Article

Novel Approach to High Accuracy and Efficiency Optical Character Recognizer for Handwritten Digits

Sanjith Sambath *

Posted Date: 27 December 2023

doi: 10.20944/preprints202312.1964.v2

Keywords: Optical Character Recognition; Handwritten Digits; Convolutional Neural Network; Textual Image Classification; Skew Correction; Skeletonization/Thinning; Character Mapping doi:10.20944/preprints202312.1964.v2

Article

Novel Approach to High Accuracy and Efficiency Optical Character Recognizer for Handwritten Digits

Sanjith Sambath

International Academy Central, Rochester Hills, MI; sanjith.sambath07@gmail.com

Abstract: The automated transcription of handwritten characters into a legible output is a multi-faceted process with diverse applications. In this paper, a novel approach to optical character recognition (OCR) for handwritten digits is proposed that, in certain components, exceeds current architectures in terms of accuracy, effectiveness, adjustability, temporal efficiency, and/or computational simplicity. This model succeeds in the adoption and enhancement of deprecated or obsolete algorithms across eight steps of image pre-processing—normalization, grayscaling, thresholding/binarization, noise removal, skew-correction, skeletonization/thinning, line separation, and character segmentation. The aforementioned model is evaluated with the use of a Convolutional Neural Network (CNN) leveraging the Balanced eMNIST dataset for training and testing. By suggesting contour-based feature extraction methods as alternatives to pixel-by-pixel iteration, the proposed paradigm demonstrates its capacity to serve as a suitable alternative to current, commonly used algorithms and computational techniques for textual image classification.

Keywords: optical character recognition; handwritten digits; Convolutional Neural Network; textual image classification; skew correction; skeletonization/thinning; character mapping

1. Introduction

Optical Character Recognition (OCR) systems represent specialized classifiers designed to engage in the localization, extraction, and classification of textual content from static and dynamic mediums. The primary objective of such a system is the seamless harvesting of a visual frame containing elements of written or typed text into an output that is amenable to interpretation by an automated system. [10] The real-world applications for OCR classifiers are wide-ranging, with implications in a plethora of fields and industries. Examples of these encompass, but are not limited to: the automatic translation and transcription of extensive texts and works for libraries and similar institutions; the tracking of individual participants at high-octane sporting events; the identification of license plates affixed to vehicles in urban settings; the facilitation of the automated processing of high volumes of documents and papers for use in financial or legal settings; and so forth.

However, it is crucial to underscore the necessity that a distinction be made of the two primary classes of OCR categorization systems. Although the gap between computer-generated and human-originated text may not appear substantial, it is, in reality, a more nuanced demarcation than initial impressions might suggest. Natural inscriptions differ fundamentally from their synthetic counterparts in the fact that they lack one paramount characteristic: uniformity. [10] With the lack of such invariability, assumptions cannot be made for an automated OCR program based on the spacing between individual characters, the thickness of different features of specific digits, the slants and skews intrinsic to each letter or number, or more. Therefore, in the pursuit of classifiable normalized text, OCR systems tailored for handwritten script rely heavily on a balance of pre-processing techniques to appropriately manipulate the provided data. [2]

In the realm of practical application, there is a rarity of ideal, pristine data that would be optimal for OCR classification in a real-time categorization scenario. As such, for OCR systems to find any success in the classification of substandard data, combinations of techniques in noise reduction and image smoothing have to also be performed. While established methodologies, including Gaussian, Median, and Bilateral blurring, alongside masking with varied dimensions [3] [8], have already been devised for the task of isolating regions of interest from within a busy image, the quest for an optimal amalgamation persists. Current implementations and combinations of these methods prove inaccurate, computationally intensive, or time-consuming, highlighting the need for further investigation to identify an ideal blend that balances efficiency and precision.

Given the array of pre-processing steps and transformations requisite prior to the actual classification stage, it is imperative to then adapt the character classifying neural network itself to fit the data. Upon the successful completion of the various preprocessing steps in an OCR system, the resulting data slated for prediction will manifest as a one-pixel-thin outline of a handwritten digit or character with a varied image size. [16] However, the challenge lies in the misalignment between this thinned, normalized data and the dominant format of the vast majority of datasets used to train CNNs. For instance, the widely employed eMNIST dataset [5], with thousands of training and testing data images for OCR applications, features 28x28 images of handwritten characters spanning multiple pixels in width. Attempting to classify the thinned, normalized data against a dataset of fundamentally differing characteristics can yield neural classifications that inadequately model the intricacies of the data.

All in all, the preceding paragraphs delineate a number of possible optimizations and target areas for OCR systems that the novel proposed model will aim to refine and improve upon in order to build a more simple and effective paradigm in comparison to current best practices.

2. Related Work

When investigating the possible optimizations for an OCR classification system, a myriad of algorithms and processes emerged from prior research across various pre-processing and neural model steps. These findings were integral to the formulation of our novel approach.

2.1. Skeletonization/Thinning

Skeletonization/Thinning (S/T) is one of the most complex and critical pre-processing steps in the formulation of an OCR system. Aimed at retaining the structural features of textual images while reducing the amount of information inherent in the image, if performed correctly, an effective S/T algorithm will compress a textual image to a one-pixel representation that still preserves its recognizability. Iterative methodologies, posited by Melhi et al. as "involving the deletion of successive layers of pixels on the boundary of the pattern while checking continuity until no more pixels can be removed," encompass the majority of current-day S/T algorithms, with some of the most effective including the Zhang-Suen and Guo-Hall models. [9] In the former, which is widely recognized as the most prevalent form of S/T algorithm, a 3x3 kernel is employed with a two-pass iteration method across the entire binary image, in which the number of non-zero neighbors of each pixel, $B(P_1) = \sum_{i=2}^{9} P_i$, serves as the key metric for contour detection. [18] For the latter, the approach assumes that any thinned object can be deconstructed into a sum of its medial curves, defining a set of curve-like pixels to be so if the majority of its pixels have exactly two 8-neighbors in G, with a few pixels being end-points or branch points. The method also utilizes a modified version of the 3x3 kernel two-pass algorithm introduced in Zhang-Suen. [18] Both S/T methodologies prove notably effective in stripping handwritten text; however, typical of iterative processes, they exhibit shortcomings in terms of temporal and computational complexity and, in some cases, retain artifacts and irregularities that diminish accuracy. [14] Figure 1 illustrates the implementation of both S/T algorithms.

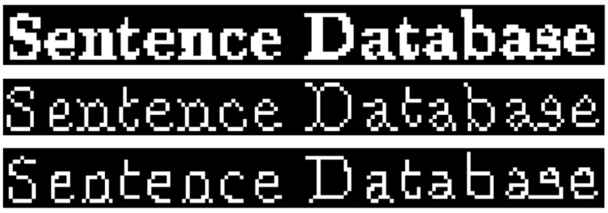


Figure 1. S/T Algorithm Illustration - Input Text (top); Zhang-Suen (middle); Guo-Hall (bottom) - Originally from [7]

2.2. Noise Removal

Across the OCR models we considered, Noise-Removal (NR) techniques also varied significantly. In one framework, the K-Algorithm NR technique [4] operated a modified Median blurring algorithm, $M\left((i-x, j-x)\ldots(i+x, j+x)\right)$, where the process was constrained by the variable 'K,' representing the count of the lowest intensity pixels in the current neighborhood matrix of the pixel under consideration. This method reduced unnecessary image desharpening, thereby diminishing the likelihood of misclassification between noise and foreground pixels. In a differing study [11], NR was performed with a Boolean-algebra-based Coordinate Logic Filter (CLF), represented by $Erosion(I) = \sum_{p=0}^{n-1} 2^p \; Erosion(S_p(i,j)); i = 1,2,...M|j = 1,2,...N$ where $Erosion(S_p(i,j))$ was the erosion of matrix elements in image "I". The main characteristics of CLFs are the direct execution of logic operations between pixel values, which ensured speed and simplicity with this method.

2.3. CNNs

Four existing, proven-effective CNN models appeared as a result of our investigations into the contexts in which other state-of-the-art OCR systems were built: AlexNet, VGG-16, ResNet-18, and DenseNet-121. [15] Each of these networks was uniquely built with certain inherent merits and characteristics. For example, AlexNet's architecture stood out with novel implementations of Rectified Linear Unit (ReLU) functionality and local response normalization layers, contributing to the enhancement of model performance by normalizing the responses of neighboring neurons. Conversely, VGG-16 implemented a swap from the large kernel filters of AlexNet to smaller 3x3 convolution layers, thereby increasing layer depth. When evaluated in an OCR setting, the models demonstrated validation accuracies of 97.50%, 98.16%, 97.83%, and 97.91%, respectively.

From these explorations, we were able to glean a number of insights into various efficient and standardized methods for pre-processing and classification in an OCR system. Additionally, we identified several avenues for improvement and enhancement to further build upon.

3. Proposed System Architecture

The comprehensive process-timeline for the architecture of our proposed OCR software is portrayed in Figure 2.

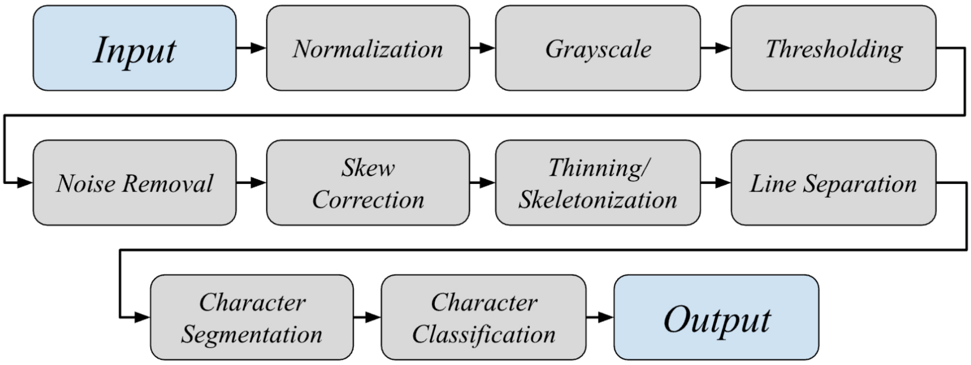


Figure 2. Proposed process for the novel OCR system

doi:10.20944/preprints202312.1964.v2

3.1. Normalization

The model will utilize the default normalized range from OpenCV of 0 to 255, as is inherently integrated into the library itself, because there is no real benefit in manipulating this value. Nevertheless, the mathematical representation of the normalization function is as follows (for n-dimensional image "I") : $I_N = (I - Min) \cdot \frac{newMax-newMin}{Max-Min} + newMin$

3.2. Grayscale

To convert the normalized image to grayscale, this model employs the Luma formula for weighted color conversion, which takes into account the varying perceived luminance of red (R), green (G), and blue (B) components. This formula can be represented as such: $Y = R(0.229) + G(0.587) + B(0.114)$ [1]

3.3. Thresholding

After this, the model transforms the current grayscale image, represented by pixels of varying shades of black, into a binary image with a single intensity threshold—white representing the foreground and black for the background. To accomplish this conversion, the proposed model will leverage the Otsu thresholding algorithm [12], a one-dimensional discrete analogue of Fisher's Discriminant Analysis. In Otsu's method, a pixel-intensity-based histogram is generated, and the probabilities of every intensity level are computed. The algorithm then iteratively explores all possible thresholds to identify the level that maximizes inter-class variance, effectively binarizing the image and optimally segregating the foreground from the background. Although slightly more computationally taxing when taken into conjecture with other methods, such as Adaptive Mean or Adaptive Gaussian Thresholding, Otsu consistently outperforms them in terms of accuracy and effectiveness. Given the marginal computational differences among the options and their considerable variations in efficacy, the latter aspect was prioritized, and Otsu was implemented.

3.4. Noise Removal

The NR step is consistently employed in this model, integrated as a substep into various preprocessing stages. While options for a blurring-based NR solution were available, the proposed framework implements a custom, adjustable algorithm that recognizes connected components above a certain bin threshold to be masked out of the image. Connected pixels are defined as pixels with 8-way connectivity with their adjacents.

3.5. Skew Correction

Skew correction algorithms in earlier OCR models utilized functions with now altered base functionalities, rendering them obsolete in the present context. As such, a new methodology is proposed. To begin, the novel character classification framework systematically populates white foreground pixels from the initial to the terminal pixel in every row of the image. A comparable operation is then conducted along the vertical axis, extending the method to columns. From there, the Suzuki and Abe contour finding algorithm can be parsed [17], which yields a list of contours outlining the edges of the text in the image. The retrieval method in said function is set to $RETR\_TREE$, ensuring the return of a full hierarchy of all contours, and the approximation method is $CHAIN\_APPROX\_SIMPLE$, so as to compress horizontal, diagonal, and vertical segments wherever possible. With this analysis, a bounding box is then constructed, encapsulating the entirety of the contoured image. The numerical ordering of the resulting box vertices' coordinates varies depending on whether the image is skewed clockwise or counter-clockwise, as depicted in Figure 3.

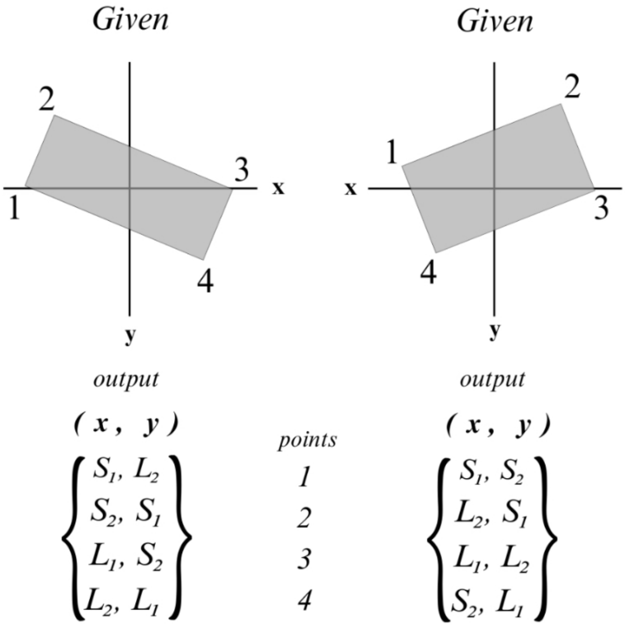


Figure 3. Bounding box coordinates ordering based on skew. S_1 is smallest, S_2 is second smallest, L_1 is largest, L_2 is second largest

As the figure suggests, a distinction can be made between a textual image that is skewed clockwise or counter-clockwise based on the arithmetic sequencing of its respective bounding box vertices. In light of this rationale, the following mathematical formula is performed to determine the deviation from 0° for any textual image, with the variable "P_n" denoting the point number, as indicated in Figure 3. For clockwise skew: $rad(\sin^{-1}((P_2y - P_3y)/(P_2x - P_3x)))$; for counter-clockwise skew: $rad(\sin^{-1}((P_1y - P_2y)/(P_1x - P_2x)))$. Ultimately, a rotation matrix is formulated based on this slope, and the original image undergoes a warping transformation to align with this matrix. Figure 4 shows all the consecutive stages of this skew correction function.

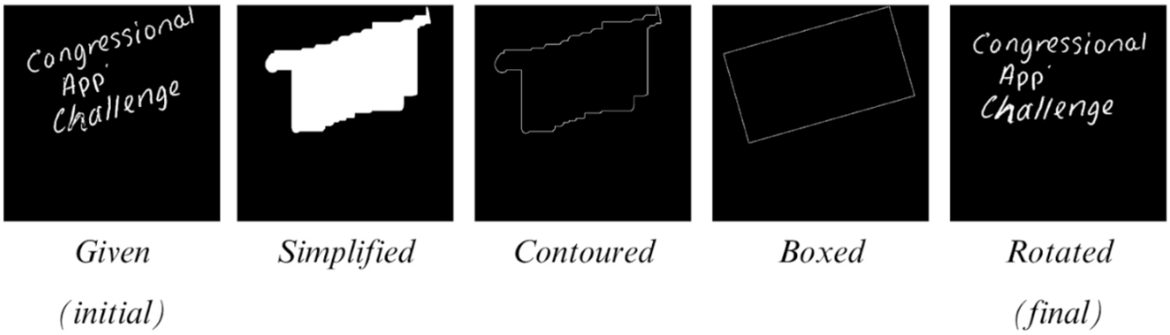


Figure 4. Skew correction algorithm sequentially shown

3.6. Skeletonization/Thinning

In the novelly proposed OCR model, an alternative S/T algorithm—distinct from the more commonly used methods [18] [6] as mentioned earlier—is implemented with the aim of ensuring uniformity across samples, a critical consideration in the context of handwritten text, where stroke thickness and line weight exhibit constant variability. This computation commences with the instantiation of a morphological 3x3 cross kernel used to transform the image. Using this base kernel, the image undergoes successive erosion and dilation operations in relation to the base kernel, compared against itself. The loop terminates when the eroded image becomes empty and is filled with zero pixels. Following this procedure, a number of sharpening and noise-reduction processes are implemented, listed chronologically as follows:

1. Image Dilation; uniform 3x3 kernel, 1 iteration
2. Median Blur; k-size of 5
3. NR function; bin size of 30
4. Image Erosion; uniform 2x2 kernel, 2 iterations
5. NR function; bin size of 8

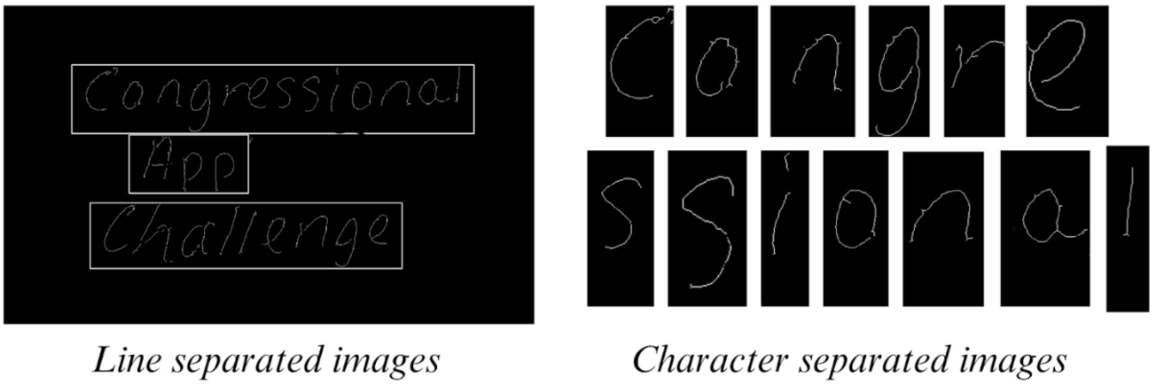
The entire S/T algorithm is then reiterated for a second iteration, and the resulting image constitutes the desired output.

3.7. Line Separation

To horizontally separate the textual elements in the image, the proposed OCR framework applies a nonuniform kernel dilation at a ratio of approximately 1:8.5 with a single morphological iteration. Then, as in the previous steps, the Suzuki and Abe contour identification operation [17] is executed (with the same parameters), followed by a subsequent series of bounding box constructions. To ensure the avoidance of overclassification, the bounding boxes are superimposed on each other, and a procedure is formulated to calculate the overlapping surface areas of the lines. The determination of whether the boxes are reclassified is dictated by the formula: $\left(\frac{intersection\_area}{box1\_area} \geq 0.05\right) \vee \left(\frac{intersection\_area}{box2\_area} \geq 0.05\right)$. In this context, "$intersection\_area$" pertains to the cumulative measurement of the x-overlap and y-overlap between bounding boxes, and the constant value, specifically set at 0.05, is deliberately done to serve the purpose of averting over-analysis while concurrently safeguarding the integrity of genuinely isolated boxes. When this program is run, should the outcome of the condition be affirmative, only the bounding boxes with the greater areas will be preserved.

3.8. Character Separation

Upon segmenting the different lines of the image, the subsequent task consists of isolating each individual character within the image. To achieve this, a straightforward calculation is executed to identify all empty columns in the line images derived from the preceding function. These empty columns are then grouped into consecutive pairs, and the average middle empty column within each group is selected. The average middle column serves as the delineation point for character segmentation. Figure 5 is an illustration of the product of both the line and character separation functions.



Line separated images

Character separated images

Figure 5. Line and character separation in test image

3.9. Character Classification - CNN

Upon the culmination of all the pre-processing steps prior to character classification, the novel OCR proposed in this paper employs a custom CNN design to associate individual handwritten digit images with their corresponding alphabetical characters. The dataset used for training is the eMNIST-Balanced dataset, containing a miscellaneous range of 131,600 training images with 47 balanced classes (consisting of lowercase, uppercase, and numerical digits).

The eMNIST images, initially stored as flattened arrays in the dataset, are reshaped and rotated to a standardized size of 28x28 pixels. The images are then normalized to a pixel intensity range of [0, 1]. To address the imbalanced nature of the dataset, we employ categorical encoding and stratified sampling during the training and validation set split. To ensure consistency in characteristics between the input data from our OCR preprocessing and the training data for the CNN, the eMNIST images are upscaled to a 1:10 version of their original size and then subjected to the same S/T algorithm employed in our OCR system. They are then scaled back down to the standard 28x28 dimensions. This procedure is implemented to maintain uniformity and facilitate accurate representation within the CNN.

The architecture of our model commences with the utilization of the Keras Sequential API for systematic layer-by-layer model construction. The primary layer in the novel OCR classifier is the 2D Convolutional layer that plays a pivotal role in feature extraction, utilizing 32 filters with a kernel size of (5,5) and a rectified linear unit (ReLU) activation function. Convolution layers play a crucial role in extracting features and recognizing spatial patterns within images, which are essential for identifying handwritten characters. The ReLU activation function enhances non-linearity, allowing the network to learn complex representations. The use of 'same' padding ensures that the spatial dimensions of the feature maps remain consistent, preserving valuable structural information at the image borders. Following this inceptive layer, a Max-Pooling with a stride of 2 is employed to downsample the feature maps, reducing computational load and fostering translation invariance. A second Convolutional layer follows, featuring 48 filters with a similar (5,5) kernel size and employing 'valid' padding to extract features without zero-padding the input. This decision is motivated by the intention to capture more central and high-level features as the network progresses. Again, the ReLU activation function is employed to introduce non-linearity, while also helping to avoid the vanishing gradient problem. Backing the convolutional layers, the network employs flattening to transform the 3D tensor into a 1D vector. This flattened representation is then fed into densely connected layers, beginning with a dense layer of 256 neurons, again choosing the ReLU activation function. A subsequent dense layer with 84 neurons further refines the learned features before the final output layer, tailored to the number of classes, employs softmax activation for multi-class classification. A visual representation of this architecture can be seen in Figure 6.

Our training setup involved selecting a set of diverse hyperparameters to enhance CNN performance. At the beginning, we opted for a batch size of 32, along with a training duration of 7 epochs. We chose categorical cross-entropy as our loss function, evaluating the difference between the labeled and predicted values generated by our output layer, which operated between two probability distributions using "softmax" activation. To optimize the model, we utilized the Adaptive Momentum Estimation (ADAM) algorithm. The incorporation of momentum facilitated the consistent movement of gradients in the same direction during optimization, leading to quicker convergence. Additionally, the use of adaptive learning rates ensured responsiveness to the magnitudes of weight parameters, addressing the challenge of smaller gradients. A validation split of 0.1 was used in the preparation of this network.

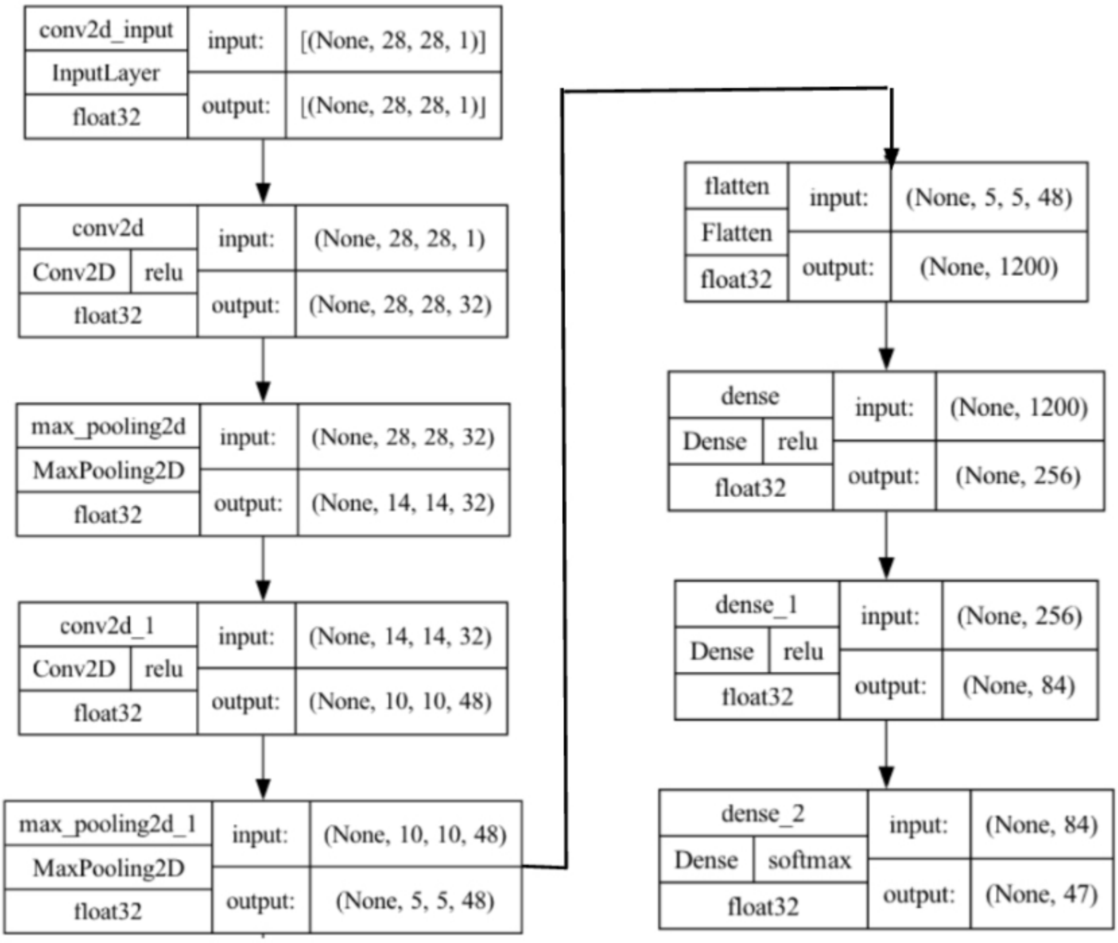


Figure 6. Novel OCR CNN layer-by-layer

4. Results

In assessing the effectiveness of the proposed novel OCR, various elements of the model discussed earlier are compared against other commonly used methods to demonstrate improvements across diverse metrics. The following sections highlight the components that exhibit significant enhancements, while other aspects of the OCR not shown demonstrate performance equivalent to current-day techniques.

4.1. Noise Removal

As the table and figure help depict, the novel implementation of NR processes succeeds in finding a more optimized amalgamation of accuracy and speed. For the most widely used NR methodology, $fastNLMeansDenoising$, which is also the native OpenCV noise-removal function, our proposed technique is over 3341% faster and eliminates 7509 more noise-related pixels. For the four blurring techniques (Basic, Gaussian, and Median, Bilateral Filter), our model exhibits a marginal increase in processing time, only up to 100%, but removes significantly more noise.

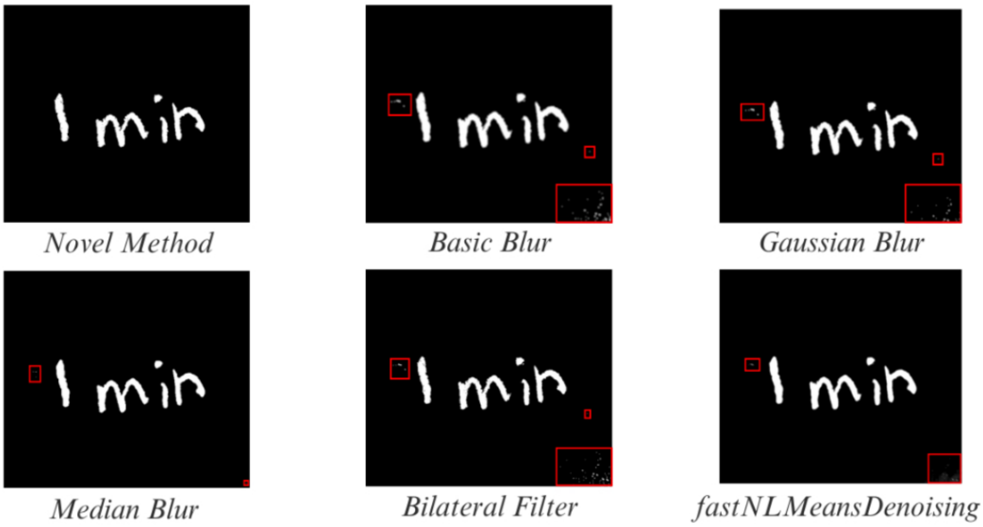
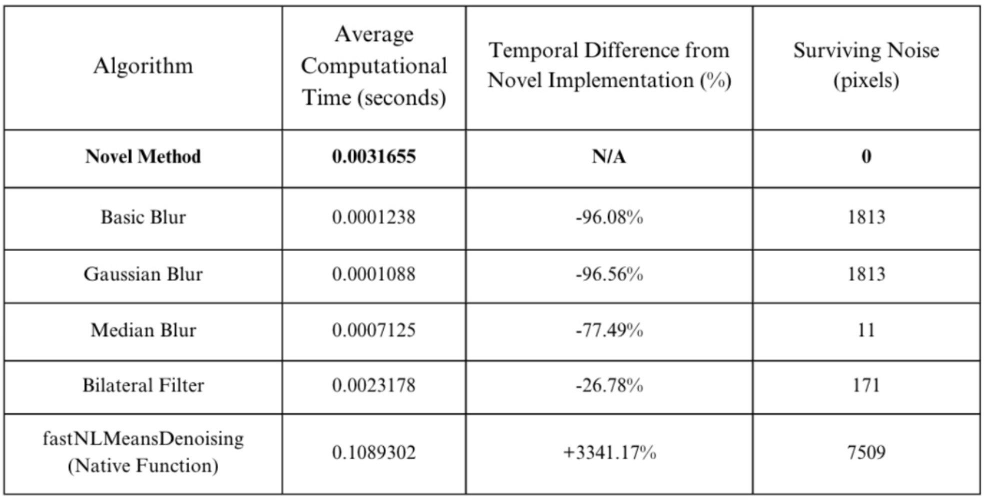


Figure 7. Implementations of varied Noise-Reduction techniques with varying effectiveness/accuracy's. Noise is localized in red.

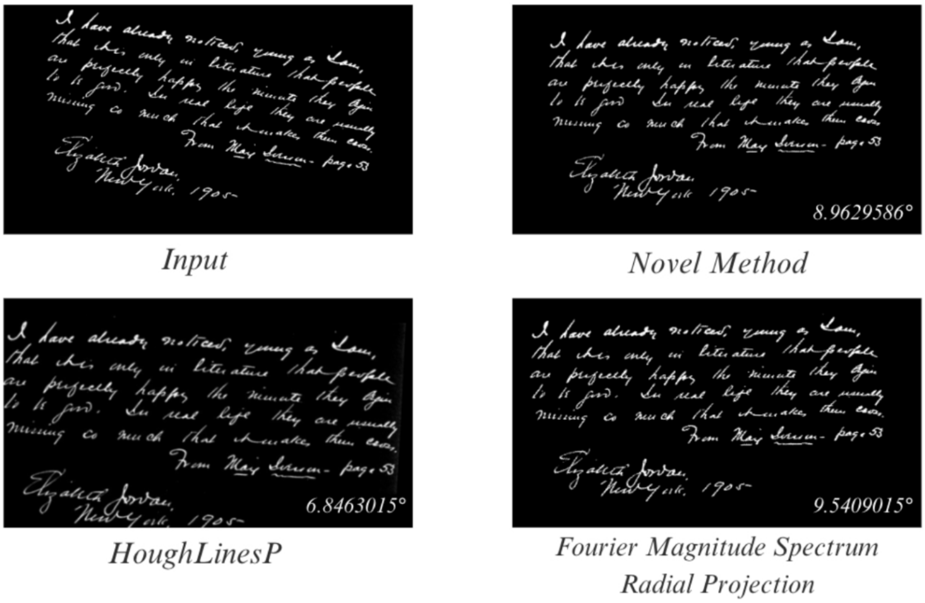


Algorithm	Average Computational Time (seconds)	Temporal Difference from Novel Implementation (%)	Surviving Noise (pixels)
Novel Method	**0.0031655**	**N/A**	**0**
Basic Blur	0.0001238	-96.08%	1813
Gaussian Blur	0.0001088	-96.56%	1813
Median Blur	0.0007125	-77.49%	11
Bilateral Filter	0.0023178	-26.78%	171
fastNLMeansDenoising (Native Function)	0.1089302	+3341.17%	7509

Figure 8. Average Computation Time (seconds), Temporal Difference from Novel Implementation (%), and Surviving Noise (pixels) of different NR algorithms for OCR use cases.

4.2. Skew Correction

The proposed Skew Correction technique adapts the obsolete *minAreaRect* function and operates over 621.29% faster than the ARPFMS skew algorithm. [13] Additionally, it performs only marginally slower than the HoughLinesP function while significantly enhancing accuracy.



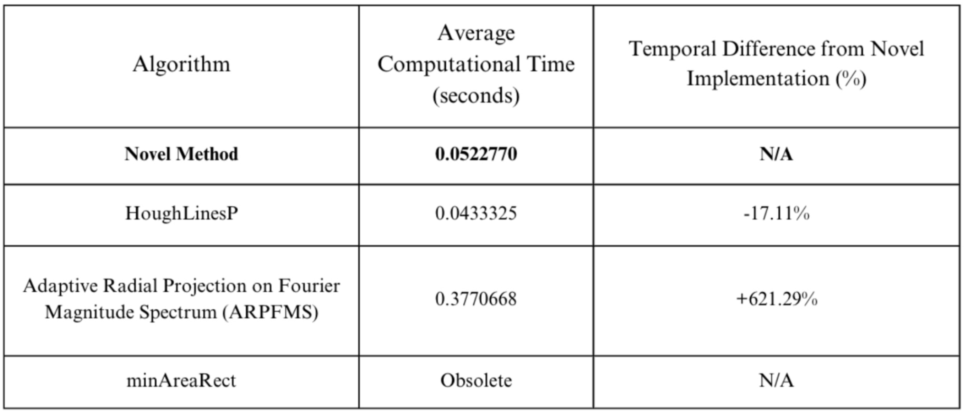
Input

Novel Method

HoughLinesP

Fourier Magnitude Spectrum Radial Projection

Figure 9. Implementations of different Skew Correction techniques with varying accuracies. The measurement for the number of degrees rotated is depicted at the bottom right.



Algorithm	Average Computational Time (seconds)	Temporal Difference from Novel Implementation (%)
Novel Method	**0.0522770**	**N/A**
HoughLinesP	0.0433325	-17.11%
Adaptive Radial Projection on Fourier Magnitude Spectrum (ARPFMS)	0.3770668	+621.29%
minAreaRect	Obsolete	N/A

Figure 10. Average Computation Time (seconds) and Temporal Difference from Novel Implementation (%) of different Skew Correction algorithms for OCR use cases.

4.3. Skeletonization/Thinning

In Figure 11, the presented S/T algorithm is juxtaposed against contemporary best practices. The novel method is found to have minimal anomalies/irregularities when compared to Zhang-Suen and is considerably more accurate than Guo-Hall. On average, the new implementation is about 67437% and 40803% faster than the Zhang-Suen Modified and Educated algorithms, respectively. Although it is estimated to be 4398% slower than Guo-Hall, because of its immense improvements in accuracy, our algorithm remains desirable.

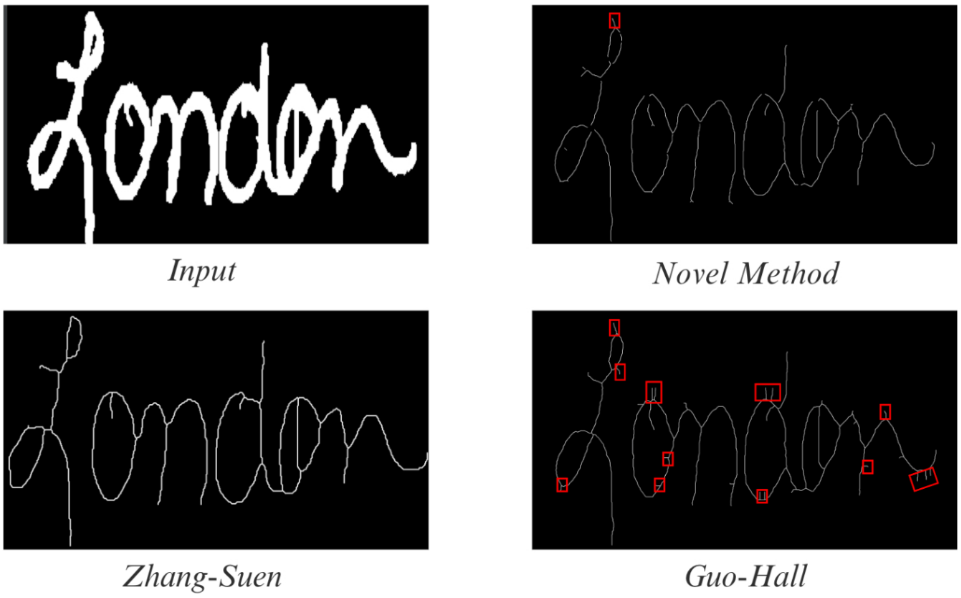
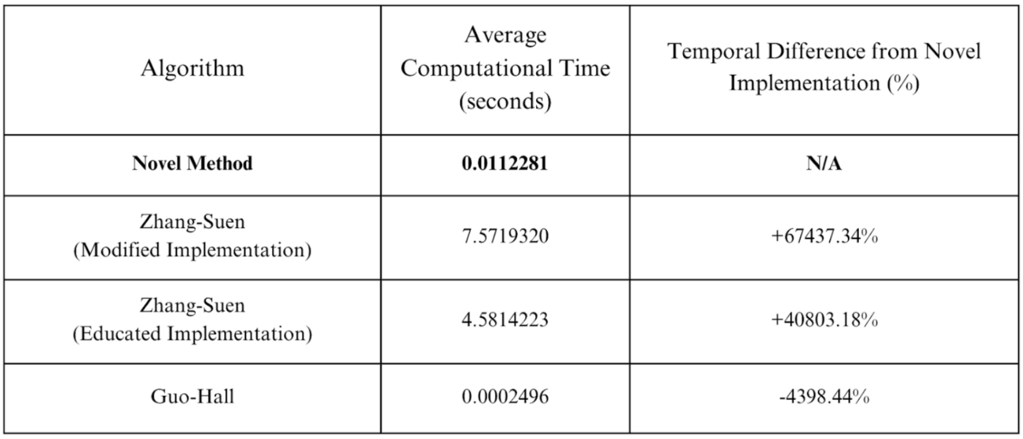


Figure 11. Different Skeletonization/Thinning Algorithms implemented. Anomalies/irregularities are localized in red.



Algorithm	Average Computational Time (seconds)	Temporal Difference from Novel Implementation (%)
Novel Method	**0.0112281**	**N/A**
Zhang-Suen (Modified Implementation)	7.5719320	+67437.34%
Zhang-Suen (Educated Implementation)	4.5814223	+40803.18%
Guo-Hall	0.0002496	-4398.44%

Figure 12. Average Computation Time (seconds) and Temporal Difference from Novel Implementation (%) of different S/T algorithms for OCR use cases.

4.4. CNN

In comparison to other pre-existing proven CNN architectures (AlexNet, VGG16, and ResNet18), our model does not significantly differ in terms of validation accuracy, with only slight improvements. However, a small decline is discerned in test accuracy, reaching 91.78%. But for the specific application at hand, this level of accuracy remains acceptable.

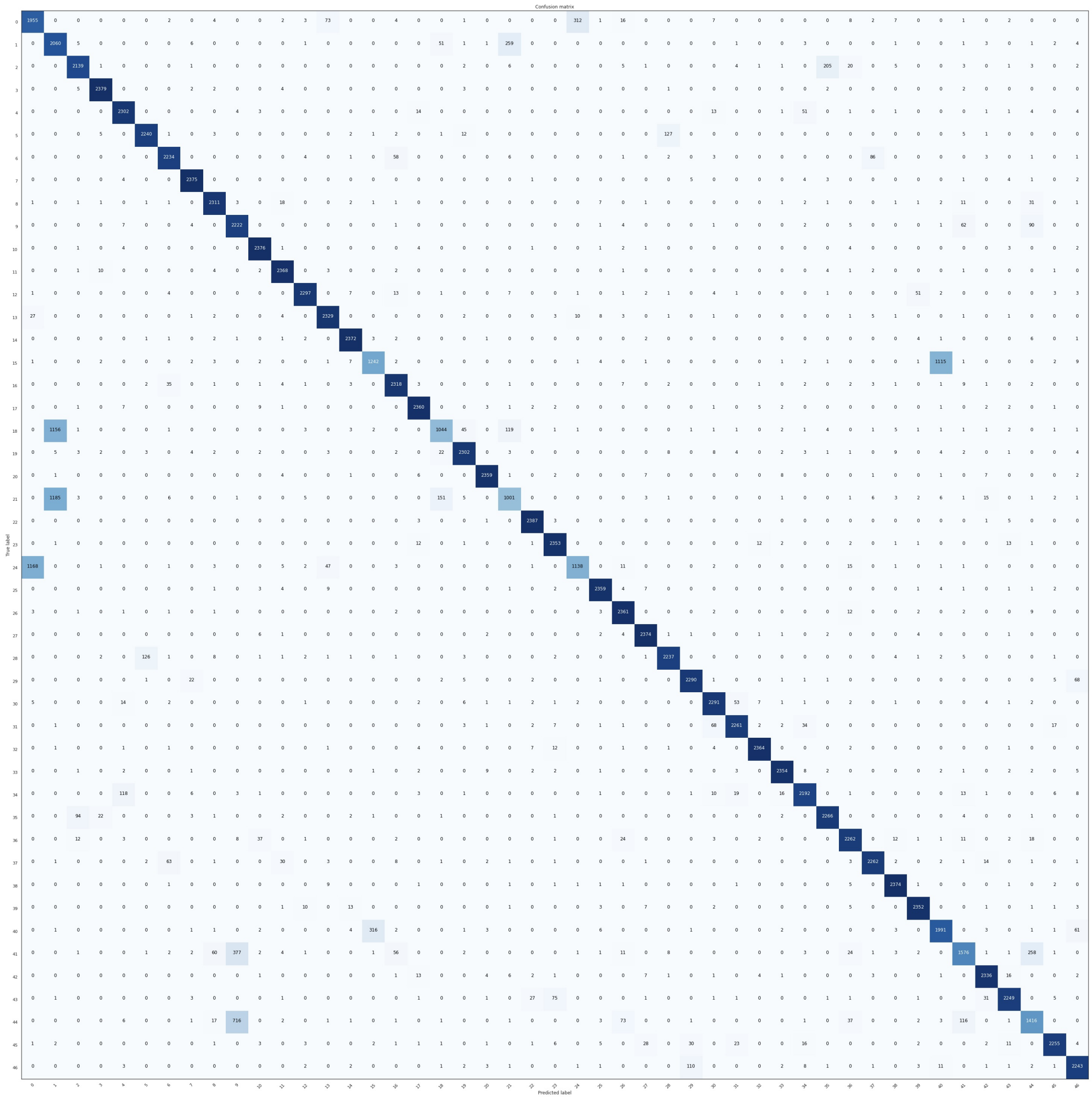


Figure 13. Confusion matrix for CNN – numeric labels correspond to the respective eMNIST digits

Model	Validation Accuracy	Test Accuracy
Novel Architecture	**97.91%**	**91.78%**
AlexNet	97.45%	97.50%
VGG16	97.94%	98.16%
ResNet18	97.87%	97.83%

Figure 14. Accuracy comparisons for CNN models for OCR use cases

5. Conclusion

Based on the outcomes of our comparisons, it can be concluded that this innovative approach to an OCR tailored for handwritten digits successfully identifies superior alternatives to currently employed techniques and methods, achieving a better balance between accuracy and efficiency. Significant advancements are observed across various domains, including Noise Removal, Skew Correction, Skeletonization/Thinning, and, to a slightly lesser extent, Character Classification through the use of a CNN. The proposed algorithms and architectures not only demonstrate adaptability but also facilitate the enhancement of deprecated or obsolete methods, optimizing them for the specific task at hand. The remaining components of the OCR, encompassing Normalization, Grayscale Conversion, Thresholding, Line Separation, and Character Separation, exhibit performance on par with contemporary standards. In summation, the combined product of the proposed mechanisms in this paper emerges as a viable and competent alternative for OCR systems targeting handwritten English digits.

References

1. Recommendation 601——one world standard; digital video and the world digital studio standard. In *Modern Television Systems*, pages 116–134. CRC Press, 12 1991.
2. Yasser Alginahi. Preprocessing techniques in character recognition. In *Character Recognition*. Sciyo, 8 2010.
3. Kaneria Avni. Image denoising techniques: A brief survey. *The SIJ Transactions on Computer Science Engineering amp; its Applications* (CSEA), 03(01):01–06, 2 2015.
4. Kanika Bansal and Rajiv Kumar. K-algorithm: A modified technique for noise removal in handwritten documents. *International Journal of Information Sciences and Techniques*, 3(3):1–8, 5 2013.
5. Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre van Schaik. EMNIST: extending MNIST to handwritten letters. *In 2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 5 2017.
6. Zicheng Guo and Richard W. Hall. Parallel thinning with two-subiteration algorithms. *Communications of the ACM*, 32(3):359–373, 3 1989.
7. Himanshu Jain and Archana Praveen Kumar. A sequential thinning algorithm for multi-dimensional binary patterns. 10 2017.
8. Fan L, Zhang F, Fan H, and Zhang C. Brief review of image denoising techniques. *Visual computing for industry, biomedicine, and art*, 2:7, Jul 2019.
9. M Melhi, S.S Ipson, and W Booth. A novel triangulation procedure for thinning hand-written text. *Pattern Recognition Letters*, 22(10):1059–1071, 8 2001.
10. Jamshed Memon, Maira ami, Rizwan Ahmed Khan, and Mueen Uddin. Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR). *IEEE Access*, 8:142642–142668, 2020.
11. S. Mohammad Mostafavi, Iman Abaspur Kazerouni, and Javad Haddadnia. Noise removal from printed text and handwriting images using coordinate logic filters. In *2010 International Conference on Computer Applications and Industrial Electronics*. IEEE, 12 2010.
12. Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1 1979.
13. Luan Pham, Phu Hao Hoang, Xuan Toan Mai, and Tuan Anh Tran. Adaptive radial projection on fourier magnitude spectrum for document image skew estimation. In *2022 IEEE International Conference on Image Processing* (ICIP). IEEE, 10 2022.
14. R. Plamondon, M. Bourdeau, C. Chouinard, and C.Y. Suen. Validation of preprocessing algorithms: A methodol- ogy and its application to the design of a thinning algorithm for handwritten characters. In *Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR '93)*. IEEE Comput. Soc. Press, None.
15. Afgani Fajar Rizky, Novanto Yudistira, and Edy Santoso. Text recognition on images using pre-trained CNN. 02 2023.
16. Khalid Saeed, Marek Tabedzki, Mariusz Rybnik, and Marcin Adamski. K3M: A universal algorithm for image skeletonization and a review of thinning techniques. *International Journal of Applied Mathematics and Computer Science*, 20(2):317–335, 6 2010.

17. Satoshi Suzuki and Keiichi Abe. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 4 1985.
18. T. Y. Zhang and C. Y. Suen. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3):236–239, 3 1984.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.