Deep Learning-based OCR for Seven Segment Displays in Laboratory Instruments

Juan Miguel Z. Marqueses
Department of Industrial Systems Engineering,
Asian Institute of Technology,
Pathum Thani, Thailand
st125144@ait.asia

Yasiru Fernando
Department of Indus Engineering and Technology,
Asian Institute of Technology,
Pathum Thani, Thailand
st125391@ait.asia

***Abstract*—Optical Character Recognition (OCR) has become a vital tool for automating text recognition tasks across diverse applications. In laboratory settings, the manual recording of instrument readings is a time-consuming, error-prone process, especially when monitoring multiple devices over extended periods. This study addresses these challenges by developing a deep learning-based application that integrates Object Detection and OCR for real-time data collection from laboratory instruments. The system employs YOLO for instrument identification and a character-based OCR approach for extracting readings, including fine details such as decimal points on seven-segment displays. Through experimentation, YOLOv11n was identified as the most effective model for object detection, while YOLOv11x demonstrated superior performance in OCR tasks using character-level annotation. The models were optimized using Ray Tune for hyperparameter tuning, achieving high precision and recall scores. The proposed solution eliminates the need for manual intervention, significantly reducing human error and enhancing efficiency in laboratory workflows. This automated approach offers a scalable and accurate method for real-time data acquisition, with potential for broader applications in similar domains.**

***Keywords*— OCR, Seven Segment, Deep Learning, Object Detection.**

I. Introduction

Optical Character Recognition (OCR) is a computer vision application that interprets handwritten or typewritten text and converts it into machine-readable formats. OCR has proven to be invaluable across various domains, including industrial applications [1], [2] and the healthcare sector [3], where it automates data entry and enhances efficiency.

In laboratory settings, experiments often require monitoring readings from multiple instruments that measure different physical quantities. A common challenge in such scenarios is the difficulty of manually recording instrument readings over extended periods [4]. Traditional methods involve video capture and the manual transcription of readings at specified intervals, which is time-consuming, resource-intensive, and prone to human error due to the repetitive and tedious nature of the task.

This study addresses these challenges by leveraging deep learning techniques to develop an application that integrates Object Detection and Optical Character Recognition. The application is designed to identify and classify different types of laboratory instruments and extract their readings in real-time, enabling automated data collection from multiple instruments over a specified time span. By employing deep learning-based object detection (YOLO) for instrument identification and OCR for reading extraction, the system eliminates the need for manual intervention, reducing error rates and improving efficiency in laboratory workflows.

The rest of this paper is organized as follows: Section 2 provides a comprehensive review of related literature. Section 3 details the methodology used in developing the application. Section 4 presents and discusses the results, while the final section concludes the paper with findings and suggestions for future improvements.

II. Literature Review

Traditional Optical Character Recognition (OCR) techniques often rely on image processing methods followed by template matching or feature comparison to identify characters [5], [6], [7]. For instance, Syafeeza et al. experimented with MATLAB-based methods, including Otsu's method (automatic image thresholding), erosion-based morphology, and binary thresholding, to preprocess Seven Segment LED displays [5]. They found that thresholding was the most effective preprocessing method for MATLAB's OCR. Similarly, Kanagarathinam et al. employed the Maximally Stable External Regions (MSER) algorithm for preprocessing, enabling accurate OCR with MATLAB [7]. Ghugardare et al. adopted a feature extraction approach, matching extracted features with stored vectors to identify characters based on maximum correlation with ASCII values [6].

Recent advancements in OCR leverage deep learning to achieve higher accuracy and robustness. Low et al. tested various pre-trained deep learning-based OCR models using their dataset, evaluating tools such as PARSeq, PaddleOCR, and MMOCR [8]. They found that DBNet from PaddleOCR was the best to detect text on seven segment displays. However, PARSeq model had the highest accuracy in recognizing the characters. Due to the pretrained nature of these models, the performance was up to standard.

Other researchers have explored specialized deep learning architectures for reading seven-segment displays. Wannachai et al. used Convolutional Neural Networks (CNNs) to process machine statuses displayed on seven-segment LEDs [9]. Their encoder-decoder-based CNN segmented the display region, followed by a high-precision CNN for digit recognition, achieving exceptional accuracy.

Similarly, Bonacic et al. employed multilayer feedforward neural networks, optimizing their architecture with a genetic algorithm to identify seven-segment digits. They further enhanced performance using ensemble averaging to combine multiple neural networks for improved reliability [10].

III. METHODOLOGY

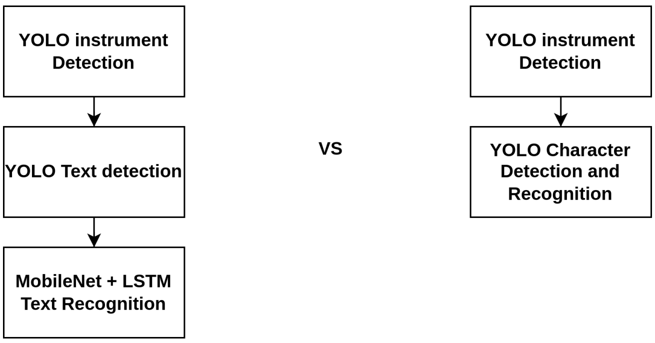

Fig. 1. Two models for OCR

We performed our application by first training an object detection model followed by training an OCR. Thereafter, both models were combined in a single application for the overall function.

A. Dataset

The dataset was sourced locally from 11 different laboratory instruments as shown in figure 1. These instruments were captured in different operating environments.

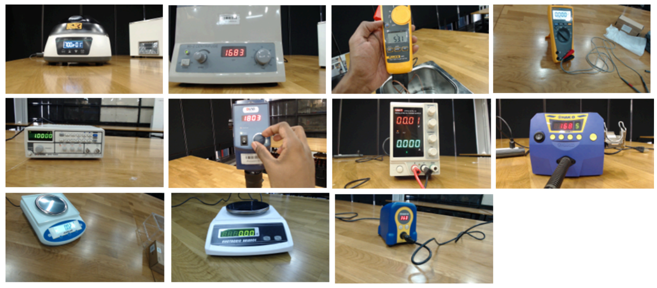

Fig. 2. Instruments in the dataset

The dataset provided images for both object detection and ocr parts of the application. For the object detection, 100 images each were sampled for each class, amounting to 1100 total images. These images were imported into Label-Studio labelling tool and labelled according to their relevant class. Thereby the dataset was exported in YOLO format.

B. Object Detection

In order to implement Object Detection to identify the specific type of laboratory instruments, three different versions of lightweight YOLO were trained on the local dataset. The results from YOLO training are given in Table 1.

TABLE I Comparison of YOLO Performance for object detection

Model	**mAP50**	**mAP50-95**
YOLOv8n	0.995	0.835
YOLOv10n	0.967	0.807
YOLOv11n	0.995	0.827

Although the mAP50-95 metric for v8 is slightly better than v11, v11 provided better inference results. Therefore, YOLOv11n was selected to be the object detection model.

C. Optical Character Recognition (OCR)

Before initiating the OCR training process, we conducted research to determine the most effective method for annotating the dataset we had collected. Among the various approaches available, two widely adopted techniques stood out: character-based annotation and word-based annotation. In our experiment, we applied both methods, utilizing over 700 annotations for character-based annotation and more than 1,900 annotations for word-based annotation.

Since the annotation methods differ, we employed two distinct model architectures: YOLOv11x for character-based annotation and YOLOv11x+MobileNetV3+LSTM for word-based annotation. The detailed architectures of these models are illustrated in Figure 3 and Figure 4.

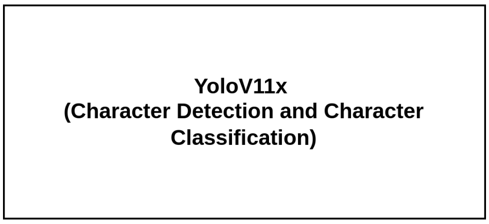

Fig. 3. Single YOLO Architecture

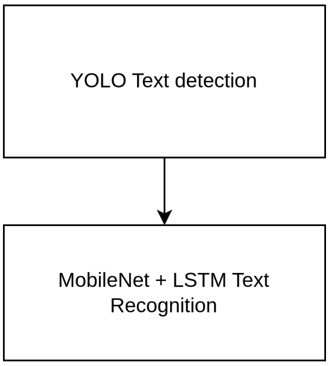

Fig. 4. YOLO+MobilNet+LSTM Architecture

To achieve optimal accuracy when training with the single YOLO architecture, we leveraged the built-in hyperparameter tuning capabilities of the Ultralytics library. Additionally, the MobileNet+LSTM architecture, integrated with a Connectionist Temporal Classification (CTC) network, was trained using the Python library **MLTU**, ensuring efficient and robust model performance.

For dataset annotation, we used Label Studio, an open-source data labeling tool that supports the annotation of text, images, videos, and more. It also facilitates exporting annotations in various formats, including YOLO, Pascal VOC XML, and JSON.

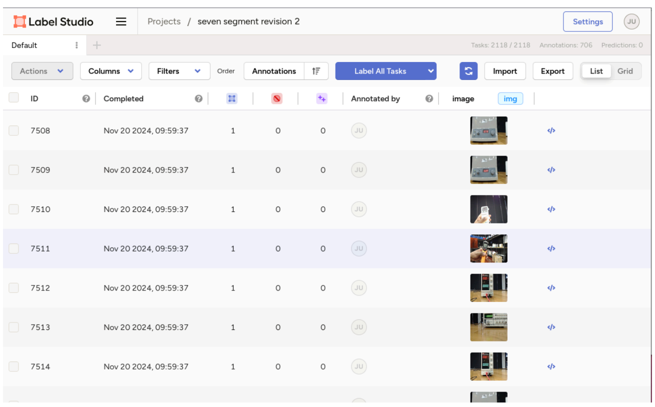

Fig. 5. Label Studio GUI

We utilized PyTorch as the deep learning framework for the OCR task due to its robust capabilities for research-oriented model training. PyTorch also enabled seamless integration with Ultralytics, allowing us to leverage a range of YOLO pretrained models, spanning versions V8 to V11.

IV. Experimental Results

B. *Optical Character Recognition (OCR)*

The experiment began with a comparison between the single YOLO and the YOLO+MobileNet+LSTM architecture. The results indicated that word-based annotation did not perform as effectively as character-based annotation in real-time applications. One notable limitation of word-based annotation is its reliance on clear and well-defined visuals, which hampers its ability to detect finer details, such as decimal points—an essential requirement for accurately recognizing values displayed on the instrument's screen.

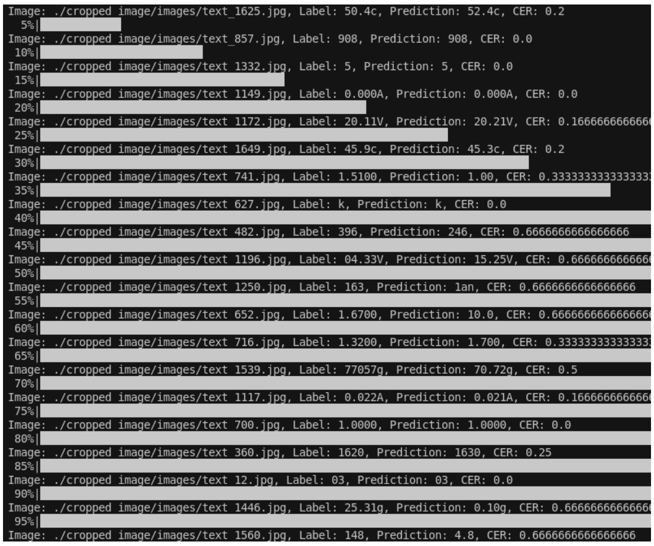

Fig. 6. Experiment on MobilenetV3+LSTM

Consequently, we focused our experiments on YOLOv11x, as character-based annotation demonstrated a significant advantage in detecting seven-segment displays in real-time applications, including capturing fine details like decimal points. Subsequently, we compared YOLOv8x, YOLOv10x, and YOLOv11x using the same hyperparameters optimized for YOLOv11x to evaluate which version delivered the best performance in detecting seven-segment displays on measuring instruments.

We began by fine-tuning the hyperparameters of YOLOv11x, one of the top-performing models from Ultralytics. Utilizing its built-in tuning capabilities with Ray Tune, we achieved optimal results, as shown in Figure 7.

The hyperparameters provided for tuning included learning rate, momentum, optimizer, dropout, warmup epochs, weight decay, hue augmentation, brightness augmentation, saturation augmentation, vertical and horizontal flips, copy-paste augmentation, erasing, and mixup.

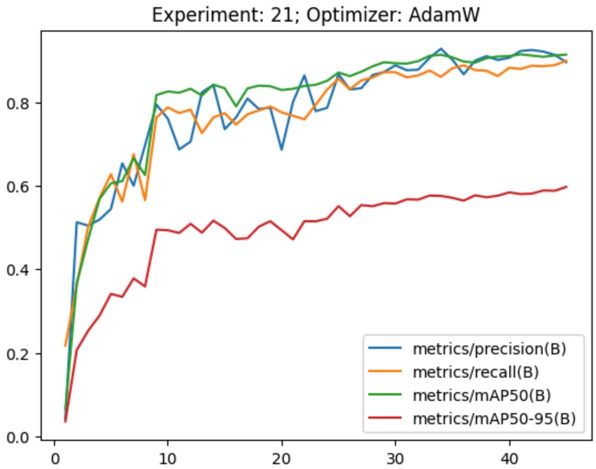

Fig. 7. Best Result from Ray-Tuner

Using the tuned hyperparameters, we applied them across YOLO versions 8x to 11x for further evaluation. As shown in the results below, the mean average precisions (mAP) were comparable, with YOLOv11x achieving the highest performance. However, as highlighted in Table 1, the YOLOv10x model exhibited a slightly lower mAP than its predecessor. This discrepancy is attributed to a reduction in image size (imgsz) for YOLOv10x, which was adjusted to 476 due to its higher computational demands on the GPU, whereas YOLOv8x and YOLOv11x maintained an imgsz of 600.

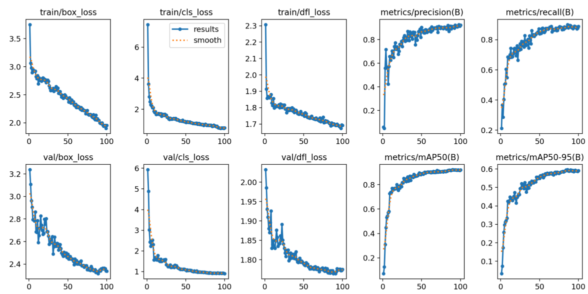

Fig. 8. Training result of YOLOV11x

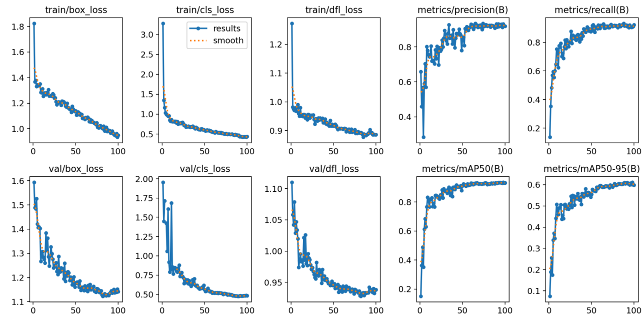

Fig. 9. Training result of YOLOV10x

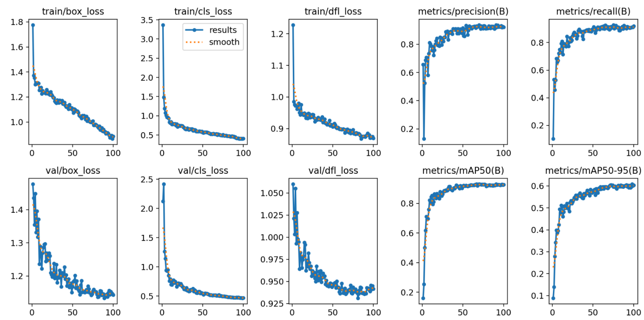

Fig. 10. Training result of YOLOV8x

TABLE II Metrics score comparison with V8x to V11x

Yolo Ver.	mAP@50	mAP@50-95	Precision	Recall
YOLOV11x	0.93545	0.61142	0.93114	0.90828
YOLOV10x	0.91714	0.59591	0.90066	0.90084
YOLOV8x	0.92792	0.60825	0.92014	0.91148

V. CONCLUSION

This study presents a robust system that combines Object Detection and Optical Character Recognition (OCR) to automate the process of reading and recording data from laboratory instruments. By leveraging YOLO-based object detection for instrument identification and a character-level OCR approach for reading extraction, the proposed application achieves high precision and efficiency in real-time data acquisition.

Key findings include:

1. **Object Detection:** Among the YOLO variants tested, YOLOv11n emerged as the optimal model for instrument identification, offering superior inference results despite comparable mAP50-95 metrics with YOLOv8n.
2. **OCR Analysis:** Character-based annotation proved to be significantly more effective than word-based annotation for real-time applications, particularly in detecting fine details such as decimal points on seven-segment displays. YOLOv11x demonstrated the highest performance in character recognition tasks, outperforming both YOLOv8x and YOLOv10x.
3. **Performance Optimization:** The use of hyperparameter tuning with Ray Tune enabled the YOLOv11x model to achieve its best performance metrics, with high precision and recall scores, further enhancing its real-world applicability.

This integrated approach overcomes the constraints of traditional manual and video-based methods, delivering a scalable and automated solution that minimizes human error and enhances workflow efficiency in laboratory environments. Future research could aim to expand the system's versatility by accommodating a broader range of instruments and operational contexts while investigating advanced neural architectures to further improve OCR precision and robustness. Additionally, the development of a comprehensive application to encapsulate the system could streamline model training and inference for enhanced detection capabilities. Ultimately, integrating the solution into embedded systems would facilitate its adoption in industrial environments, aligning with the principles of Industry 4.0 and paving the way for advancements toward Industry 5.0.

REFERENCES

[1] Y. Li and H. Qian, "Automatic Recognition System for Numeric Characters on Ammeter Dial Plate," in 2008 The 9th International Conference for Young Computer Scientists, Hunan, China: IEEE, Nov. 2008, pp. 913–918. doi: 10.1109/ICYCS.2008.498.

[2] V. E. Bugayong, J. Flores Villaverde, and N. B. Linsangan, "Google Tesseract: Optical Character Recognition (OCR) on HDD / SSD Labels Using Machine Vision," in 2022 14th International Conference on Computer and Automation Engineering (ICCAE), Brisbane, Australia: IEEE, Mar. 2022, pp. 56–60. doi: 10.1109/ICCAE55086.2022.9762440.

[3] E. Finnegan, M. Villarroel, C. Velardo, and L. Tarassenko, "Automated method for detecting and reading seven-segment digits from images of blood glucose metres and blood pressure monitors," Journal of Medical Engineering & Technology, vol. 43, no. 6, pp. 341–355, Aug. 2019, doi: 10.1080/03091902.2019.1673844.

[4] Y. Abdelrahman, M. El-Salamony, and M. Khalifa, "Visual Data Acquisition for Measuring Devices using Deep Learning," in 2021 3rd Novel Intelligent and Leading Emerging Sciences Conference (NILES), Giza, Egypt: IEEE, Oct. 2021, pp. 301–304. doi: 10.1109/NILES53778.2021.9600549.

[5] A. R. Syafeeza, N. A. Hamid, Y. C. Wong, M. F. Ramli, and Z. M. Noh, "Smart Data Recognition System for Seven Segment LED Display".

[6] R. P. Ghugardare, S. P. Narote, P. Mukherji, and P. M. Kulkarni, "Optical character recognition system for seven segment display images of measuring instruments," in TENCON 2009 - 2009 IEEE Region 10 Conference, Singapore: IEEE, Nov. 2009, pp. 1–6. doi: 10.1109/TENCON.2009.5395994.

[7] K. Kanagarathinam and K. Sekar, "Text detection and recognition in raw image dataset of seven segment digital energy meter display," Energy Reports, vol. 5, pp. 842–852, Nov. 2019, doi: 10.1016/j.egyr.2019.07.004.

[8] L. M. Low, F. H. Mohd Salleh, Y. F. Law, and N. Z. Zakaria, "Detecting and recognizing seven segment digits using a deep learning approach," ITM Web Conf., vol. 63, p. 01007, 2024, doi: 10.1051/itmconf/20246301007.

[9] A. Wannachai, W. Boonyung, and P. Champrasert, "Real-Time Seven Segment Display Detection and Recognition Online System Using CNN," in Bio-inspired Information and Communication Technologies, vol. 329, Y. Chen, T. Nakano, L. Lin, M. U. Mahfuz, and W. Guo, Eds., in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 329. , Cham: Springer International Publishing, 2020, pp. 52–67. doi: 10.1007/978-3-030-57115-3_5.

[10] I. Bonačić, T. Herman, T. Krznar, E. Mangić, G. Molnar, and M. Čupić, "Optical Character Recognition of Seven-segment Display Digits Using Neural Networks".