



SYLLABLE-BASED DETECTION AND RECOGNITION OF BRAILLE CELLS

Risky Aswi Ramadhani^{*}, Made Ayu Dusea Widyadara¹, Wahyu Cahyo Utomo¹, Marga Asta
Jaya Mulya²

¹Faculty of Engineering and Computer Science, Informatics Engineering, Universitas Nusantara PGRI
Kediri, Kediri City, East Java, Indonesia

²Electronics and Informatics Research Organization, Badan Riset Dan Inovasi Nasional (BRIN),
Tangerang, Indonesia

*Corresponding Author's Email: riskyaswiramadhani@gmail.com

Received: 13 February 2026

Revised: 4 April 2026

Accepted: 17 April 2026

Abstrak

Braille digitization plays an important role in improving access to written information for visually impaired individuals. However, automatic recognition of Braille text remains challenging due to the small size, dense spatial arrangement, and subtle variations of dot patterns. This study proposes a syllable-based Braille cell detection and recognition framework using a customized Faster R-CNN architecture. The proposed framework adopts a two-stage deep learning approach consisting of detection and classification. In the first stage, Faster Region-Based Convolutional Neural Network (Faster R-CNN) is utilized to detect and localize Braille cells through a Region Proposal Network (RPN). In the second stage, the detected regions are cropped and processed using an AlexNet-based convolutional neural network for syllable classification.

The method is evaluated on a dataset comprising 50 Braille syllable classes. A two-stage training strategy is applied to enhance localization and classification performance. Experimental results demonstrate stable training convergence, as indicated by decreasing loss values during training. Based on confusion matrix analysis, the classification model achieves an approximate accuracy of 93%, with precision of 92%, recall of 93%, and an F1-score of 92%. The proposed framework integrates detection and classification into a unified pipeline capable of generating labeled syllables from input images. These results demonstrate its potential for automated Braille digitization while highlighting opportunities for improvement through more comprehensive evaluation and larger datasets.

Keyword: Braille Recognition, Computer Vision, Faster R-CNN, Image Processing, Syllable-Based Detection.

Copyright: ©2026 The authors. This article is published by LPPM and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

INTRODUCTION

Braille is a tactile writing system that represents characters using combinations of raised dots arranged within a cell, enabling visually impaired individuals to access written information [1] [2]. Due to its effectiveness, Braille is widely used worldwide for communication and literacy among visually impaired communities [3], [4]. However, automatic recognition of Braille text from images remains a challenging task due to the small size of the dots, their dense spatial arrangement, and subtle variations in dot patterns. Most existing approaches for Braille recognition rely on direct

classification or segmentation-based methods. These approaches often require pre-segmented Braille cells or struggle with irregular layouts and noise, limiting their applicability in real-world scenarios. Furthermore, the large number of possible dot combinations, especially at the syllable level, increases the complexity of classification tasks.

In this study, the dataset consists of 50 syllable classes, each representing a distinct Braille character, which further complicates direct image-based classification using conventional convolutional neural network (CNN) architectures. To address these challenges, this study adopts an object detection-based approach using Faster Region-Based Convolutional Neural Network (Faster R-CNN). Faster R-CNN incorporates a Region Proposal Network (RPN) that generates candidate regions, which are then classified using a separate model [5], [6]. This approach enables the model to first localize Braille cells before performing classification, making it more suitable for handling small and densely arranged patterns. Faster R-CNN has been widely used in object detection tasks due to its effectiveness in localizing complex objects [7], [8], [9]. In this study, the Faster R-CNN algorithm employs the AlexNet model for both the region proposal network (RPN) and the classification task.

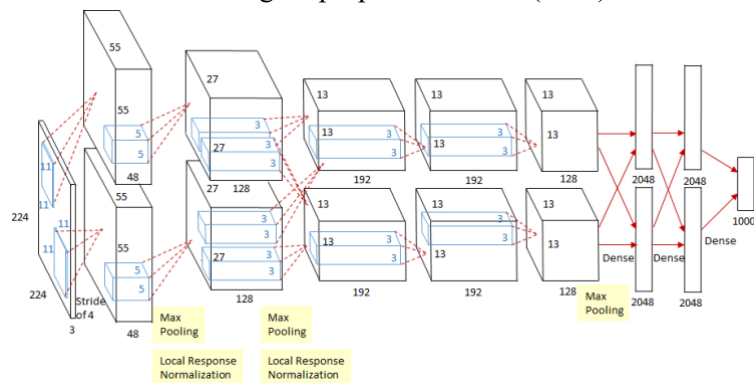


Figure 1. AlexNet Architecture[10]

AlexNet demonstrated superior performance in terms of recognition accuracy when compared to traditional machine learning and computer vision approaches [11], [12], [13]. This constituted a significant breakthrough in the field of machine learning and computer vision, particularly in the context of visual recognition and classification tasks [14], [15]. The implementation of the Faster RCNN algorithm in this study differs slightly from the original Faster RCNN algorithm [16]. In this study, the box generated by RPN will be the predicted box, and the region of interest (ROI) will take the image contained in the predicted box and then save it. The image will be classified using a classification model with a different framework. RPN uses the PyTorch framework, and for classification, TensorFlow is used, as illustrated in Figure 2.

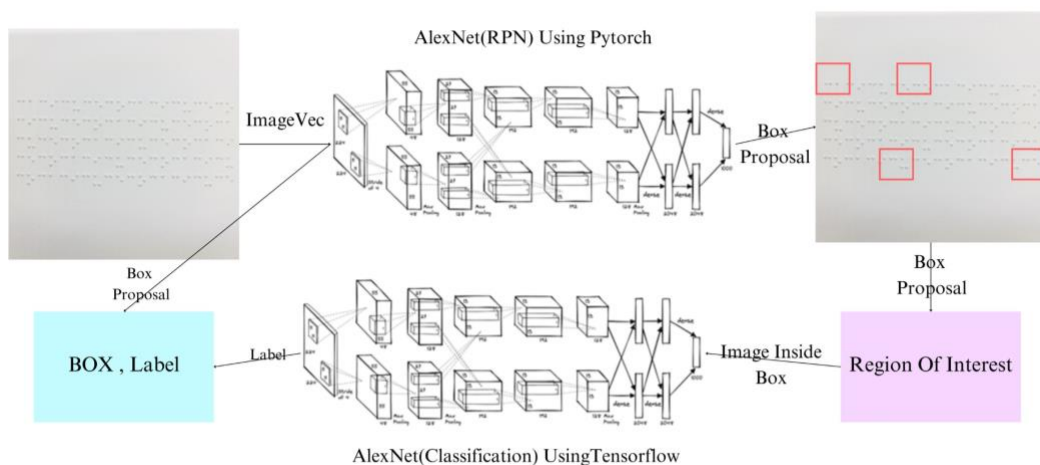


Figure 2. Faster RCNN using 2 framework

The main contributions of this study are as follows. First, a syllable-based Braille recognition framework is proposed using a customized Faster R-CNN architecture. Second, a two-stage detection–classification pipeline is developed to improve recognition performance for small and densely arranged Braille patterns. Third, a dataset consisting of 50 Braille syllable classes is utilized to support training and evaluation. The results demonstrate that the proposed approach is capable of effectively recognizing and classifying Braille characters from image data.

RESEARCH METHODS

The proposed method consists of several sequential stages, including Braille image acquisition, preprocessing, detection, cropping, and classification. Each stage processes the output of the previous stage to produce the final prediction of Braille syllables. To provide a clearer understanding of the overall workflow, the complete process of the proposed method is summarized in Algorithm 1.

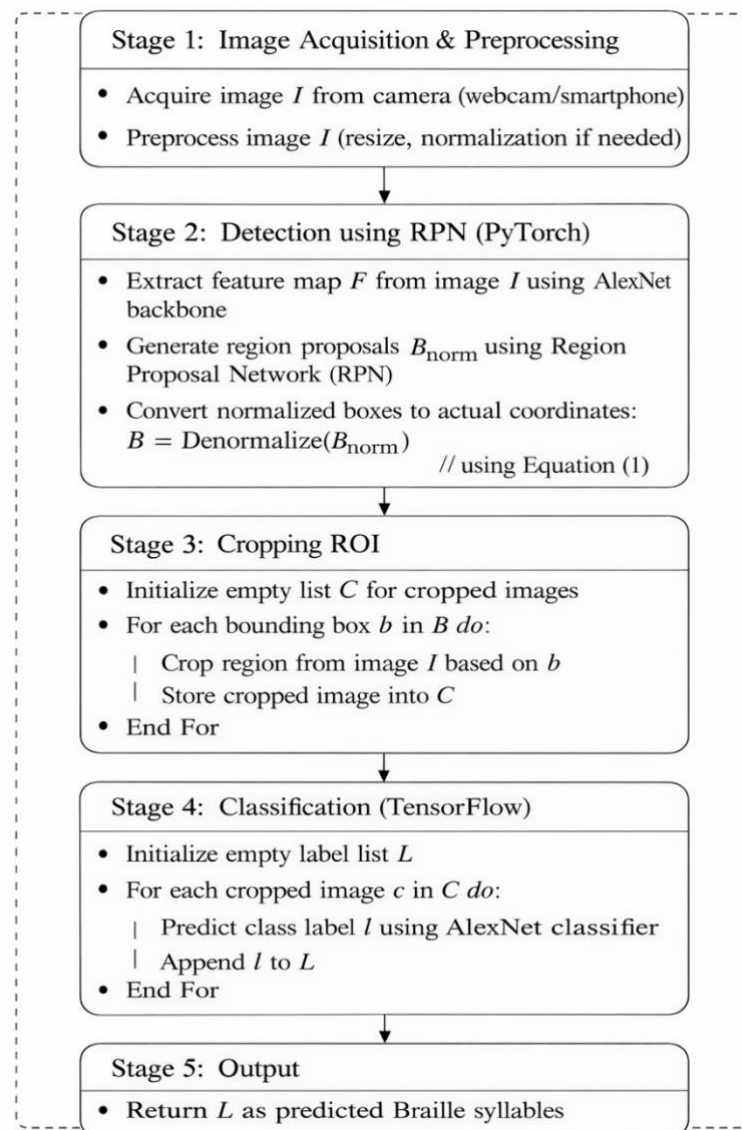


Figure 3. Algorithm 1- Braille Detection and Classification Pipeline

Algorithm 1 presents the overall workflow of the proposed system, covering image acquisition, preprocessing, Braille cell detection using a Region Proposal Network, region cropping, and syllable classification using the AlexNet model. This pipeline highlights the integration of detection and

classification stages within a unified framework, enabling efficient and accurate recognition of Braille syllables. Each stage in Algorithm 1 is further described in detail in the following subsections.

2.1 Braille Image Acquisition And Preprocessing

Braille image data is taken and collected using hardware such as a cellphone camera. All images were standardized to a resolution of 640×480 pixels to maintain consistency during model training and evaluation. The acquisition distance was maintained at approximately 10–30 cm from the Braille surface to preserve the visibility of embossed dot patterns. Data collection was conducted under controlled variations of illumination, including natural daylight and indoor lighting conditions, to simulate real-world usage scenarios. Furthermore, the dataset incorporates variations in embossing quality, including clearly raised dots, moderately visible patterns, and low-contrast or worn surfaces. These variations are intentionally included to enhance the robustness of the detection and classification models against differences in image quality and environmental conditions. All collected images were subsequently annotated using the LabelMe application by assigning bounding boxes to each Braille cell, producing labeled data for training the detection model.

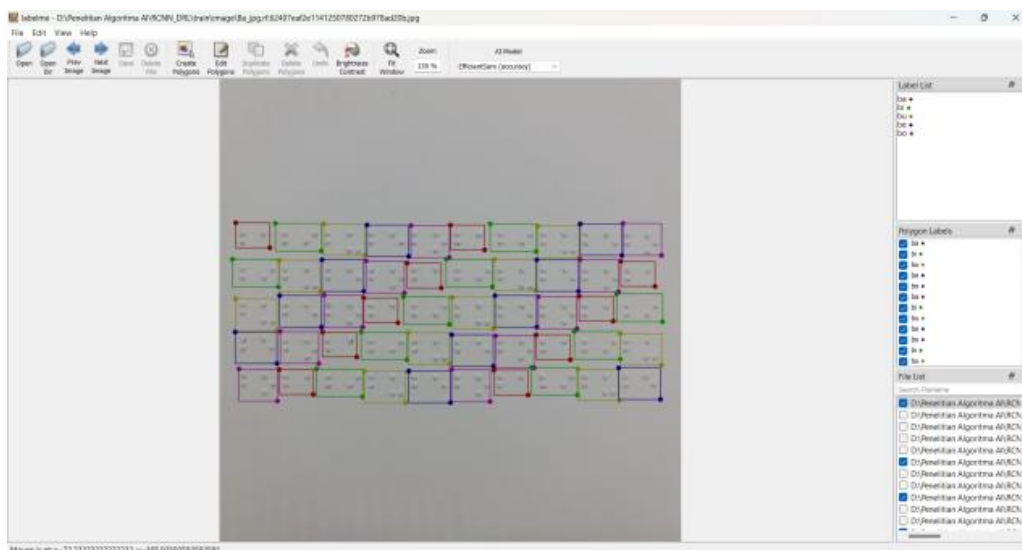


Figure 3. Labelme application

After labeling and giving boxes to the data using labels, the data is ready for training. First, the data goes into the box prediction model where the model learns the pattern to produce accurate box data between the actual box and the predicted box. Before entering the detection stage, the images undergo several preprocessing steps to improve quality and consistency. First, the images are converted into grayscale to reduce computational complexity. Then, Gaussian denoising is applied using a 3×3 kernel to reduce noise while preserving important structural features. After that, adaptive thresholding is performed to enhance the visibility of Braille dots, especially under uneven lighting conditions. Finally, normalization is applied to scale pixel values into a consistent range, ensuring stable input for the model.

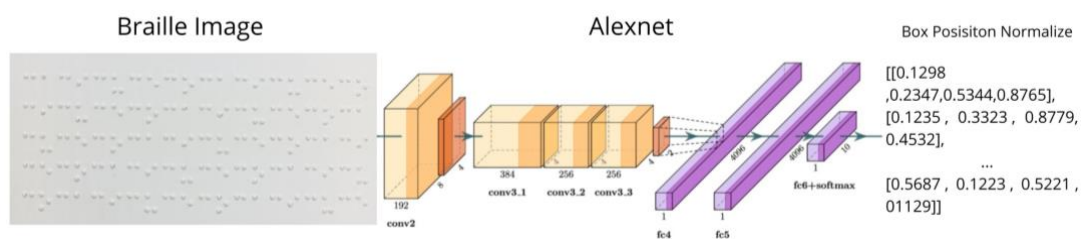


Figure 4. RPN for FasterRCNN

The process of the Braille image data flow can be seen in Figure 4 where the model learns to produce the position of the box in the form of a normalized vector, where in Figure 4 you can see an example of the results of the RPN model, the normalized box position is changed back to the actual position value (Denormalization , Equation 1) during the process in the ROI.

$$box = [xmin \times img_{width}, ymin \times img_{height}, xmax \times img_{width}, ymax \times img_{height}] \quad (1)$$

After that, cropping is done on the data based on the position of the box.

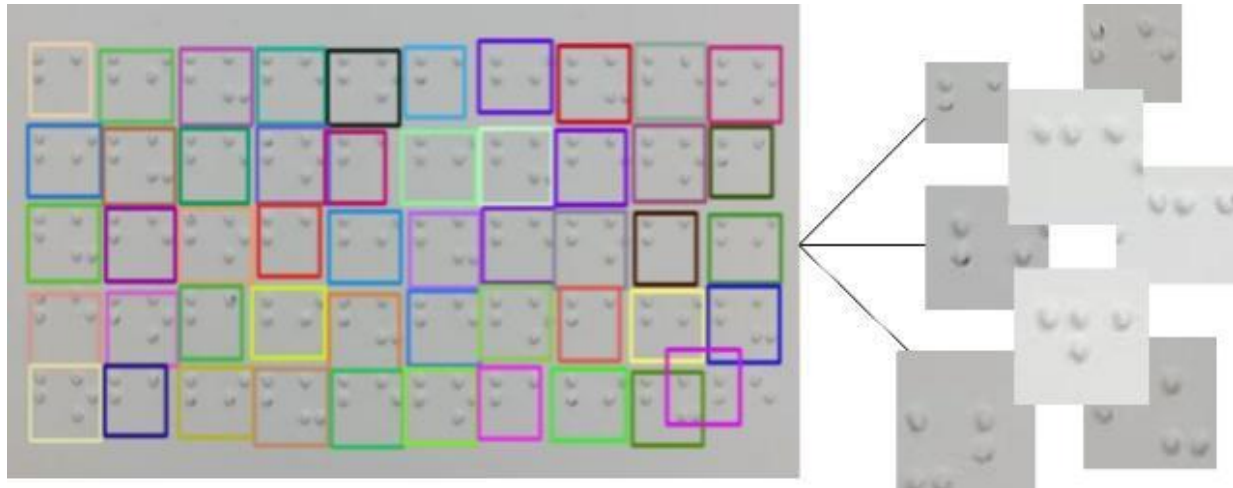


Figure 5. Cropping on the data based on the position of the box.

After the image goes through the cropping process, the image is put into temporary storage and then entered into the classification model where the cropped images are classified one by one to speed up the computation and do not require too large parameters.

After RPN is successfully trained, the next step is to create a classification head model / FasterRCNN, where the model is created with Alexnet as in the RPN model.

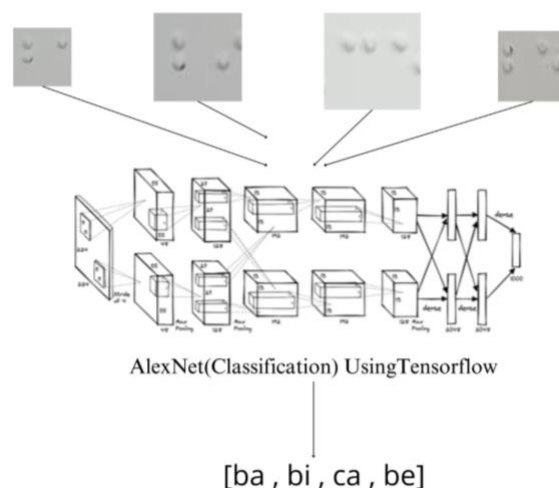


Figure 6. FasterRCNN

Then all the cropped dataset images based on their boxes are manually collected in different folders based on the original syllables of the cropped images, this is done to make it easier for the

algorithm to label images based on the folder name, then the dataset is divided based on its batch of 128 and for model training it requires 50 epochs

2.1.1 Faster RCNN

Faster R-CNN is an algorithm specifically designed for object detection and consists of three main components that work together efficiently, namely: (1) RPN (Region Proposal Network), which is responsible for generating proposal boxes for object detection; (2) ROI (Region Of Interest), which selects and focuses attention on relevant areas in the image; and (3) a classification network, which analyses the results from the ROI to classify the detected objects. However, there are other algorithms that are also used in this context, such as Fast R-CNN. According to Girshick and Wang [17], [18], the Fast R-CNN network receives the entire image as input, along with a list of object suggestions generated by the previous algorithm.

The process begins with the network processing the entire image using multiple convolutional layers and a maximum pooling layer to generate a convolutional feature map. Once the feature map is generated, for each given object proposal, the ROI pooling layer is tasked with extracting a fixed length feature vector from the feature map. This process is crucial as it allows the network to capture relevant information from objects of different sizes in the image, thus improving the accuracy and efficiency of the overall object recognition and classification. Thus, the application of the Faster R-CNN method provides a better solution for object recognition compared to previous approaches.

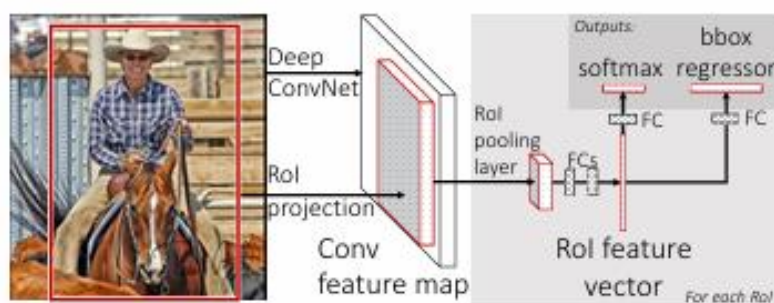


Figure 7. Fast R-CNN architecture[19]

However, the Fast R-CNN method uses a selective search technique that takes longer to determine the region or proposal box for object detection [20], [21]. Selective Search combines exhaustive search and segmentation by utilizing image structure to guide sampling [22], [23], but it often becomes a bottleneck in applications requiring high-speed processing. Therefore, Faster R-CNN is adopted in this study, as it replaces Selective Search with an RPN that can generate proposal boxes more efficiently and accurately. In addition to classical approaches, modern object detection frameworks such as YOLO, RetinaNet, and anchor-free detectors have also been widely developed to achieve real-time performance. Nevertheless, these methods generally prioritize speed over fine-grained localization accuracy, which is critical in Braille recognition tasks due to the small size, dense arrangement, and subtle spatial differences between Braille dots. Compared to these approaches, Faster R-CNN provides a better balance between detection accuracy and localization precision, making it more suitable for detecting Braille patterns.

Despite these advancements, existing studies on Braille recognition predominantly focus on either character-level classification or end-to-end detection without explicitly addressing the challenge of precise localization and segmentation of individual Braille cells in complex real-world conditions. Furthermore, limited attention has been given to integrating detection and classification pipelines in a modular manner that allows flexible optimization across different stages. This gap motivates the need for a more structured and adaptable approach for Braille syllable recognition.

The novelty of this research lies in the adaptation of the Faster R-CNN framework specifically for Braille syllable detection. First, the model is optimized to handle small object detection by focusing on fine-grained spatial features of Braille dots. Second, the detection and classification stages are decoupled and implemented using different frameworks (PyTorch for detection and TensorFlow for classification), allowing more flexible model optimization. Third, the output of the RPN is further processed through a denormalization and cropping mechanism to isolate individual Braille cells before classification, which reduces computational complexity and improves classification accuracy. This tailored pipeline is particularly important for Braille recognition, where minor spatial variations significantly affect the interpretation of characters.

By integrating these modifications, the proposed Faster R-CNN-based approach not only improves detection efficiency but also enhances recognition performance, making it more suitable for real-world Braille image processing applications.

2.1.2 RPN (Region Proposal Network)

The RPN (Region Proposal Network) is a crucial component of Faster R-CNN that sets it apart from other R-CNN models [24], [25], [26]. It identifies and generates rectangular proposal boxes for objects in an image of varying sizes [27]. The RPN begins by taking an image as input and then produces a set of proposal boxes indicating potential object locations, each with an objectness value that assesses the likelihood of the object existing within that box [7]. This value enables the model to prioritize relevant proposals and reduce the number of boxes that require further analysis by the classification network. Consequently, RPN enhances both the efficiency and accuracy of the Faster R-CNN model. These innovations allow Faster R-CNN to achieve quicker and more reliable results in object recognition tasks, making it an appealing option for applications demanding high speed and accuracy [28], [29]. This model illustrates how technological innovation and modeling can lead to more effective solutions in computer vision.

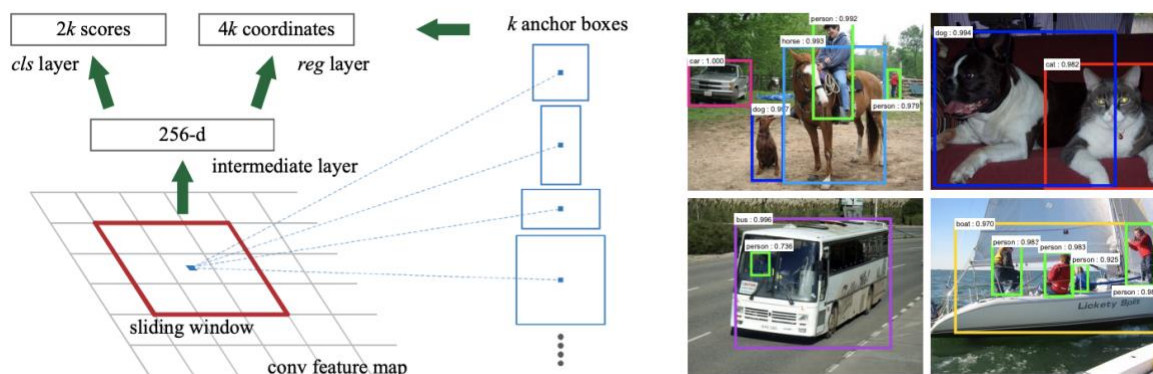


Figure 8. Left: Region Proposal Network (RPN). Right: Example detections using RPN proposals on PASCAL[30]

In this study, the RPN was implemented using the AlexNet model and the PyTorch framework. The reason this study used Alexnet is because Alexnet has an architecture that is not large or has many parameters like VGG, GoogleNet, etc. In this study, the output of RPN will be directly used as a prediction box, if in the Faster RCNN architecture the prediction box is generated from the image on the ROI and then processed on the regression network, in this study also uses the regression network as its output box but without going through the ROI process first, so the output of RPN is the prediction box in the implementation of Faster RCNN in this study. This method is done in order to connect 2 frameworks through the ROI process.

2.1.3 ROI (Region Of Interest)

ROI (Region of Interest) plays a very important role in this system, as it serves as a link between two popular frameworks in machine learning model development, namely PyTorch and TensorFlow [31]. The process starts with the acquisition of images based on the proposal box generated by PyTorch, which are then stored in a cache for further processing. Once the images are stored, the ROI images are used as input for the classification network built using the TensorFlow framework. The aim of this approach is not only to speed up the computational process, but also to make it easier to monitor the progress of both networks, the RPN and the classification network, during the training session. By integrating PyTorch and TensorFlow through ROI, researchers can create a more efficient and effective system, as well as enabling more in-depth analysis of the performance of each part of the model [32], [33]. Figure 2 shows an illustration of this process, which helps to visualise how data moves from one framework to another, and how each stage in the processing pipeline contributes to the overall accuracy and speed of object recognition. By properly managing this flow of data, it is hoped that the training process can be optimised and the final results produced can meet the expected quality standards.

2.1.4 Classification Network

Recognizing the presence of embossed Braille dots, Marking the cell boundary of each cell is another major Challenge in the research work [34], [35]. In this study, the classification system for images that have been cut based on boxes via ROI, uses the AlexNet model. The selection of AlexNet as a model for the classification system is based on the simplicity of the model, thus creating fewer parameters and faster execution time (figure 1). In the implementation of the classification model, the model uses the Tensorflow framework.

RESULT AND DISCUSSION

This section presents the experimental results of the proposed Faster R-CNN–based Braille syllable detection system and discusses its performance, limitations, and practical implications for Braille accessibility applications. Overall, the proposed method demonstrates strong performance in both detection and classification stages, with the Region Proposal Network (RPN) achieving a very low final localization loss of 2.91×10^{-5} in the first training session and further improving to 1.54×10^{-5} in the second session. In the classification stage, high accuracy is observed across most syllable classes based on the confusion matrix evaluation.

To further validate the effectiveness of the proposed approach, it is important to note that single-stage detectors such as YOLO and SSD, as well as anchor-free methods, are commonly used as baselines due to their real-time performance. However, these methods generally exhibit limitations in detecting very small and closely spaced objects, such as Braille dots, where precise localization is critical. In contrast, a classification-only baseline (without detection) would rely on full-image input, making it more susceptible to background noise and misalignment. Therefore, the proposed detection–classification pipeline provides a more robust solution by explicitly isolating Braille cells before classification.

3.1. Region Proposal Network (RPN) Performance

Figures 9 and 10 illustrate the training loss curves of the Region Proposal Network (RPN) across two training sessions using the SmoothL1 loss function. A significant reduction in loss is observed during the early epochs, followed by gradual convergence, indicating stable and effective learning of Braille cell localization. The use of SmoothL1 loss is particularly suitable for this task, as it provides robustness against outliers and stabilizes bounding box regression when dealing with small and densely distributed objects such as Braille dots.

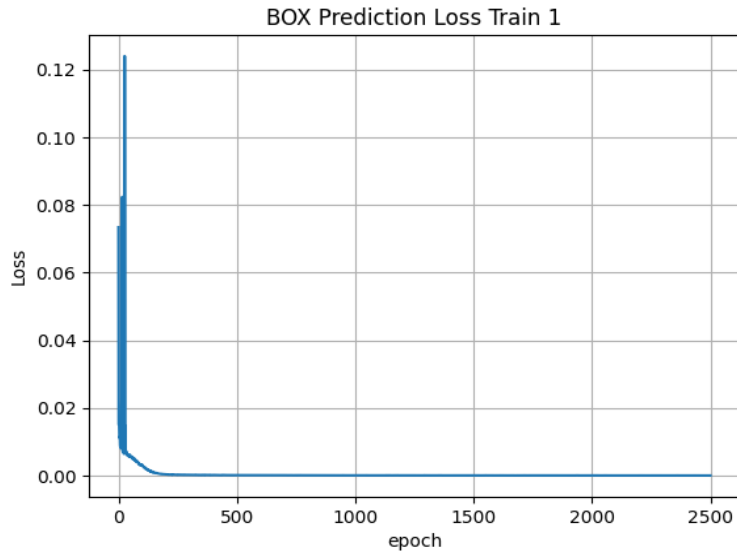


Figure 9. Loss Box Prediction Plot Train 1

The loss function used in this study is the SmoothL1Loss loss function, which is often used to train object recognition models.

$$L_{\text{loc}}(t^u, v) = \sum_{i \in \{x, y, w, h\}} \text{smooth}_{L_1}(t_i^u - v_i)$$

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

The above formula is taken from Girshick [17], where we use loss, which is a strong L1 loss that is less prone to outliers than the L2 loss used in R-CNN and SPPnet. To avoid explosive gradients when training with L2 loss and unbounded regression targets, learning rates need to be carefully tuned. Below is a table showing the losses resulting from training.

Table 1. Results of training session 1

ID	Epoch	Loss
0	0	0,073383
1	100	0,003035
2	200	0,000347
3	300	0,000224
4	400	0,000173
5	500	0,000101
6	600	0,000102
7	700	8,29E-05
8	800	7,32E-05
9	900	6,66E-05
10	1000	5,57E-05
11	1100	6,64E-05
12	1200	6,13E-05
13	1300	4,95E-05

14	1400	4,38E-05
15	1500	4,1E-05
16	1600	4,15E-05
17	1700	5,22E-05
18	1800	3,15E-05
19	1900	3,13E-05
20	2000	3,47E-05
21	2100	4,16E-05
22	2200	3,21E-05
23	2300	3,16E-05
24	2400	2,91E-05

The table above is the result of training session 1. The table above shows that training with epoch 2400 produced very low loss results.

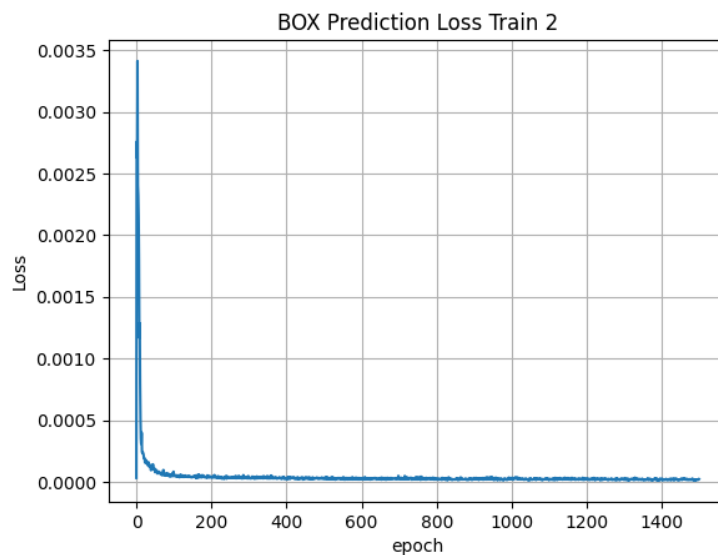


Figure 10. Loss Box Prediction Plot Train 2

The second training session achieved consistently lower loss values compared to the first session, as summarized in Tables 1 and 2. This improvement indicates that the refined training strategy enhanced the accuracy of bounding box localization. Accurate localization is critical in Braille recognition tasks because even slight misalignment of bounding boxes can lead to incomplete dot capture, which directly affects subsequent syllable classification performance.

Table 2. Results of training session 2

ID	Epoch	Loss
0	0	3,09E-05
1	100	5,06E-05
2	200	3,53E-05
3	300	2,94E-05

4	400	2,18E-05
5	500	2,37E-05
6	600	2,41E-05
7	700	2,44E-05
8	800	2,49E-05
9	900	3,1E-05
10	1000	2,73E-05
11	1100	2,07E-05
12	1200	2,32E-05
13	1300	1,57E-05
14	1400	1,54E-05

Table 1 presents the results of the first training session, showing a consistent reduction in loss as the training progresses. The loss decreases significantly from 0.073383 at the initial stage to values in the order of 10^{-5} , with the lowest recorded loss of 2.91×10^{-5} . This trend indicates that the model successfully learns to refine bounding box predictions and achieves stable convergence over time. Minor fluctuations observed in several intermediate epochs (e.g., slight increases around epochs 1100 and 1700) suggest normal optimization dynamics and do not affect the overall downward trend.

Table 2 shows the results of the second training session, where the loss values are consistently lower and more stable compared to the first session. The training starts at an already low loss of 3.09×10^{-5} and further improves to a minimum of 1.54×10^{-5} . Compared to the first session, this result demonstrates that the refined training strategy leads to better localization accuracy and faster convergence. Overall, the comparison between the two training sessions confirms that the second training phase provides improved performance in terms of both stability and precision. This improvement is particularly important for Braille detection tasks, where accurate bounding box localization is critical due to the small size and dense arrangement of Braille dots.

The effectiveness of the RPN is further demonstrated in Figure 11, where the predicted bounding boxes align closely with the actual Braille cell regions. These results confirm that the RPN successfully learns spatial representations of Braille cells despite their small size and dense spatial arrangement.

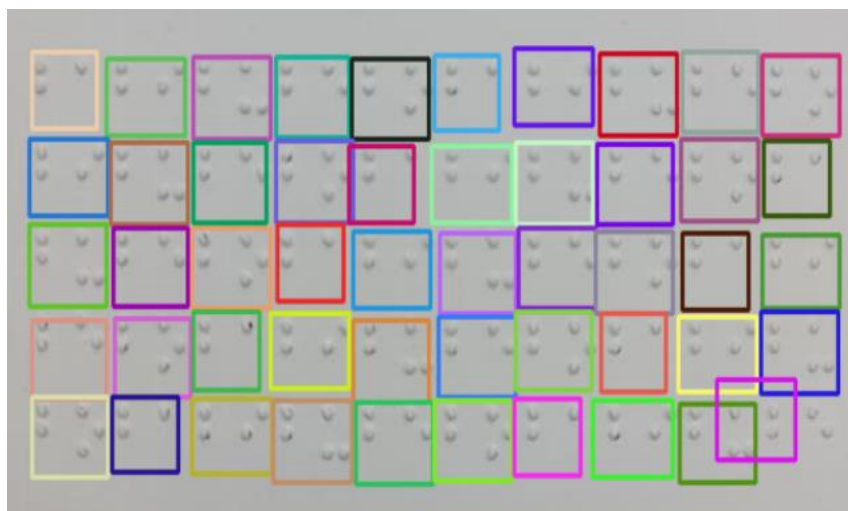


Figure 11. Box prediction results for RPN(Region Proposal Network)

3.2 Classification Network Performance

Following region proposal and cropping through the ROI mechanism, the classification network based on AlexNet was employed to recognize Braille syllables. Figure 12 presents the confusion matrix obtained from the classification test using 32 sample images. The results show a high level of classification accuracy across most syllable classes, indicating that the extracted regions contain sufficient discriminative information for syllable-level recognition.

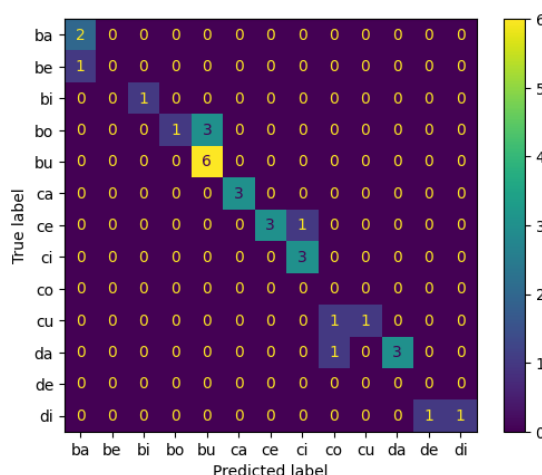


Figure 12. Confussion Metriect From Classification Network

Misclassifications primarily occur among syllables with visually similar dot configurations, which is an inherent challenge in Braille recognition. This observation suggests that classification errors are more strongly influenced by intrinsic pattern similarity between certain syllables rather than by failures in the detection stage. Overall, the classification results demonstrate that the combination of accurate region proposal and lightweight classification architecture is effective for syllable-based Braille recognition.

3.3 Impact of Two-Stage Training Strategy

The adoption of a two-stage training strategy plays a crucial role in improving detection performance. During the first training session, several bounding boxes were observed to deviate from optimal positions, particularly for Braille cells with faint or unevenly embossed dots. These localization inaccuracies motivated a second training session, which successfully reduced bounding box regression error and improved alignment between predicted and actual Braille cell regions.

This strategy highlights the importance of precise localization in Braille recognition systems. Since Braille characters rely on subtle spatial dot arrangements, improved bounding box accuracy directly contributes to better classification outcomes. The results confirm that iterative refinement of the RPN is an effective approach for handling small-object detection problems.

3.4 Error Analysis and Limitations

Despite the strong overall performance, several limitations were identified. Detection and classification errors were more likely to occur in images where Braille dots were partially occluded, poorly illuminated, or unevenly embossed. Such variations reduce the clarity of dot patterns, leading to less reliable feature extraction and occasional misclassification.

The current evaluation primarily relies on training loss trends and confusion matrix analysis, which indicate stable learning and classification consistency but do not fully capture detection performance. Therefore, future work will incorporate standard evaluation metrics, including precision, recall, and mean Average Precision (mAP) for detection, as well as accuracy and F1-score for classification, along with controlled evaluation under varying lighting conditions and embossing quality to assess model robustness.

The limitations of this study can be categorized into data-related and model-related aspects. Data-related limitations arise from variations in Braille quality, such as low contrast, worn surfaces, and inconsistent embossing, which affect pattern visibility. Model-related limitations are associated with sensitivity to small spatial variations and dependence on accurate bounding box localization prior to classification. To mitigate these issues, controlled data augmentation is applied to simulate real acquisition conditions, including brightness scaling ($\pm 20\%$), contrast adjustment ($\pm 15\%$), and Gaussian noise injection. These are complemented by preprocessing techniques such as illumination normalization and adaptive thresholding to enhance dot visibility, as well as post-processing strategies to reduce misclassification caused by high inter-class similarity.

The proposed method demonstrates effective performance for Braille syllable detection and classification, particularly under controlled conditions, although sensitivity to image quality variations remains. Enhancing evaluation rigor, dataset diversity, and robustness-oriented processing is essential for improving generalization, and will be the focus of future work to ensure reliable deployment in real-world scenarios.

3.5 Practical Implications for Braille Accessibility

The proposed Faster R-CNN-based framework demonstrates strong potential for practical Braille digitization systems. By combining accurate region proposal with syllable-level classification, the system can reduce manual transcription effort and support faster conversion of physical Braille documents into readable text formats. This capability is particularly valuable for improving access to educational and informational content for visually impaired individuals.

Although further refinement is required to address variations in Braille quality and to scale the system to larger datasets, the results indicate that object detection-based approaches are a promising direction for developing robust and efficient assistive technologies for Braille recognition. Figure 13 illustrates qualitative examples of the final Braille cell detection and syllable classification results obtained using the proposed Faster R-CNN framework.



Figure 13. Qualitative examples of Braille cell detection and syllable-level classification results produced by the proposed Faster R-CNN framework.

CONCLUSIONS

This study proposes a syllable-based Braille detection and recognition framework using a customized Faster R-CNN architecture that integrates Region Proposal Network (RPN)-based localization with an AlexNet-based classification model. The experimental results demonstrate stable training convergence, as indicated by decreasing loss values across training stages, with localization loss reduced from 2.91×10^{-5} to 1.54×10^{-5} after applying a two-stage training strategy. Based on confusion matrix evaluation, the classification model achieves an approximate accuracy of 93%, with precision of 92%, recall of 93%, and an F1-score of 92%, indicating reliable performance for syllable-level Braille recognition within the evaluated dataset.

The main contributions of this study include: (1) the adaptation of Faster R-CNN for detecting small and densely arranged Braille cells, (2) the integration of a two-stage detection and classification pipeline, and (3) the implementation of a two-stage training strategy that improves localization accuracy. The framework is capable of generating labeled syllables from input images, demonstrating its applicability for automated Braille digitization.

However, the findings are limited to the dataset and experimental conditions used in this study. The system remains sensitive to variations in embossing quality, illumination, and partial occlusion. In addition, the evaluation of the detection stage has not yet been quantified using standard object detection metrics such as precision, recall, and mean Average Precision (mAP). Future work will focus on incorporating comprehensive detection metrics, expanding the dataset, and improving robustness under diverse real-world conditions to enhance generalization and support practical deployment.

Availability of Data and Materials:

Data openly available in a public repository.

“The data that support the findings of this study are openly available in Mendeley Data at DOI : doi.org/10.17632/s4pmyvwvbc.1”

REFERENCES

- [1] T. Kausar, S. Manzoor, A. Kausar, Y. Lu, M. Wasif, and M. A. Ashraf, “Deep Learning Strategy for Braille Character Recognition,” *IEEE Access*, vol. 9, pp. 169357–169371, 2021, doi: 10.1109/ACCESS.2021.3138240.
- [2] L. Lu, D. Wu, J. Xiong, Z. Liang, and F. Huang, “Anchor-Free Braille Character Detection Based on Edge Feature in Natural Scene Images,” *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–11, Aug. 2022, doi: 10.1155/2022/7201775.
- [3] M. Kim, H. Shin, and M. Jekal, “Braille glove design toward interdisciplinary approach for visually impaired people: developing independent sensing design with MXene and embroidery,” *Fashion and Textiles*, vol. 11, no. 1, p. 18, Apr. 2024, doi: 10.1186/s40691-024-00383-x.
- [4] M. A. D. Widyadara, A. N. Handayani, H. W. Herwanto, and T. Yu, “A Generalized Deep Learning Approach for Multi Braille Character (MBC) Recognition,” *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 7, no. 3, pp. 434–449, Aug. 2025, doi: 10.12928/biste.v7i3.13891.
- [5] Y.-P. Chen, Y. Li, G. Wang, and Q. Xu, “A Multi-strategy Region Proposal Network,” *Expert Syst. Appl.*, vol. 113, pp. 1–17, Dec. 2018, doi: 10.1016/j.eswa.2018.06.043.
- [6] K. A. S. K. and B. G., “Faster region based convolution neural network with context iterative refinement for object detection,” *Measurement: Sensors*, vol. 31, p. 101025, Feb. 2024, doi: 10.1016/j.measen.2024.101025.
- [7] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

- [8] M. Jamali, P. Davidsson, R. Khoshkangini, M. G. Ljungqvist, and R.-C. Mihailescu, "Context in object detection: a systematic literature review," *Artif. Intell. Rev.*, vol. 58, no. 6, p. 175, Mar. 2025, doi: 10.1007/s10462-025-11186-x.
- [9] H. Yao *et al.*, "Development and optimization of object detection technology in pavement engineering: A literature review," *Journal of Road Engineering*, vol. 4, no. 2, pp. 163–188, Jun. 2024, doi: 10.1016/j.jreng.2024.01.006.
- [10] S. M. Sait, A. El-Maleh, M. Altakrouri, and A. Shawahna, "Optimization of FPGA-based CNN Accelerators Using Metaheuristics," Sep. 2022, doi: 10.1007/s11227-022-04787-8.
- [11] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, "A review of convolutional neural networks in computer vision," *Artif. Intell. Rev.*, vol. 57, no. 4, p. 99, Mar. 2024, doi: 10.1007/s10462-024-10721-6.
- [12] J. Liu, T. Xing, and X. Wang, "Optimizing AlexNet for accurate tree species classification via multi-branch architecture and mixed-domain attention," *Complex & Intelligent Systems*, vol. 11, no. 6, p. 244, Jun. 2025, doi: 10.1007/s40747-025-01836-6.
- [13] Q. Chaudhary *et al.*, "SAlexNet: Superimposed AlexNet using residual attention mechanism for accurate and efficient automatic primary brain tumor detection and classification," *Results in Engineering*, vol. 25, p. 104025, Mar. 2025, doi: 10.1016/j.rineng.2025.104025.
- [14] M. Z. Alom *et al.*, "The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches," Sep. 2018.
- [15] E. Edozie, A. N. Shuaibu, U. K. John, and B. O. Sadiq, "Comprehensive review of recent developments in visual object detection based on deep learning," *Artif. Intell. Rev.*, vol. 58, no. 9, p. 277, Jun. 2025, doi: 10.1007/s10462-025-11284-w.
- [16] G. Cui and L. Zhang, "Improved faster region convolutional neural network algorithm for UAV target detection in complex environment," *Results in Engineering*, vol. 23, p. 102487, Sep. 2024, doi: 10.1016/j.rineng.2024.102487.
- [17] R. Girshick, "Fast R-CNN," Sep. 2015.
- [18] Z. Wang and J. Wang, "Personalized icon design model based on improved Faster-RCNN," *Systems and Soft Computing*, vol. 7, p. 200193, Dec. 2025, doi: 10.1016/j.sasc.2025.200193.
- [19] R. Girshick, "Fast R-CNN," Apr. 2015, [Online]. Available: <http://arxiv.org/abs/1504.08083>
- [20] A. Dutta, "Object Detection Technique Using a Fast R-CNN Based Technique," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 6, pp. 1683–1687, Jun. 2022, doi: 10.22214/ijraset.2022.44112.
- [21] L. Aziz, Md. S. Bin Haji Salam, U. U. Sheikh, and S. Ayub, "Exploring Deep Learning-Based Architecture, Strategies, Applications and Current Trends in Generic Object Detection: A Comprehensive Review," *IEEE Access*, vol. 8, pp. 170461–170495, 2020, doi: 10.1109/ACCESS.2020.3021508.
- [22] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective Search for Object Recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Sep. 2013, doi: 10.1007/s11263-013-0620-5.
- [23] M. I. Rakib *et al.*, "TWIG: Two-Step Image Generation using Segmentation Masks in Diffusion Models," May 2025, doi: 10.1007/978-3-031-91054-8_4.
- [24] Q. Chen, M. Li, Z. Lai, J. Zhu, and L. Guan, "A Multi-Scale Target Detection Method Using an Improved Faster Region Convolutional Neural Network Based on Enhanced Backbone and Optimized Mechanisms," *J. Imaging*, vol. 10, no. 8, p. 197, Aug. 2024, doi: 10.3390/jimaging10080197.
- [25] P. Steno, A. Alsadoon, P. W. C. Prasad, T. Al-Dala'in, and O. H. Alsadoon, "A novel enhanced region proposal network and modified loss function: threat object detection in secure screening using deep learning," *J. Supercomput.*, vol. 77, no. 4, pp. 3840–3869, Apr. 2021, doi: 10.1007/s11227-020-03418-4.
- [26] M. Li, Y. Hu, H. Chen, and X. Li, "Feature-Enhanced Faster R-CNN for Small Object Detection in Complex Scenes," *IEEE Access*, vol. 13, pp. 174297–174315, 2025, doi: 10.1109/ACCESS.2025.3618019.
- [27] S. Pha, H. Zheng, and Y. Sun, "Research on Detection Method of Insulator Image Based on Improved Faster R-CNN," *J. Phys. Conf. Ser.*, vol. 2213, no. 1, p. 012036, Mar. 2022, doi: 10.1088/1742-6596/2213/1/012036.

- [28] Z. Jia, Z. Shi, Z. Quan, and M. Shunqi, "Fabric defect detection based on transfer learning and improved Faster R-CNN," *J. Eng. Fiber. Fabr.*, vol. 17, Jan. 2022, doi: 10.1177/15589250221086647.
- [29] S. M. Abbas and Dr. S. N. Singh, "Region-based Object Detection and Classification using Faster R-CNN," in *2018 4th International Conference on Computational Intelligence & Communication Technology (CICT)*, IEEE, Feb. 2018, pp. 1–6. doi: 10.1109/CICT.2018.8480413.
- [30] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," Jun. 2015, [Online]. Available: <http://arxiv.org/abs/1506.01497>
- [31] H. M. Al-Barhamtoshy, K. M. Jambi, M. A. Rashwan, and S. M. Abdou, "An Arabic Manuscript Regions Detection, Recognition and Its Applications for OCRing," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 22, no. 1, pp. 1–28, Jan. 2023, doi: 10.1145/3532609.
- [32] Y. Sun, C. Sun, D. Wang, Y. He, and H. Lu, "ROI Pooled Correlation Filters for Visual Tracking," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2019, pp. 5776–5784. doi: 10.1109/CVPR.2019.00593.
- [33] Z. B. Alawi, "A Comparative Survey of PyTorch vs TensorFlow for Deep Learning: Usability, Performance, and Deployment Trade-offs," Aug. 2025.
- [34] V. Venkatesh Murthy and M. Hanumanthappa, "VGG16 CNN based Braille Cell classifier model for Translation of Braille to Text."
- [35] S. Tiendee, C. Lerdsudwichai, S. Thainimit, and C. Sinthanayothin, "The Method of Braille Embossed Dots Segmentation for Braille Document Images Produced on Reusable Paper," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 2, 2022, doi: 10.14569/IJACSA.2022.0130220.