR** | recto | 99.06% | 99.99% | 0.995 |
| **TS-OBR** | verso | 99.10% | 99.71% | 0.994 |

**Table 6.** Comparison of results on both datasets.

| Dataset | Method | Type | Pre | Rec | F1 | Time |
|---|---|---|---|---|---|---|
| DSBI | Haar | Recto dot | 98.38% | 95.75% | 0.970 | 0.89s |
| BEP | Haar | Recto dot | 99.25% | 90.50% | 0.947 | 0.95s |
| DSBI | TS-OBR | Recto dot | 99.65% | 99.97% | 0.998 | 1.45s |
| BEP | TS-OBR | Recto dot | 98.46% | 99.90% | 0.992 | 1.54s |
| DSBI | TS-OBR | Recto character | 99.06% | 99.99% | 0.995 | 1.59s |
| BEP | TS-OBR | Recto character | 96.18% | 99.93% | 0.980 | 1.53s |

slight reduce from 99.65% to 98.46%. Braille exam papers are more complex than printed Braille books, for the size and appearance of Braille dots stabbed vary greatly because of the difference of stab habits and strength. And some detected errors are from those erased dots on BEP dataset, which are also difficult for manual detection. Finally, the F1 value on the BEP is 0.992, which demonstrates the good generalization ability of our proposed method and framework.

For Braille character recognition, the precision on the BEP reduced from 99.06% to 96.18% compared with the results on DSBI, which means there are average 12 wrong recognized characters in one Braille exam paper. Compared with Braille dots detection, Braille character recognition is more rigorous. As long as one of the six points in the Braille cell is wrong, the whole Braille character recognition is wrong. Though our framework has gained an excellent performance on Braille dot detection, it still needs more efforts and study for manually stabbed Braille exam answer papers.

Table 6 also shows the whole system TS-OBR only costs about 1.5 second for recto dots detection with high recognition rate on the two datasets.

## 5 CONCLUSION

Double-sided Braille image recognition is challenging for various Braille papers, interference of recto and verso dots, deformation and skewness of Braille images. This paper proposes a novel two-stage learning framework for double-sided Braille image recognition. The experimental results on the public dataset DSBI and our Braille exam answer paper dataset BEP show the effectiveness, good generalization and fast ability for Braille dots detection and Braille characters recognition. Our future work is to further optimize our framework and apply it in real Braille images recognition applications.

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

## References


1. Available: https://github.com/yeluo1994/DSBI
2. Ahonen, T., Hadid, A., Pietikäinen, M.: Face description with local binary patterns: Application to face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**, 2037–2041 (2006)
3. Al-Salman, A., AlOhali, Y., Alkanhal, M.I., AlRajih, A.: An arabic optical braille recognition system (2007)
4. Al-Shamma, S.D., Fathi, S.: Arabic braille recognition and transcription into text and voice. 2010 5th Cairo International Biomedical Engineering Conference pp. 227–231 (2010)
5. Antonacopoulos, A., Bridson, D.: A robust braille recognition system. In: Document Analysis Systems (2004)
6. Babadi, M.Y., Jafari, S.: Novel grid-based optical braille conversion: from scanning to wording (2011)
7. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) **1**, 886–893 vol. 1 (2005)
8. Isayed, S., Tahboub, R.: A review of optical braille recognition. 2015 2nd World Symposium on Web Applications and Networking (WSWAN) pp. 1–6 (2015)
9. Li, J., Yan, X.: Optical braille character recognition with support-vector machine classifier. 2010 International Conference on Computer Application and System Modeling (ICCASM 2010) **12**, V12–219–V12–222 (2010)
10. Li, R., Liu, H., Wan, X., Qiang, Y.Z.: Dsbi: Double-sided braille image dataset and algorithm evaluation for braille dots detection. International Conference on Video and Image Processing (ICVIP) (2018)
11. Mennens, J., Tichelen, L.V., Francois, G., Engelen, J.J.: Optical recognition of braille writing. In: ICDAR (1993)
12. Namba, M., Zhang, Z.: Cellular neural network for associative memory and its application to braille image recognition. The 2006 IEEE International Joint Conference on Neural Network Proceedings pp. 2409–2414 (2006)
13. Organization, H.: Visual impairment and blindness. http://www.who.int/en/news-room/fact-sheets/detail/blindness-and-visual-impairment, accessed 2018
14. Viola, P.A., Jones, M.J.: Rapid object detection using a boosted cascade of simple features. In: CVPR (2001)


View publication stats