https://doi.org/10.2352/EI.2025.37.2.SDA-339

A Novel Post-Processing Method for Convolutional Neural Networks in Character Recognition

Jarmo Koponen; School of Computing, Kuopio campus; University of Eastern Finland, P. O. Box 1627, FI-70211; Kuopio, Finland

Abstract

This research presents a novel post-processing method for convolutional neural networks (CNNs) in character recognition, specifically designed to handle inconsistencies and irregularities in character shapes. Convolutional Neural Networks (CNNs) are powerful tools for recognizing and learning character shapes directly from source images, making them well-suited for recognition of characters that contain inconsistencies in their shapes. However, when applied to multi-object detection for character recognition, CNNs require post-processing to convert the recognized characters into code sequences, which has so far limited their applicability. The developed method solves this problem by directly post-processing the inconsistent characters identified by the convolutional neural model into labels corresponding to the source image. An experiment with real pharmaceutical packaging images demonstrates the functionality of the method, showing that it can handle different numbers of characters and labels effectively. As a scientific contribution to the fields of imaging and deep learning, this research opens new possibilities for future studies, particularly in the development of more accurate and efficient multi-object character recognition with post-processing and their application to new areas.

Introduction

Character recognition is a fundamental task in various applications, ranging from document digitization to automated processing of packaging labels. A character can be defined as a pattern that closely matches one of the symbols that the recognition system is designed to identify [1]. Optical character recognition (OCR) techniques work effectively only with clear characters in high-quality images on an uncomplicated background, while requiring character consistency in terms of format and viewing angle [2].

Methods with deep neural networks has made it possible to recognize distorted text and irregular characters on surfaces exposed to light, even in low-quality images. In these methods, deep-learning neural networks have been defined for character recognition to extract and learn the features of text regions and characters from a large set of training images, and then recognize the characters in subsequent images using the model they have learnt. Such methods greatly contribute to the recognition of packaging texts in real-life conditions [2].

In the pharmaceutical industry, ensuring the accurate and efficient tracking of batch and expiration date codes is critical for regulatory compliance and patient safety. Current methods struggle with the variability in printing quality and character recognition, leading to inefficiencies and errors in automated systems. Worldwide, the distribution of medicines is strictly regulated. The batch code and expiration date [3] of each package are precisely verified and documented when receiving medicines from wholesalers and distributing them to customers in pharmacies [4]. This repetitive and precision-demanding task is poorly suited for humans. Automated dispensing systems automate the retrieving, shelving, and distributing pharmaceutical products, freeing up staff time for more meaningful expert work. Cartons are used for a high percentage of pharmaceutical products due to versatility and protective properties [5]. Pharmaceutical industry manufacturers select printing methods tailored to their specific needs [5]. Thermal and laser printers deliver high quality prints. In contrast inkjet printers easily print streaks, blurs, and reflective gloss. Dot matrix printers, depending on the quality of the ribbon, produce lower quality prints with low contrast codes [5]. The variability in character appearances from these printing methods reduces Optical Character Recognition (OCR) accuracy, achieving high accuracy when character shapes display minimal diversity [6]. Incorporating deep learning text recognition into pharmacy automation is a crucial step towards overcoming recognition challenges.

Deep neural networks for object detection can be categorized into one-stage and two-stage detectors, both of which have their own benefits. Single-stage detectors typically sacrifice some accuracy for real-time capabilities, while two-stage detectors offer high accuracy but lower execution speed [7]. R-CNN, a pioneering two-stage method, stands for 'Regions with Convolutional Neural Networks,' and combines region proposals with convolutional neural networks. Developed by Ross Girshick and his collaborators, R-CNN represented a major advance in the field of computer vision when it was introduced [8]. Although the newer versions of R-CNN are often discussed, the original model had several excellent features, such as a fixed-size 4096-dimensional feature vector for all candidate regions, which ensures that the features of each region are accurately extracted and learned, and a bounding box refinement layer, which provides highly accurate bounding box coordinates, as demonstrated by the high Intersection over Union (IoU) values. R-CNN uses selective search to generate region proposals by segmenting images into various scales and iteratively merging these based on feature similarities generating region proposals for potential object locations in image. Each region proposal is scaled to a fixed size for a full feature map, which is then processed by a convolutional network to extract features and classify the regions.

A trained model uses CNN to predict whether a region proposal contains an object and, if so, its class. Class-specific linear SVM classifiers are trained using fixed-length features extracted by the CNN. The classification layer utilizes SVM to predict the class of each region proposal. The Bounding Box Regression Layer produces adjustment values for the bounding boxes of the region proposals, allowing the creation of accurate, class-specific bounding boxes [9], in addition to confidence scores for each detection. Overlapping region proposals are pruned using the non-maximum suppression (NMS) method, ensuring that each identified object is only represented in the results only once, based on the proposal with the highest score. In multi-object recognition, R-CNN assigns each proposal region to the class with the highest score value, allowing for accurate recognition of multiple objects from the same image.

Efficient pre-processing method enables high-precision binarization of source images allowing the R-CNN model to recognize expiration date and batch code characters of pharmaceutical packages with different printing methods with high

accuracy [10]. Different printing methods cause inconsistencies and varying letter widths and heights in character shapes, reflected in the bounding box dimensions of the model's output. In the model's output, the default order of recognition results is based on their accuracy, with the class having the highest recognition score listed first. They do not correspond to their order in the source image. Therefore, a targeted and efficient post-processing method for the model's output layer data into package-specific manufacturing marking codes needs to be developed.

The novel method developed in this study post-processes the CNN model's recognition results from pharmaceutical package source images. It interprets the spatial relationships of the data and, by horizontally enlarging bounding boxes, identifies all code regions formed by overlapping bounding boxes. The method then reconstructs and arranges these code sequences in an order that precisely aligns with the layout of the source image, accurately outputting manufacturing markings, such as batch and expiration date codes.

Results: This paper presents a novel post-processing method combined with CNN models to study the recognition of variably positioned and irregularly shaped text characters on pharmaceutical packaging. The post-processing method reconstructs and arranges batch and expiration codes in complex layouts by interpreting spatial relationships between characters. Bounding boxes are expanded horizontally, allowing overlapping regions to merge into coherent code areas, which are then identified and ordered to reflect the original layout in the source image. The results show that when the CNN model accurately identifies characters, the post-processing method arranges them to match the source image order, achieving a 100% recognition score. In cases of initial recognition inaccuracies, the post-processing method improves overall recognition performance, despite being unable to correct missing or incorrect characters. Furthermore, based on the results of this paper, we present new ideas to inform future research efforts.

The structure of this paper is as follows. Section II reviews deep learning approaches to text recognition, highlighting a gap in post-processing inconsistent character shapes recognized by CNNs. Section III presents the post-processing method developed to organize recognition results from deep learning models. Section IV describes the experimental setup used to assess the method's performance. Section V reports the initial recognition results and baseline evaluations. Section VI provides an analysis of the experimental results, demonstrating the method's effectiveness. Finally, Section VII discusses the findings and proposes directions for future research.

Related works

Convolutional Neural Networks (CNNs) excel at recognizing diverse text types—handwritten, computer fonts, and scene text—within a single model trained on their salient features [11]. Despite this, there has been insufficient research on the consistent post-processing of recognized characters across entire source images. Existing methods often overlook post-processing steps that ensure consistent organization and recognition, especially when characters are irregularly shaped [12]. Techniques like bounding box refinement and error correction, while commonly discussed, assume uniformity in text shapes, which is often not the case in practical scenarios such as pharmaceutical packaging [13, 14, 15]. For general object detection, methods have been designed for objects with clear boundaries [16], but text detection presents more complex challenges due to the fragmentation of text lines and characters. Connectionist Text Proposal Network (CTPN) localizes text by predicting vertical positions and classifying text proposals using anchor regression, which improves the accuracy crucial for comprehensive text reading. However, these methods, including CTPN, primarily focus on text detection and localization without converting the recognized text into consistent sets of labels, especially when dealing with irregular character shapes. Additionally, these methods lack testing with real-world images containing varying numbers of markers and labels, where consistent processing is required [17]. In addition, many methods developed for scene text detection with CNNs, such as CRAFT [18], TextSnake [19], and SegLink [20], focus solely on detecting text in different orientations and shapes, but they do not address character shape inconsistencies across images. Even advanced frameworks like TextDragon [21], which provide end-to-end text recognition solutions, do not effectively handle the recognition or post-processing of inconsistently formatted characters. Additionally, approaches using reinforcement learning to refine bounding boxes have improved text alignment but still fall short of addressing the broader need for consistent post-processing across diverse and irregular text formats [22].

This review of related work highlights a significant gap in current research on post-processing CNN-recognized characters, particularly in scenarios with inconsistent character shapes.

Post-Processing the deep learning models recognition results

In this section, the developed novel post-processing method for a character-based convolutional neural network is reviewed. The following sections are organized as follows: **Section A** presents a brief background on previous work in recognizing inconsistent characters on pharmaceutical packaging. **Section B** addresses the need for post-processing of CNN recognition results. **Section C** discusses the operation of the post-processing method, and **Section D** provides a mathematical description of the proposed method.

A. Background:

Pharmaceutical packages pose significant challenges to text recognition due to the variety of character shapes produced by different printing methods and the outward curvature of the cardboard packaging, which varies in degree. To address these issues, the dataset consisted of images taken from actual cardboard pharmaceutical packages with manufacturing markings printed using different methods. These images were first preprocessed and then binarized using a method detailed in [10]. The resulting binarized images were used to train the novel deep learning application for text recognition with a R-CNN model, as described in detail in [2].

B. The Need for Post-Processing in Character Recognition:

Although the developed R-CNN-based character recognition application recognizes characters with inconsistent shapes in the dataset with high precision and recall, its output layer requires post-processing, as the model orders characters based on recognition score rather than their sequence in the source image, as illustrated in Table 1. This misalignment presents a significant challenge for applications that require a precise text sequence, such as manufacturing labels with expiration dates and batch codes, which require characters to be read in a specified order to ensure correctness. To address this, a post-processing method is required to reorganize the R-CNN recognition output to precisely align with the exact sequence of manufacturing marking codes as they appear in

the source image. The task is further complicated by the various layouts of manufacturing marking codes on pharmaceutical packaging, as well as variations in character count and line structure. Furthermore, when recognizing characters with varying shapes due to inconsistent printing methods, the dimensions of the bounding boxes in the R-CNN model output also vary. This variability prevents the use of a simplified technique based on uniform row height or consistent bounding box size, as demonstrated in the pharmaceutical package image in Table 1.

Recognized Characters and Bounding Boxes	BoundingBoxCoordinates	Score	Label
	614, 317, 628, 350	0.99992	'1'
	633, 254, 655, 279	0.99982	'2'
	560, 321, 591, 345	0.99962	'2'
	213, 255, 242, 272	0.99947	'Era'
	590, 318, 606, 348	0.99947	'0'
	221, 298, 252, 322	0.99900	'Kayt'
	566, 253, 592, 279	0.99880	'2'
	614, 252, 629, 279	0.99873	'0'
	273, 303, 302, 320	0.99854	'viim'
	326, 299, 358, 323	0.99783	'Utg'
	524, 316, 544, 346	0.99747	'8'
	630, 318, 654, 348	0.99676	'8'
	501, 318, 521, 347	0.99583	'0'
	371, 299, 400, 322	0.99479	'dat'
	260, 256, 296, 274	0.99404	'Sats'
	550, 251, 568, 283	0.99290	'6'
	517, 251, 553, 281	0.98861	'8'
	589, 251, 612, 283	0.96254	'3'

Table 1: *Default R-CNN recognition output for batch and expiration date codes on pharmaceutical packaging.*

In the left image, characters recognized from pharmaceutical package are each enclosed by bounding boxes. The table on the right provides the corresponding bounding box coordinates, recognition confidence scores, and character labels from the model.

C. Post-Processing Method:

Interpreting Spatial Relationships in Character Recognition: The developed post-processing method interprets the spatial relationships of characters by analyzing the distances between their bounding boxes in the model's output layer. First, the method expands the bounding boxes of the recognized characters, causing adjacent bounding boxes to overlap. Next, the overlapping bounding boxes are identified using a graph data structure [23], which detects the formed regions of connected components. Furthermore, the boundaries of the connected components are defined based on the contours of the enclosed bounding boxes. The center points of the resulting code regions are then calculated, arranged, and indexed in the natural human reading order, from top-left to bottom-right. Finally, the characters within the bounding boxes are output, preserving their original arrangement as in the source image.

The flowchart representing this process is shown in Figure 1.

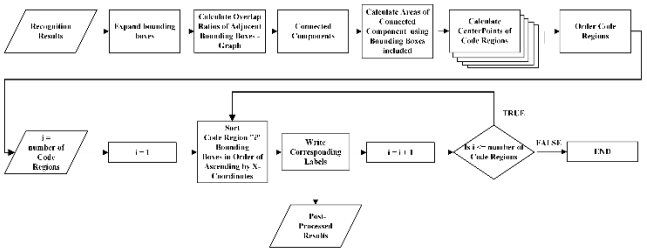

Figure 1. Post-Processing Method for R-CNN-Based Character Recognition.

D. Mathematical Description of the Post-Processing Method

To provide a detailed understanding of the method, this section presents a mathematical description of bounding box properties, including expansion principles, overlap detection, identification of connected components, code region formation, ordering of code regions, and the process for writing the final output.

Bounding Box Representation

To accurately detect and analyze character objects within an image, specific bounding box properties are defined and utilized. For each detected character object, the bounding box is represented as: $B_C = (x_{min}, y_{min}, x_{max}, y_{max})$ (1)

The bounding box properties are defined as follows:

$$\text{Width: } (\text{width}(B_C) = x_{max} - x_{min}) \quad (2)$$

$$\text{Height: } (\text{height}(B_C) = y_{max} - y_{min}) \quad (3)$$

Centroid:

$$x_c = \frac{x_{min}+x_{max}}{2}, \quad y_c = \frac{y_{min}+y_{max}}{2} \quad (4)$$

The bounding box is then expanded along the x-axis by a factor $\propto$ to allow adjacent bounding boxes to overlap:

Expanded Bounding Box:

$$x'_{min} = x_c - \alpha \cdot (x_{max} - x_{min}) \quad (5)$$

$$x'_{max} = x_c + \alpha \cdot (x_{max} - x_{min}) \quad (6)$$

Overlap Condition

To determine whether two bounding boxes overlap in the x-direction, the following condition is used:

$$x'_{1,max} > x'_{2,min} \quad \text{and} \quad x'_{2,max} > x'_{1,min} \quad (7)$$

Graph Construction

Using the bounding boxes B_C, the overlap ratio between any two bounding boxes B_{C1} and B_{C2} is calculated as the Intersection over Union (IoU), defined as:

$$Overlap\ Ratio = \frac{Area\ of\ Intersection}{Area\ of\ Union} \quad (8)$$

After calculating the overlap ratio, an undirected graph is constructed, where nodes represents characters and edges indicate overlaps between bounding boxes. The overlap ratio between a bounding box and itself is set to zero to simplify the graph representation. Each node in the graph corresponds to a recognized character represented by the bounding box associated with that character. A graph $G = (V, E)$ is then constructed, where each node $v_i \in V$ corresponds to a character represented by a bounding box B_{Ci}, and each edge $e_{ij} \in E$ represents the overlap between the bounding boxes B_{Ci} and B_{Cj}. An edge e_{ij} exists between the nodes v_i and v_j if there is any overlap between their bounding boxes.

Identifying Connected Components:

Using the constructed graph G, connected components Ck are identified. Each node in the graph represents a recognized character, and edges between nodes represents overlaps between the bounding

boxes of these characters. The connected components, Ck, represent groups of characters (nodes) that are spatially related, forming a contiguous code region within the image. These connected components are essential for grouping characters that must be processed together as part of the same code or region.

Bounding Regions of Connected Components (Code Regions)

For each connected component Ck , referred to here as a code region, the bounding region $B(Ck)$ is determined by combining the bounding boxes of all characters within the component. Here, B_i represents the bounding box of the *i*-th character within the connected component Ck, extending the general bounding box definition B_C presented earlier.. The combined bounding box B_{Ck} for a connected component Ck is given by:

$$x_{min}^{Ck} = \underset{B_i \in C_k}{min}\left(x_{min}^{B_i}\right), \quad x_{max}^{Ck} = \underset{B_i \in C_k}{max}\left(x_{min}^{B_i} + width(B_i)\right) \tag{9}$$

$$y_{min}^{Ck} = \underset{B_i \in C_k}{min}\left(y_{min}^{B_i}\right), \quad y_{max}^{Ck} = \underset{B_i \in C_k}{max}\left(y_{min}^{B_i} + \text{height}(B_i)\right) \tag{10}$$

The centroid (x_c^{Ck}, y_c^{Ck}) of each connected component Ck is calculated as:

$$x_c^{Ck} = \frac{x_{min}^{Ck} + x_{max}^{Ck}}{2}, \quad y_c^{Ck} = \frac{y_{min}^{Ck} + y_{max}^{Ck}}{2} \tag{11}$$

These centroids are used to spatially organize the connected components, aiding in the subsequent processing steps such as sorting and indexing.

Ordering of Connected Components

After calculating the centroids, the connected components $C_{k1}, C_{k2}, \ldots, C_{km}$ are sorted in ascending order based on the y-coordinates of their centroids, from the smallest y-coordinate (top) to the largest y-coordinate (bottom):

$$C_{k1}, C_{k2}, \ldots, C_{km} \tag{12}$$

If two components have the same y-coordinate, they are then ordered by their x-coordinates:

$$\text{If} \quad y_c^{C_{ki}} = y_c^{C_{kj}} \text{, then } x_c^{C_{ki}} \leq x_c^{C_{kj}} \tag{13}$$

After sorting, each connected component is indexed based on this order, from smallest to largest, and this index is used in the final output process.

Retrieving and Arranging Characters

Once the code regions are ordered, the bounding boxes $B_{k1}, B_{k2}, \ldots, B_{kn}$ within each connected component C_k are arranged in ascending order by their x- coordinates:

$$x_{min}^{B_{k1}} \leq x_{min}^{B_{k2}} \leq \cdots \leq x_{min}^{B_{kn}} \tag{14}$$

The characters m_{ki} where m_{ki} represents the *i*-th character within each code region C_k, are then arranged in the same order:

$$S(C_k) = m_{k1}, m_{k2}, \ldots, m_{kn} \tag{15}$$

Generation of Final Output from Ordered Character Sequences

The sequences $S(C_k)$ of all connected components are combined in the order of their indices to generate the final result:

$$S_{\text{final}} = \bigcup_{i=1}^{m} S(C_{ki}) \tag{16}$$

Experimental Setup

This section details the experimental setup used to assess the performance of the post-processing method in recognizing text characters on pharmaceutical packaging.

Imaging and Preprocessing

A total of 19 pharmaceutical packaging samples were imaged using a fixed camera setup, with four light sources positioned diagonally above, below, left, and right of the package surface. Each image was sequentially illuminated and captured. The camera axis was perpendicular to the package surface [10]. Subsequently, the images were then pre-processed by converting them to grayscale, followed by adaptive filtering to enhance the contrast between the text and the background while removing the noise and binarization using OTSU-method. The binarized sub-images from each light source were then merged into the final result image [10], which was used as input for the deep learning model in the subsequent stage.

Character Recognition Using the CNN Model

Formed result images were fed into the R-CNN model with a pre-trained AlexNet, which was adapted using transfer learning to recognize characters on pharmaceutical packaging. The CNN extracted features from the resized region proposals, and the classification layer, utilizing an SVM, assigned each region to one of 43 character classes. The bounding box refinement layer of the R-CNN enhances the localization accuracy of detected text characters. Non-Maximum Suppression (NMS) eliminates overlapping region proposals, ensuring that only the highest-confidence identification for each object is retained. This approach enables the model to recognize characters in expiration dates and batch codes, despite inconsistencies in character shapes. [2]

Initial Recognition Results and Baseline Performance Evaluation

To establish a baseline for evaluation of the novel post-processing method, the R-CNN model's recognition performance was first assessed using the dataset described in Section III, A: Background. Ground truth characters were manually assigned and saved for each image to enable direct comparison with the model's output. The evaluation was conducted in two stages, dividing the images into two subsets based on the model's recognition performance. In the first stage, the post-processing method was applied to recognition results from images where the model produced correct outputs. In the second stage, the method was tested on results from images containing errors, forming a structured research pipeline.

Figure 2 provides visual examples of recognition results, illustrating the model's performance across images with both high and lower precision and recall rates, These images represents the two-stage evaluation of the post-processing method, applied to images where the model exhibited high precision and recall (top row), as well as images with lower precision and recall performance (bottom row).

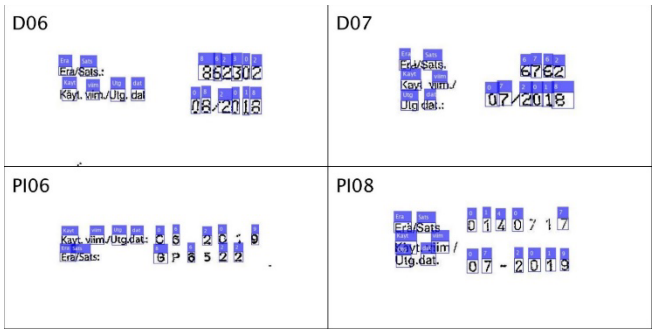

Figure 2: Example recognition results for manufacturing marking texts on pharmaceutical packaging. The top row show images with high recognition precision and recall, while the bottom row illustrates reduced precision and recall.

Figure 3 represent a visual breakdown of the model's recognition performance across all dataset images, the correctly identified, missing and incorrectly recognized text objects are shown image-by-image.

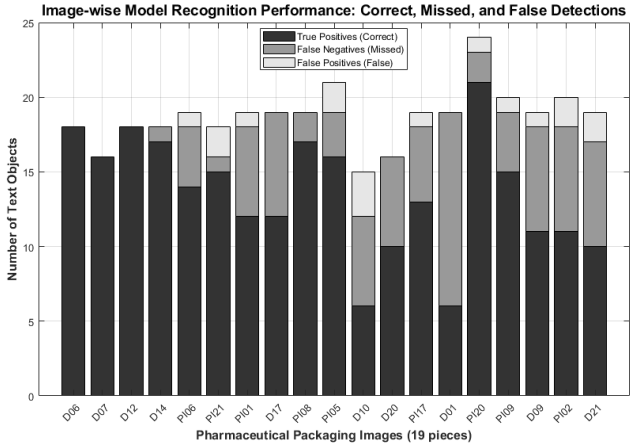

Figure 3. Visual comparison of the model's recognition performance for 19 pharmaceutical package image dataset. Correctly identified, missing, and falsely identified text objects are shown per image. For a detailed numerical breakdown, refer to the Table 2.

Table 2 presents a numerical breakdown of the model's performance, showing the total number of ground truth characters (GT_Count), true positives (TP), false negatives (FN), and false positives (FP) for each image.

Image	D06	D07	D12	D14	PI06	PI21	PI01	D17	PI08	PI05	D10	D20	PI17	D01	PI20	PI09	D09	PI02	D21	Totals
GT_Count	18	16	18	18	18	16	18	19	19	19	12	16	18	19	23	19	18	18	17	339
TP	18	16	18	17	14	15	12	12	17	16	6	10	13	6	21	15	11	11	10	258
FN	0	0	0	1	4	1	6	7	2	3	6	6	5	13	2	4	7	7	7	81
FP	0	0	0	0	1	2	1	0	0	2	3	0	1	0	1	1	1	2	2	17

Table 2. Numerical breakdown of the model's recognition performance. The table shows ground truth text characters (GT_Count), correctly recognized characters (TP), missing characters (FN), and falsely recognized characters (FP) for each image, with totals in the final column.

The Levenshtein distance [24] metric was used to quantify the accuracy of the model's output relative to the ground labels, as well as the improvement achieved through post-processing. Levenshtein distance calculates the minimum number of operations (insertion, deletion, or substitution) required to transform one string into another. This metric is instrumental in evaluating how closely the recognized text matches the ground truth. To further assess the models and post-processing methods performance, the recognition score was calculated based on the Levenshtein distance using the equation (17):

$$\text{Recognition Score (\%)} = \left(1 - \frac{Levenshtein\ distance}{maxLen}\right) \times 100 \qquad (17)$$

Where maxLen is the length of the longer string, either the ground truth or the recognized text. This equation generates a percentage score, with 100% indicating identical strings and lower percentages indicating greater differences. The shorter the Levenshtein distance relative to the longest character string, the higher the recognition score. To obtain the recognition score for each image, the deep learning model's output and the ground truth text were compared using Levenshtein distance. Recognition scores were calculated both for the initial output of the model and after the application of the post-processing method.

Experimental Results

The developed post-processing method was evaluated in two stages. In the first stage, recognition scores were calculated for a subset of images where the model produced correct recognition results, and the results were analyzed. In the second stage, recognition scores were calculated for a subset of images containing recognition inaccuracies, and the results were analyzed.

A. Post-Processing Performance Using Images with Correct Recognition

Table 3 presents the ground truth text, the detected text produced by the model, and the post-processed text, along with recognition scores for the detected text (Recognition Score Detected) and the post-processed results (Recognition Score Post-Processed) across the selected images.

Image	Ground Truth Text	Recognition Score Detected	Detected Text	Recognition Score Post-Processed	Post-Processed Text
D06	Era Sats 8 6 2 3 0 2 Kayt viim Utg dat 0 8 2 0 1 8	15	0 2 2 8 6 8 8 0 3 0 Sats Utg dat 1 viim Era 2 Kayt	100	Era Sats 8 6 2 3 0 2 Kayt viim Utg dat 0 8 2 0 1 8
D07	Era Sats 6 7 6 2 Kayt viim 0 7 2 0 1 8 Utg dat	16	6 2 0 8 6 0 Era dat 7 2 viim Sats 7 Kayt 1 Utg	100	Era Sats 6 7 6 2 Kayt viim 0 7 2 0 1 8 Utg dat
D12	Era Sats 8 6 2 3 0 2 Kayt viim Utg dat 0 8 2 0 1 8	15	0 2 2 8 6 8 8 0 3 0 Sats Utg dat 1 viim Era 2 Kayt	100	Era Sats 8 6 2 3 0 2 Kayt viim Utg dat 0 8 2 0 1 8

Table 3. Post-processing results for correctly recognized text by the model. Comparison of ground truth text, model-recognized text, and post-processed text, with recognition scores for both detected and post-processed results

The achieved recognition scores for all images highlight the post-processing method's effectiveness in aligning recognized characters with the ground truth. A recognition score of 100 indicates perfect alignment between the post-processed text and the

ground truth text. The method successfully processes the output layer data, addressing variations in character locations and bounding box sizes, to align the recognized characters accurately with the ground truth sequence in the source image.

B. Post-Processing Performance Using Images with Recognition Inaccuracies

Table 4 presents the ground truth text, the detected text produced by the model, and the post-processed text, along with recognition scores for the detected text (Recognition Score Detected) and the post-processed results (Recognition Score Post-Processed) across selected images with recognition inaccuracies.

Image	Ground Truth Text	Recognition Score Detected	Detected Text	Recognition Score Post-Processed	Post-Processed Text
D14	Era Sats 8 6 2 3 0 2 Kayt viim Utg dat 0 8 2 0 1 8	27	6 0 2 Era 0 8 0 2 2 1 8 Utg Sats viim dat 8 Kayt	97	Era Sats 8 6 2 0 2 Kayt viim Utg dat 0 8 2 0 1 8
PI06	Kayt viim Utg dat 0 6 2 0 1 9 Era Sats G P 6 5 2 2	21	2 Era Sats 2 6 Utg 2 dat 8 9 Kayt 0 0 viim 6	88	Kayt viim Utg dat 0 6 2 0 9 Era Sats 8 6 2 2
PI21	7 2 2 9 3 0 6 2 0 2 2 Lot Kayt viim Utg dat	27	0 3 Utg dat 0 7 2 2 2 9 Lot 6 7 Kayt 7 2 2 viim	77	7 0 2 6 2 7 7 9 2 0 3 2 2 Lot Kayt viim Utg dat

Table 4. - Recognition scores for images D14, PI06, and PI21 with model recognition inaccuracies, comparing ground truth, detected text and post-processed text for images with models' recognition inaccuracies, showing recognition scores for both detected and post-processed results.

Table 4 displays the recognition scores for images D14, PI06, and PI21, where the model's initial output included missed or incorrect recognitions. The post-processing method achieved significant improvements, increasing recognition scores by 70, 67, and 50 points for D14, PI06, and PI21, respectively. These results demonstrate the method's effectiveness in reorganizing and refining recognized characters, aligning their order as closely to the ground truth as permitted by the initial recognition results. However, the method's performance is limited by the initial model output, as it cannot recover characters that are completely missing or severely misrecognized.

Discussion

A novel post-processing method was developed to address challenges in aligning CNN-recognized text with ground truth in images containing manufacturing codes. The results demonstrate the method's ability to combine adjacent characters into codes and arrange them correctly. Despite the challenges posed by bounding box size variations, which significantly affect character arrangement, the method successfully organizes characters with the source image.

Validation on a real pharmaceutical packaging image dataset confirms the method's effectiveness in accurately post-processing recognition results into batch and expiration date codes. By post-processing the recognition results from the model's output layer, the method converts character recognition data into an electronic format suitable for pharmacy automation, enabling deep learning-based character recognition. A key advantage of the method is its ability to function effectively despite imprecise bounding box dimensions and placements, allowing consistent merging aligned with the source image. This ensures accurate generation of code sequences. The method arranges the characters within these code sequences in the same order as the source image, facilitating the digitization of batch and expiration date codes. However, the method's performance is constrained by the accuracy of the initial output, as it cannot recover missing or severely misrecognized characters, though it effectively reorders and aligns the recognized ones.

Future research will focus on investigating the use of a cascade R-CNN detector for the task of recognizing text characters, leveraging its optimized GPU acceleration and multi-stage bounding box regression in combination with the novel post-processing method introduced in this study.

References

[1] R. G. Casey, and E. Lecolinet, "A survey of methods and strategies in character segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 690-706, 1996.

[2] Koponen J, Haataja K and Toivanen P. Novel Deep Learning Application: Recognizing Inconsistent Characters on Pharmaceutical Packaging, *F1000Research, 12:427, https://doi.org/10.12688/f1000research.131775.2,* 2023.

[3] E. Bauer, *Pharmaceutical Packaging Handbook*, CRC Press, 2016.

[4] N. A. Shete, R. S. Mohan, R. N. Kotame, S. J. Gore, and R. R. Tagad, "Changing scenario of packaging in the pharmaceutical industry," *World Journal of Pharmaceutical Research*, vol. 9, no. 1, pp. 1728, 2020. Available: www.wjpr.net.

[5] World Health Organization, *Quality assurance of pharmaceuticals: a compendium of guidelines and related materials. Volume 2. Good manufacturing practices and inspection*, World Health Organization, 2024.

[6] J. Koponen, K. Haataja, and P. Toivanen, "Recent advancements in machine vision methods for product code recognition: A systematic review," *F1000Research*, vol. 11, 2022.

[7] P. Soviany, and R. T. Ionescu, "Optimizing the trade-off between single-stage and two-stage object detectors using image difficulty prediction," *arXiv preprint*, arXiv:1803.08707, 2018.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, Ohio, 2014, pp. 580-587.

[9] C. Chen, M. Y. Liu, O. Tuzel, and J. Xiao, "R-CNN for small object detection," in *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision*, Taipei, Taiwan, Nov. 20-24, 2016, Revised Selected Papers, Part V, Springer International Publishing, pp. 214-230.

[10] J. Koponen, K. Haataja, and P. Toivanen, "Text Recognition of Cardboard Pharmaceutical Packages by Utilizing Machine Vision," *Electronic Imaging*, vol. 33, pp. 1-7, 2021.

[11] A. F. Rizky, N. Yudistira, and E. Santoso, "Text recognition on images using pre-trained CNN," *arXiv preprint*, arXiv:2302.05105, 2023.

[12] B. N. Kumar Rao, K. Pranitha, Krishnaveni, C. V. Ranjana, and M. Chakkaravarthy, "Text Recognition from Images Using Deep Learning Techniques," in *Intelligent Computing and Applications: Proceedings of ICDIC 2020*, Singapore: Springer Nature, pp. 265-279, 2022.

[13] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *International Journal of Computer Vision*, vol. 116, pp. 1-20, 2016.

[14] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 2016, pp. 4159-4167.

[15] L. Neumann, and J. Matas, "Real-time lexicon-free scene text localization and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1872-1885, 2015.

[16] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *Computer Vision–ECCV 2016: 14th European Conference*, Amsterdam, The Netherlands, Oct. 11-14, 2016, Part VIII, Springer International Publishing, pp. 56-72.

[17] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257-276, 2023.

[18] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, 2019, pp. 9365-9374.

[19] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "Textsnake: A flexible representation for detecting text of arbitrary shapes," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, 2018, pp. 20-36.

[20] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, 2017, pp. 2550-2558.

[21] W. Feng, W. He, F. Yin, X. Y. Zhang, and C. L. Liu, "Textdragon: An end-to-end framework for arbitrary shaped text spotting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Seoul, Korea, 2019, pp. 9076-9085.

[22] J. Tang, W. Qian, L. Song, X. Dong, L. Li, and X. Bai, "Optimal boxes: boosting end-to-end scene text recognition by adjusting annotated bounding boxes via reinforcement learning," in *European Conference on Computer Vision (ECCV)*, Tel Aviv, Israel, Oct. 2022, pp. 233-248, Springer Nature Switzerland.

[23] J. A. Bondy, and U. S. R. Murty, *Graph theory with applications*, London: Macmillan, 1976.

[24] U. R. Abdurakhmanova, "Understanding the Levenshtein distance equation for beginners," *The American Journal of Engineering and Technology*, vol. 3, no. 6, pp. 134-139, 2021. Available: https://doi.org/10.37547/tajet/Volume03Issue06-24.

Author Biography

Jarmo Koponen received a bachelor's degree (2018) and a master's degree in software engineering from the University of Eastern Finland (2019). Since 2020, he has worked in the University of Eastern Finland as a project researcher and PhD student. He has +20 years of International work experience in development of machine vision systems for the paper and mining industries, whereof +15 years at Honeywell. In his current work, he has focused on character recognition involving surface curvature and diverse character shapes in the field of imaging and computer vision.