*electronics*

MDPI

*Article*

# Low-Computational-Cost Algorithm for Inclination Correction of Independent Handwritten Digits on Microcontrollers

**H. Waruna H. Premachandra [1], Maika Yamada [2], Chinthaka Premachandra [2,*] and Hiroharu Kawanaka [3]**

[1] ICT Center (Information Communication Technology Center), Wayamba University of Sri Lanka, Makandura, Gonawila (NWP) 60170, Sri Lanka; waruna@wyb.ac.lk
[2] Department of Electronic Engineering, School of Engineering, Shibaura Institute of Technology, Tokyo 135-8548, Japan; ag17111@shibaura-it.ac.jp
[3] Department of Electrical and Electronic Engineering, Mie University, Mie 514-8507, Japan; kawanaka@elec.mie-u.ac.jp
* Correspondence: chintaka@sic.shibaura-it.ac.jp

**Abstract:** In recent years, the digitization of documents has progressed, and opportunities for handwritten document creation have decreased. However, handwritten notes are still taken for memorizing data, and automated digitalization is needed in some cases, such as making Excel sheets. When digitizing handwritten notes, manual input is required. Therefore, the automatic recognition and input of characters using a character recognition system is useful. However, if the characters are inclined, the recognition rate will be low. Therefore, we focus on the inclination correction problem of characters. The conventional method corrects the inclination and estimates the character line inclination. However, these methods do not work when characters exist in independent positions. Therefore, in this study, we propose a new method for estimating and correcting the tilt of independent handwritten digits by analyzing a circumscribed rectangle and other digital features. The proposed method is not based on an AI-based learning model or a complicated mathematical model. It is developed following a comparatively simple mathematical calculation that can be implemented on a microcontroller. Based on the results of the experiments using digits written in independent positions, the proposed method can correct the inclination with high accuracy. Furthermore, the proposed algorithm is low-computational cost and can be implemented in real-time on a microcontroller.

**Keywords:** document image processing; character inclination estimation; circumscribed rectangle; digit feature analysis



**Citation:** Premachandra, H.W.H.; Yamada, M.; Premachandra, C.; Kawanaka, H. Low-Computational-Cost Algorithm for Inclination Correction of Independent Handwritten Digits on Microcontrollers. *Electronics* **2022**, *11*, 1073. https://doi.org/10.3390/electronics11071073

Academic Editors: Enrico Vezzetti, Andrea Luigi Guerra, Gabriele Baronio, Domenico Speranza and Luca Ulrich

Received: 9 February 2022
Accepted: 28 March 2022
Published: 29 March 2022



## 1. Introduction

Owing to document digitalization, handwritten document usage is decreasing. However, there are many situations in which handwriting is needed. For example, writing down experimental data in notes during scientific experiments using apparatus is still common. Sometimes, automatically converting memorized data to electronic documents, such as Excel sheets, sheets by OCR platforms (ex: Nanonets-OCR platform), is important because manual insertion is time-consuming. The automatic conversion of written documents to electronic documents does not work well when written data are inclined. This inclination correction problem is solved when the written data exists as character lines. However, they do not work well when characters exist as independent characters.

Many studies have been conducted on document image analysis [1–7], including mathematical expression recognition [8], character recognition, etc. Character tilt correction is related to document image analysis. In conventional research, some methods are used to correct the inclination of character lines, such as the inclination correction of the character string using a wavelet transform and statistical-based methods [8–21]. When it is a character string (line) with two or more characters, the slope of the character string is estimated, and the slope is corrected. However, these methods cannot be applied to

independent character tilt correction because the slope cannot be estimated as a character line. Therefore, in this study, we examined the tilt estimation and correction of handwritten numbers because there are many situations where independent digits are written compared to independent character writing (Figure 1). Furthermore, the conventional methods cannot be implemented on small controllers since the algorithms include comparatively complex computing.

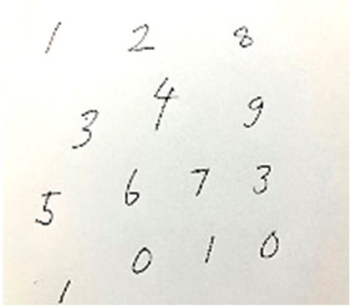

**Figure 1.** Example of a handwritten document.

The purpose of this study is to develop a low-computational cost algorithm to estimate the slope of an independent handwritten number. Furthermore, the algorithm should be implemented on a microcontroller. Focusing on the fact that the number is vertically long, we obtained the minimum external rectangle surrounding the digit structure. Accordingly, the tilt was corrected by rotating the image area of the external rectangle. However, we found that tilt correction is not possible with external rectangles in the case of digit "4". Thus, we perform separated processing to achieve a tilt correction of digit "4" as described below. The number "4" was identified from other numbers, and the tilt was estimated using only the"4" feature without relying on the circumscribed rectangle. Furthermore, to distinguish between the number "4" and other numbers, we pay attention to the unique characteristics of the number "4" and propose a new identification method. In this work, we try to handle a handwritten digit inclination correction problem only using the image data at hand, without using any datasets for training, since the training stage is not included in the proposal. This time the inclination of the digits is measured by calculating its circumscribing rectangle and analyzing other unique features. The overall process comprises trigonometry calculation and the Hough Transform (HT). The HT is comparatively time-consuming. But, in this study, we have reduced the computation cost of HT by reducing the number of pixels for the HT process, and the proposed algorithm can be implemented on microcontrollers.

Based on the experimental results of the proposed method, the inclination of many numbers can be corrected with good accuracy, and the method is very effective for numbers inclined in a certain range. Furthermore, the proposed method has a short processing time, and it can be implemented in real-time with a small computer, such as a Raspberry Pi. Therefore, the proposed method is of an applicable level.

## 2. Tilt Estimation and Correction

### *2.1. Tilt Estimation with the Circumscribed Rectangle*

A sample input image is displayed on the left side of Figure 2. As shown in the figure, some digits are inclined at a certain level. In this study, we first estimated the tilt angle and then performed tilt correction following the angle. The main processing flow for tilt estimation and correction is shown in Figure 3.

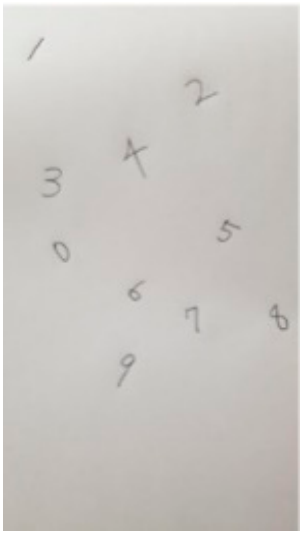

**Figure 2.** Example of a target document.

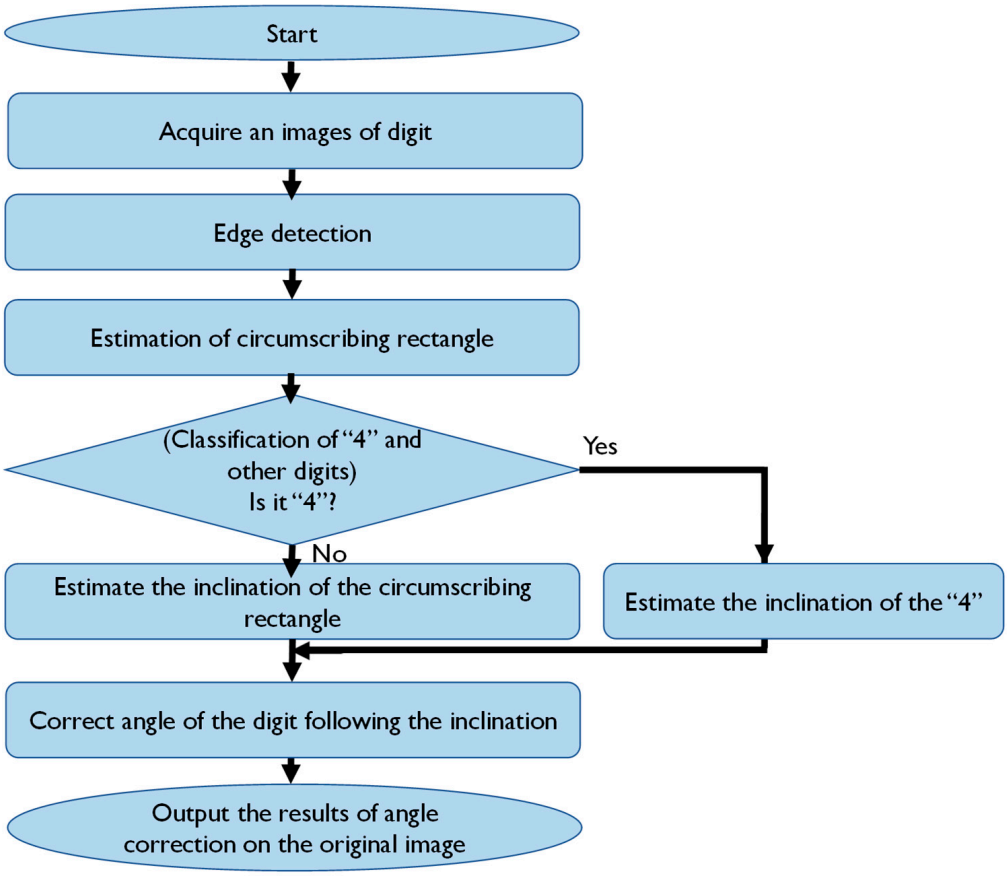


**Figure 3.** Overall processing flow of tilt estimation and correction.

To determine the circumscribed rectangle, the image was first binarized based on discriminant analysis [22]. This time, digits written on a consistent-color paper were used. The binarization following discriminant analysis can separate the digits and background when the digits are written with a colored pen or pencil. Subsequently, the lines related to the digits were further extracted using the Canny edge detector [23–25]. For character line components, they were sandwiched between two close contours (edges). Through this feature, the lines that correspond only to the digits were determined, and objects other

than the digits were removed. Figure 4 shows an example image obtained following these processes. Subsequently, we calculated the slope of the long side of the circumscribed rectangle. In this work, a vertical circumscribed rectangle was not considered, and the minimum circumscribed rectangle surrounding the digit was estimated, as shown in Figure 5.

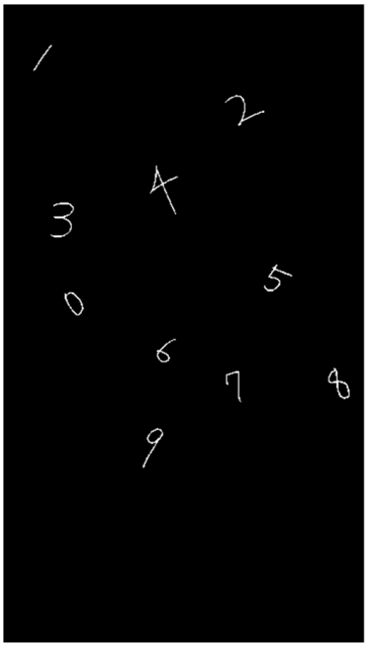

**Figure 4.** Sample image of digit extraction.

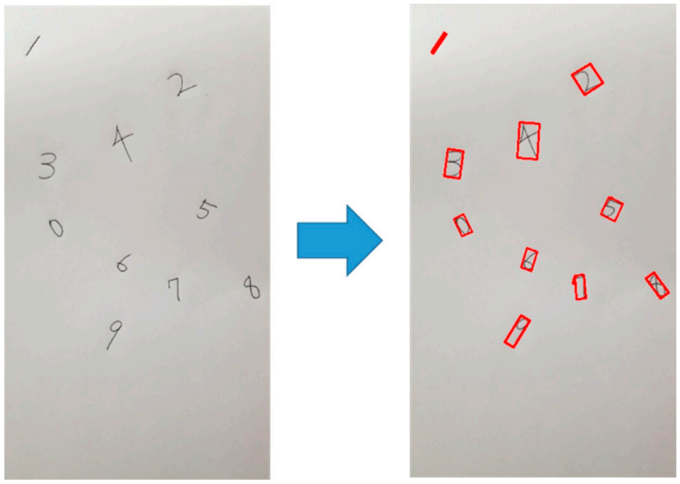

**Figure 5.** Estimation of the minimum circumscribed rectangle.

Figure 6 shows how to find the slope of the long side of the estimated circumscribed rectangle. Because the coordinates of the four corners of the rectangle can be obtained from the circumscribed rectangle, the length was calculated using those coordinates. As shown in Figure 6, with the corner points of the rectangle $(ax, ay)$, $(bx, by)$, the side length of the axis $l$ can be expressed by Equation (1). Consequently, the lengths of the long and short sides of the rectangle are calculated. Then, the long side length is used. In Figure 6, the rectangle $\theta$, which represents the angle of the long side against the horizontal direction, can be expressed by Equation (2). Conversely, $\theta'$, which represents the angle of the long

side against the vertical direction, can be expressed by Equation (3). $\theta'$ becomes the tilt and considered the tilt of the digit.

$$l = \sqrt{(ax - bx)^2 + (ay - by)^2} \tag{1}$$

$$\theta = \cos^{-1}\left(\frac{bx - ax}{l}\right) \tag{2}$$

$$\theta' = 90 - \cos^{-1}\left(\frac{bx - ax}{l}\right) \tag{3}$$

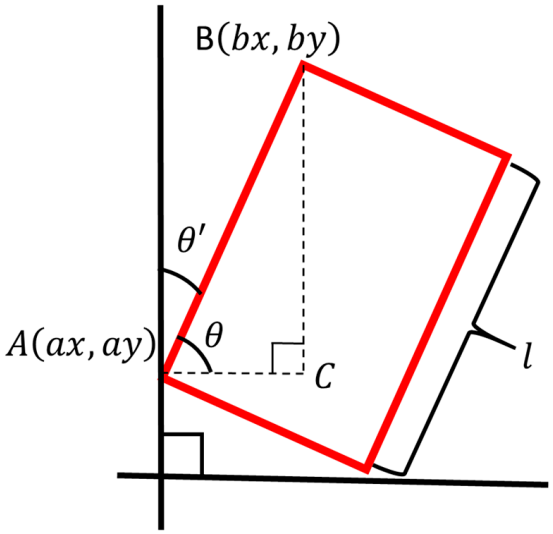


**Figure 6.** Calculation of the side length of the long axis.

### *2.2. Separation of "4" and other Digits*

Figure 7 shows the minimum circumscribed rectangle surrounding the digit "4". As shown in Figure 4, the circumscribed rectangle considering the rotation of "4" looks like a rhombus with respect to the number, and the slope of the circumscribed rectangle does not coincide with the slope of the digit. Therefore, we first separated "4" and other digits. In the case of "4", a special process was performed to estimate the tilt.

**Figure 7.** Circumscribed rectangle of the digit "4".

To separate "4" from the other digits, we focus on the line crossing part "+" of the "4". The status of the perpendicular crossing of two lines is not available in the other digits, and it has a unique feature for "4". Here, we first estimated the circumscribed rectangle and determined its short length ($w$). Then we detect the straight lines within the rectangle. If at least two lines are detected, then we estimated the angles between each two-line pair ($\varnothing_i$). Here, $i$ = 1,2. In addition, the lengths of the detected lines $l_j$ were determined. The line pair that fulfills the conditions in Equations (4) and (5) were extracted as the crossing line pair of "4". Through this idea, we were able to separate "4" from the other digits with a high success rate.

$$80^{\circ} < \varnothing_i < 100^{\circ} \tag{4}$$

$$l_j \geq w \times 0.8 \tag{5}$$

In the above process, line detection was performed following the Hough transform. The Hough transform is used to detect shapes with mathematical forms, such as lines, circles, and ellipses [26–29]. In this study, we used the Hough transform process for line detection. Generally, a straight line in an $(x, y)$ space can be represented by Equation (6). Here, $a$ and $b$ constants. In the Hough transform, if the $(x, y)$ space line is converted to the $(r, \theta)$ space by Equation (7), then the line almost appears as a single dot. As shown in Figure 8, if we calculate all the lines that pass through the single dot in the $(x, y)$ space, they appear as curved lines in the $(r, \theta)$ space, as shown in Figure 9. In the case where the dots are on a straight line in $(x, y)$ space, there is a crossing point for curved lines in the $(r, \theta)$ space (Figure 9). By following these conditions, line detection can be performed [30–32].

$$y = a_0 x + b_0 \tag{6}$$

$$\rho = x_i \cos\theta + y_i \sin\theta \tag{7}$$

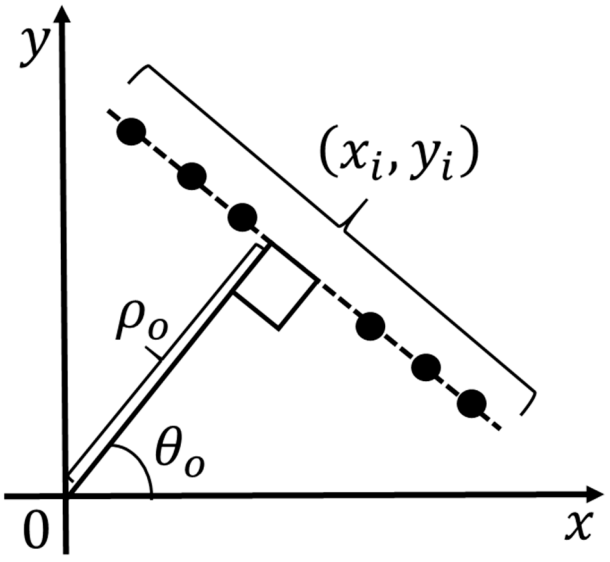


**Figure 8.** Line in the $(x, y)$ space.

The Hough Transform (HT) is a comparatively time-consuming method due to its voting process. However, in this study, we apply HT to the image domain within the estimated circumscribed rectangle. Therefore, the HT process is conducted on a comparatively smaller number of pixels. As a result, the computational time for HT could be reduced.

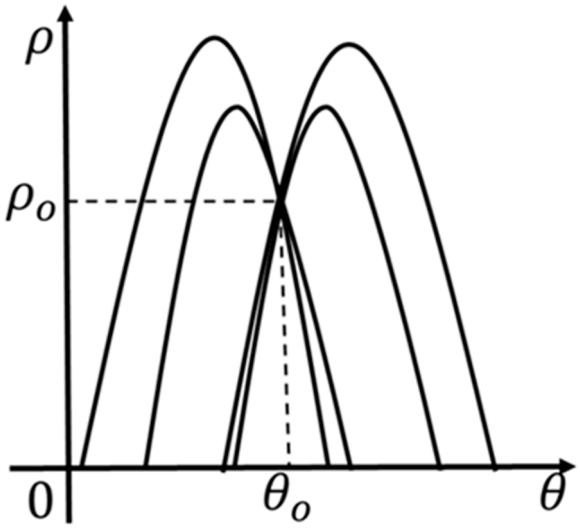


**Figure 9.** $(x, y)$ space to $(r, \theta)$ space conversion.

### *2.3. Tilt Estimation of "4"*

The digit "4" structure includes a diagonal line. We used this diagonal line part to estimate the tilt of "4". As shown in Figure 10, the detected digit "4" is divided into four parts following the perpendicular cross lines. Then, we calculated the angle required for the diagonal line to move to the top-left part (see Figure 10 right). This angle is the tilt of the "4", considering that "4" is inclined.

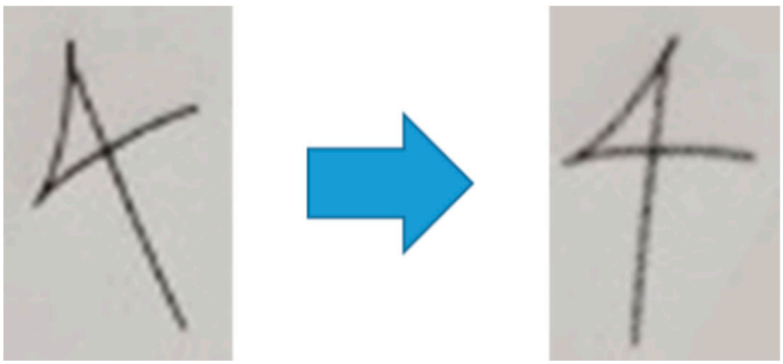

**Figure 10.** Estimation of the diagonal line part of "4".

In this study, we detected diagonal lines by calculating the white pixels in each of the four parts gained following the cross part of a digit "4" image (Figure 11). The part containing the most white pixels includes the diagonal line. The diagonal line was detected using this feature.

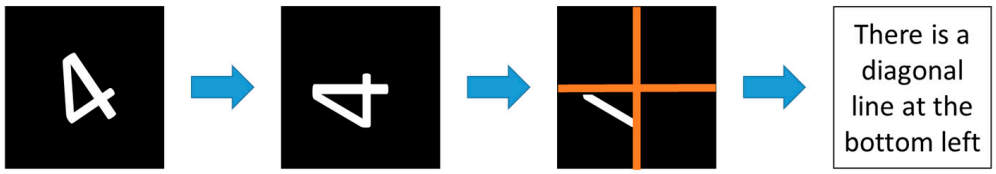


**Figure 11.** Tilt correction example of "4".

## 3. Tilt Correction

In the above sections, we mainly discuss the estimation of the tilt of digits that exist in individual locations. After estimating the tilt, correction must be performed. In this study, tilt correction was performed following an affine transformation. Affine transform is a method that can translate images to different sizes and rotate them in different directions [33–35].

In this study, we applied an affine transformation to correct the digits by rotating their image parts. Following the above equation, if we rotate an image setting using (0, 0) as the image original, the above equation becomes Equation (8), where $\theta$ is the rotation angle.

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{8}$$

Figure 12 shows an example of the tilt correction by the affine transformation. Here, the tilt estimation is performed using the following method.

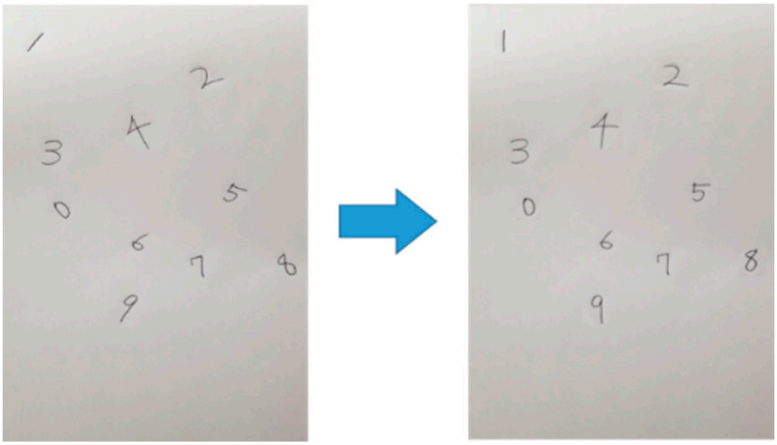

**Figure 12.** Example of the tilt correction of digits written in independent positions.

## 4. Experimental Evaluations

### *4.1. Experimental Environment*

We prepared document images with 300 handwritten digits ranging from 0 to 9. Most of the digits were written in an inclined manner, not vertically. The images were taken while keeping the camera almost 10–15 cm away from the documents. Here, we kept the camera approximately parallel to the documents when the images were captured. The resolution of the image was 480 × 640 pixels. We conducted experiments to confirm the effectiveness of the proposed digit tilt estimation and correction. Considering the application issues, we conducted experiments on a microcontroller (Raspberry Pi 4 hardware). The execution time per image under the above condition is 176 ms. In addition to that, we picked up some relevant image data from famous digit image datasets [36–38] and conducted experiments.

### *4.2. Experimental Results*

The input and output images of numbers 0–9 are shown in Figures 13–22. Table 1 shows the success rate of the angle corrections for each number. Table 2 shows the success rate of the angle corrections for each number when the tilt angle is less than 30°. Table 3 shows the success rate of distinguishing between "4" and other digits.

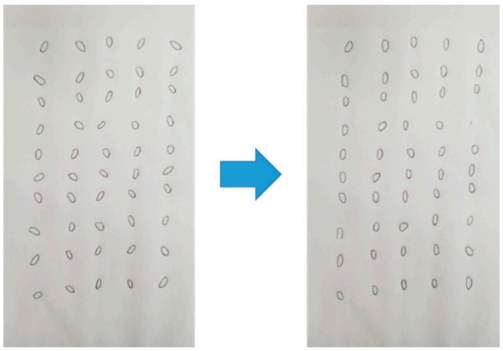

**Figure 13.** Tilt correction examples of "0" following the proposal (**left**: original, **right**: tilt correction results).

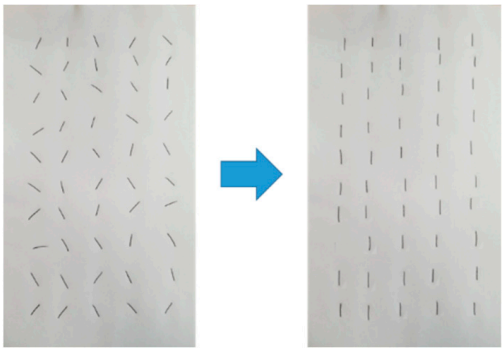

**Figure 14.** Tilt correction examples of "1" following the proposal (**left**: original, **right**: tilt correction results).

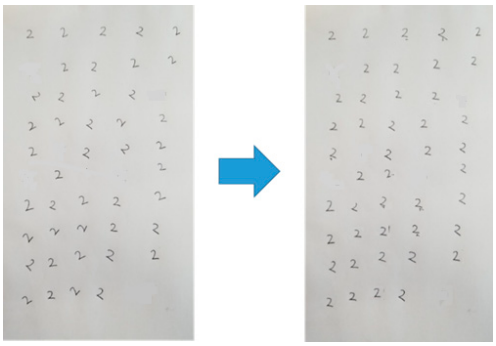

**Figure 15.** Tilt correction examples of "2" following the proposal (**left**: original, **right**: tilt correction results).

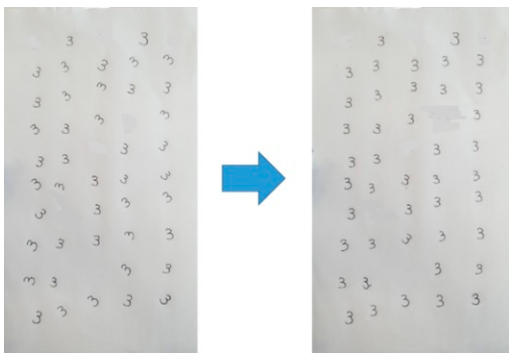

**Figure 16.** Tilt correction examples of "3" following the proposal (**left**: original, **right**: tilt correction results).

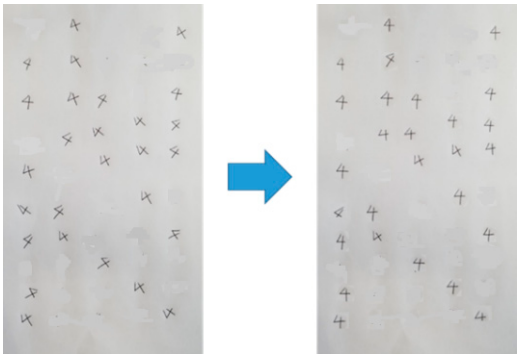

**Figure 17.** Tilt correction examples of "4" following the proposal (**left**: original, **right**: tilt correction results).

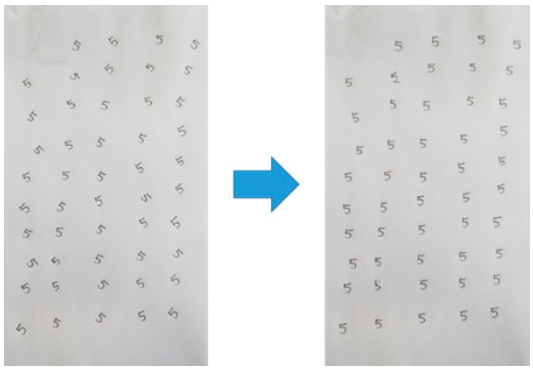

**Figure 18.** Tilt correction examples of "5" following the proposal (**left**: original, **right**: tilt correction results).

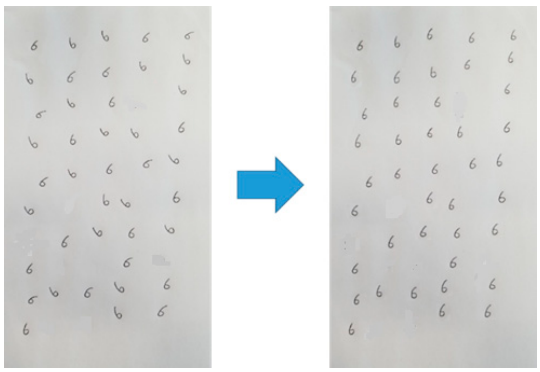

**Figure 19.** Tilt correction examples of "6" following the proposal (**left**: original, **right**: tilt correction results).

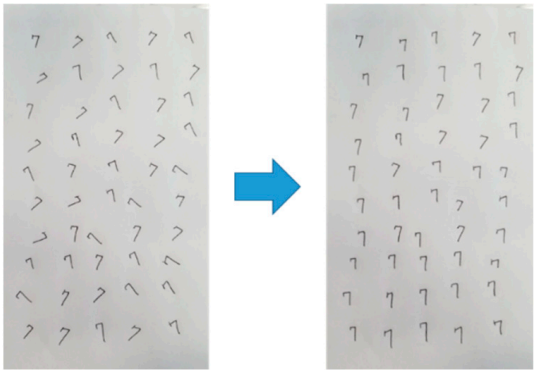

**Figure 20.** Tilt correction examples of "7" following the proposal (**left**: original, **right**: tilt correction results).

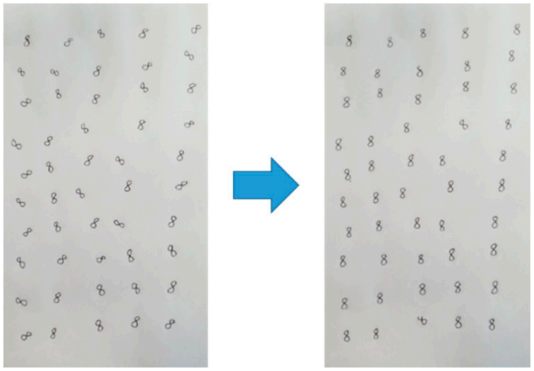

**Figure 21.** Tilt correction examples of "8" following the proposal (**left**: original, **right**: tilt correction results).

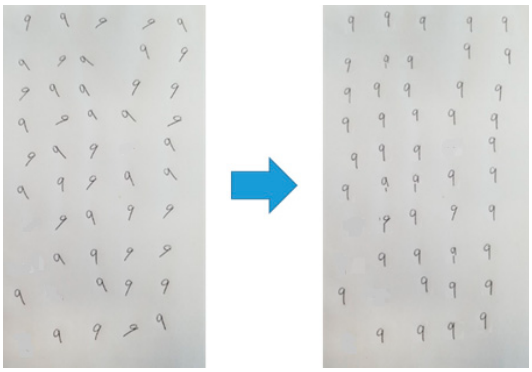

**Figure 22.** Tilt correction examples of "9" following the proposal (**left**: original, **right**: tilt correction results).

**Table 1.** Overall success rate.

| Number | Success Rate of Inclination Correction (%) |
|---|---|
| 0 | 96 |
| 1 | 100 |
| 2 | 92 |
| 3 | 91 |
| 4 | 86 |
| 5 | 93 |
| 6 | 91 |
| 7 | 92 |
| 8 | 95 |
| 9 | 90 |

**Table 2.** Success rate when the inclined angle is within 30°.

| Number | Success Rate of Inclination Correction (%) |
|---|---|
| 0 | 97 |
| 1 | 99 |
| 2 | 94 |
| 3 | 93 |
| 4 | 90 |
| 5 | 93 |
| 6 | 92 |
| 7 | 93 |
| 8 | 97 |
| 9 | 94 |

**Table 3.** Classification success rate of $\lceil 4 \rfloor$ and other numbers.

| Number | Success Rate of Classification (%) |
|---|---|
| 0 | 98 |
| 1 | 100 |
| 2 | 95 |
| 3 | 94 |
| 5 | 97 |
| 6 | 93 |
| 7 | 93 |
| 8 | 90 |
| 9 | 91 |

The angles of the digits other than 4 were successfully corrected with a probability of 90% or more. The cause of the failure to correct the angle was that the circumscribed rectangle considering rotations was not generated to surround the entire number. The reason is that during the binarization process, some pixels belonging to the digit line were detected as background pixels. An example of the failure to generate a circumscribed rectangle is shown in Figure 23. As shown in the figure, if one number is split into several parts and recognized, the processing will fail. The number is split and recognized because the lines of the numbers are thin, and the lines may disappear at the binarization stage. However, if the line is clearly written this error will not occur. Solving this issue by proposing the new binarization approaches or applying the latest methods in the literature is an important future work [39]. In Tables 1 and 2, the slope correction rate is higher regarding many digits when the slope is within approximately 30°. Therefore, this method is considered to be effective when the angle of the number is within 30°.

**Figure 23.** Example of an error in the circumscribed rectangle generation.

Table 4 shows the success rate of the tilt correction. Based on the findings, the success rate is very high when the numbers are neatly binarized.

**Table 4.** Success rate when the digits and background are clearly separated in the binarized process.

| Number | Success Rate of Inclination Correction (%) |
|---|---|
| 0 | 99 |
| 1 | 100 |
| 2 | 97 |
| 3 | 96 |
| 4 | 92 |
| 5 | 95 |
| 6 | 96 |
| 7 | 96 |
| 8 | 98 |
| 9 | 95 |

Future prospects include handling the thin lines of numbers, increasing the accuracy rate of circumscribed rectangle acquisition and "4" recognition processing, and handling arbitrary input images. To increase the accuracy of circumscribed rectangle acquisition, other processes can be added, such as increasing the contrast. In this study, we used an input image with less noise and less distortion. Therefore, to support an arbitrary input image, it is necessary to remove noise and correct image distortion.

## 5. Conclusions

In this study, we propose a method that estimates and corrects the slope of independent numbers using a circumscribed rectangle and other specific digit features. We corrected the slope of the digits when the circumscribed rectangle on the slope could be obtained correctly. In the case of the digit "4", the tilt angle cannot be easily estimated with the circumscribed rectangle, but it could be corrected using the positional relationship between the cross part of "4" and the diagonal line portion. We conducted experiments to confirm the proposed method by preparing appropriate document images with inclined digits. The results revealed that the proposed method achieved a success rate of approximately 90% or more. The circumscribed rectangle was not estimated correctly, mainly due to the disappearance of the digit line parts at the binarization stage. Improving this issue will be one of the main future stages of this work. In addition, the proposed method has a short processing time, and it can be implemented in real-time with a small computer, such as a Raspberry Pi. Therefore, the proposed method is at an applicable level.

**Author Contributions:** Conceptualization, H.W.H.P. and H.K.; methodology, H.W.H.P. and M.Y.; software, H.W.H.P. and M.Y.; investigation, C.P.; resources, H.W.H.P.; data curation, H.W.H.P.; writing—original draft preparation, H.W.H.P.; writing—review and editing, H.K. and C.P.; supervision, H.K.; project administration, C.P.; funding acquisition, C.P. All authors have read and agreed to the published version of the manuscript.

**Funding:** Shibaura Institute of Technology.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Wang, J.-R.; Chuang, Y.-Y. Shadow Removal of Text Document Images by Estimating Local and Global Background Colors. In Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 1534–1538.
2. Brown, M.S.; Tsoi, Y.-C. Geometric and shading correction for images of printed materials using boundary. *IEEE Trans. Image Process.* **2006**, *15*, 1544–1554. [CrossRef] [PubMed]
3. Mtimet, J.; Amiri, H. Document class recognition using a support vector machine approach. In Proceedings of the 2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Monastir, Tunisia, 21–23 March 2016; pp. 161–166.
4. Sreelakshmi, U.K.; Akash, V.G.; Rani, N.S. Detection of variable regions in complex document images. In Proceedings of the 2017 International Conference on Communication and Signal Processing, Melmaruvathur, India, 6–8 April 2017.
5. Garg, R.; Chaudhury, S. Automatic Selection of Parameters for Document Image Enhancement Using Image Quality Assessment. In Proceedings of the 2016 12th IAPR Workshop on Document Analysis Systems, Santorini, Greece, 11–14 April 2016; pp. 422–427.
6. Kieu, V.; Visani, M.; Journet, N.; Mullot, R.; Domenger, J. An efficient parametrization of character degradation model for semi-syntheticimage generation. In Proceedings of the Workshop on Historical Document Imaging and Processing, Washington, DC, USA, 24 August 2013.
7. El-Etriby, S.S.; Amin, K.M. Detection and correction of deformed historical arabic manuscripts. In Proceedings of the International Conference on Computer and Communication Engineering (ICCCE'10), Kuala Lumpur, Malaysia, 22–23 June 2021.
8. Wang, J.; Du, J.; Zhang, J.; Wang, Z.-R. Multi-modal Attention Network for Handwritten Mathematical Expression Recognition. In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, Australia, 20–25 September 2019; pp. 1181–1186.
9. Papandreou, A.; Gatos, B. A Coarse to Fine Skew Estimation Technique for Handwritten Words. In Proceedings of the 2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, 25–28 August 2013; pp. 225–229.
10. Vinciarelli, A.; Luettin, J. A new normalization technique for cursive handwritten words. *Pattern Recognit. Lett.* **2001**, *22*, 1043–1050. [CrossRef]

11. Premachandra, C.; Goto, K.; Tsuruoka, S.; Kawanaka, H.; Takase, H. Speedy Character Line Detection Algorithm using Image Block-Based Histogram Analysis. In Proceedings of the Lecture Notes in Computer Science, Aachen, Germany, 4 July 2015; Volume 9164, pp. 481–488. [CrossRef]
12. Goto, K.; Premachandra, C.; Tsuruoka, S.; Takase, H.; Kawanaka, H. Fast algorithm for character line extraction from handwritten examination papers. In Proceedings of the 2014 Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS), Kitakyushu, Japan, 3–6 December 2014; pp. 1454–1458.
13. de Neto, S.A.F.; Bezerra, B.L.D.; Toselli, A.H.; Lima, E.B. A Handwritten Text Recognition System Based on a Pipeline of Optical and Language Models. In Proceedings of the ACM Symposium on Document Engineering 2020, Virtual, 29 September–1 October 2020; pp. 1–4.
14. Neto, A.; Bezerra, B.; Toselli, A. Towards the Natural Language Processing as Spelling Correction for Offline Handwritten Text Recognition Systems. *Appl. Sci.* **2020**, *10*, 7711. [CrossRef]
15. Neto, A.F.D.S.; Bezerra, B.L.D.; Lima, E.B.; Toselli, A.H. HDSR-Flor: A Robust End-to-End System to Solve the Handwritten Digit String Recognition Problem in Real Complex Scenarios. *IEEE Access* **2020**, *8*, 208543–208553. [CrossRef]
16. Nagai, A. On the Improvement of Recognizing Single-Line Strings of Japanese Historical Cursive. In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, Australia, 20–25 September 2019; pp. 621–628.
17. Kieu, V.-C.; Stutzmann, D.; Vincent, N. Vacuity Measure for Handwritten Character Analysis. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Boston, MA, USA, 9–11 June 2010; pp. 561–566.
18. Hao, Y.; Zhu, B.; Nakagawa, M. Large Improvement in Line-Direction-Free and Character-Orientation-Free On-Line Handwritten Japanese Text Recognition. In Proceedings of the 2014 14th International Conference on Frontiers in Handwriting Recognition, Crete Island, Greece, 1–4 September 2014; pp. 329–334.
19. Campos, V.B.; Gómez, V.R.; Rossi, A.H.T.; Ruiz, E.V. Text Line Extraction Based on Distance Map Features and Dynamic Programming. In Proceedings of the 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), Niagara Falls, NY, USA, 5–8 August 2018; pp. 357–362.
20. Dutta, A.; Garai, A.; Biswas, S.; Das, A.K. Segmentation of text lines using multi-scale CNN from warped printed and handwritten document images. *Int. J. Doc. Anal. Recognit. (IJDAR)* **2021**, *24*, 299–313. [CrossRef]
21. Bonyani, M.; Jahangard, S.; Daneshmand, M. Persian handwritten digit, character and word recognition using deep learning. *Int. J. Doc. Anal. Recognit. (IJDAR)* **2021**, *24*, 133–143. [CrossRef]
22. Otsu, N. A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66. [CrossRef]
23. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *8*, 679–698. [CrossRef] [PubMed]
24. Zhou, P.; Ye, W.; Wang, Q. An Improved Canny Algorithm for Edge Detection. *J. Comput. Inf. Syst.* **2011**, *7*, 1516–1523.
25. Mallat, S.; Zhong, S. Characterization of Signals from Multi scale Edges. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 710–732. [CrossRef]
26. Deng, G.; Wu, Y. Double Lane Line Edge Detection Method Based on Constraint Conditions Hough Transform. In Proceedings of the 2018 17th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), Wuxi, China, 19–23 October 2018; pp. 107–110.
27. Nasseri, M.H.; Moradi, H.; Nasiri, S.; Hosseini, R. Power Line Detection and Tracking Using Hough Transform and Particle Filter. In Proceedings of the 2018 6th RSI International Conference on Robotics and Mechatronics (IcRoM), Tehran, Iran, 23–25 October 2018; pp. 130–134.
28. Premachandra, H.W.H.; Premachandra, C.; Parape, C.D.; Kawanaka, H. Speed-up ellipse enclosing character detection approach for large-size document images by parallel scanning and Hough transform. *Int. J. Mach. Learn. Cybern.* **2017**, *8*, 371–378. [CrossRef]
29. Ishida, Y.; Izuoka, H.; Chinthaka, H.; Premachandra, N.; Kato, K. A study on plane extraction from distance images using 3D Hough transform. In Proceedings of the 6th International Conference on Soft Computing and Intelligent Systems, and the 13th International Symposium on Advanced Intelligence Systems, Kobe, Japan, 20–24 November 2012; pp. 812–816.
30. Premachandra, C.; Gohara, R.; Kato, K. Fast lane boundary recognition by a parallel image processor. In Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 9–12 October 2016.
31. Vladimir, T.; Dongwoon, J.; Kim, D.H. Hough Transform with Kalman Filter on GPU for Real-Time Line Tracking. In Proceedings of the Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, Taichung, Taiwan, 3–5 July 2013; pp. 212–216.
32. Fernandes, L.A.F.; Oliveira, M.M. Real-time line detection through an improved Hough transformvoting scheme. *Pattern Recognit.* **2008**, *41*, 299–314. [CrossRef]
33. Belokurov, V. Implementation of affine transform for image rotation using a HLS language. In Proceedings of the 2018 7th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 10–14 June 2018; pp. 1–4.
34. Sazaki, Y.; Putra, S. Implementation of Affine Transform Method and Advanced Hill Cipher for securing digital images. In Proceedings of the 10th International Conference on Telecommunication Systems Services and Applications (TSSA), Bali, Indonesia, 6–7 October 2016.

35. Ono, S.; Premachandra, C. Generation of Panoramic Images by Two Hemispherical Cameras Independent of Installation Location. *IEEE Consum. Electron. Mag.* **2020**, *11*, 17–25. [CrossRef]
36. Kusetogullari, H.; Yavariabdi, A.; Cheddad, A.; Grahn, H.; Hall, J. ARDIS: A Swedish historical handwritten digit dataset. *Neural Comput. Appl.* **2020**, *32*, 16505–16518. [CrossRef]
37. Kusetogullari, H.; Yavariabdi, A.; Hall, J.; Lavesson, N. DIGITNET: A Deep Handwritten Digit Detection and Recognition Method Using a New Historical Handwritten Digit Dataset. *Big Data Res.* **2021**, *23*, 100182. [CrossRef]
38. Cheddad, A.; Kusetogullari, H.; Hilmkil, A.; Sundin, L.; Yavariabdi, A.; Aouache, M.; Hall, J. SHIBR—The Swedish Historical Birth Records: A semi-annotated dataset. *Neural Comput. Appl.* **2021**, *33*, 15863–15875. [CrossRef]
39. Lins, R.D.; Bernardino, R.B.; Smith, E.B.; Kavallieratou, E. ICDAR 2021 Competition on Time-Quality Document Image Binarization. In *Document Analysis and Recognition—ICDAR 2021. ICDAR 2021. Lecture Notes in Computer Science*; Lladós, J., Lopresti, D., Uchida, S., Eds.; Springer: Cham, Switzerland, 2021. [CrossRef]