A YOLOv10-Based System for Automated FPT University Student Card Information Extraction

Toan Quy Nguyen *, Tuan Minh Le, Linh Van Nguyen, Phuc Hoang Minh Lai, Khang Vo Hoang Nguyen, Thuan Van Tran, and Thu Vo Minh Le

Department of Information Technology, Faculty of Computer Science and Engineering, FPT University, Ho Chi Minh City, Vietnam
Email: toannqse182785@fpt.edu.vn (T.Q.N.); tuanlmse184475@fpt.edu.vn (T.M.L.); linhnvse184598@fpt.edu.vn (L.V.N.); phuclhmse184914@fpt.edu.vn (P.H.M.L.); khangnvhse184634@fpt.edu.vn (K.V.H.N.); thuantvse184562@fpt.edu.vn (T.V.T.); thulvm@fpt.edu.vn (T.V.M.L.)
*Correspondence author

***Abstract*—This paper presents a model for automated recognition and information extraction from student Identification cards (ID card). The model utilizes the You Only Look Once Version 10 (YOLOv10) model to identify student ID cards. Subsequently, the detected card is rotated to its correct orientation, and any unwanted black regions are removed using Tesseract. Precise cropping of the student ID card is then achieved by identifying the intersection points on its four edges. The cropped card is subsequently standardized to a resolution of 640×440 pixels, and the region containing crucial information (name, student ID, and photo) is extracted and processed using Optical Character Recognition (OCR). This enables the efficient extraction and recognition of the necessary data from the student ID card. Following the research and model training process, the model's accuracy reached 94.6%, leaving us highly satisfied with the research outcomes. In the future, this model holds the potential for widespread adoption by numerous schools to facilitate easier student management within the institution.**

***Keywords*—detect student ID card, YOLOv10, computer vision, Optical Character Recognition (OCR), Tesseract**

I. Introduction

In the digital age, the application of automation methods in information management is becoming increasingly crucial. Student ID cards, a vital management tool in universities and colleges, contain a wealth of essential personal information. Automatically recognizing and extracting information from student ID cards will help simplify and enhance the efficiency of various management activities, from attendance, library borrowing, and returning to other administrative procedures. However, the rapid advancement of computer vision, especially in the field of Optical Character Recognition (OCR), offers a promising solution to automate this process. This research delves into the development of a sophisticated computer vision model specifically designed to extract information from student ID cards, streamlining administrative workflows within educational institutions. This model leverages advanced computer vision techniques combined with the powerful Tesseract OCR engine. Tesseract, a widely adopted open-source OCR platform, has benefited significantly from contributions by researchers like Christian Clausner at Google AI. The effectiveness of the model was evaluated on over 1800 images, including 900 FPT University student ID cards captured at various angles and lighting conditions to enhance the model's accuracy. Experimental results demonstrate the system's ability to detect student ID cards with exceptional accuracy, exceeding 94%. The rotation process ensures minimal angle deviation, typically less than 1 degree, while OCR achieves 95% accuracy in information extraction. This robust approach provides a reliable and efficient solution for automated student ID card processing, with potential applications spanning various real-world scenarios within educational institutions. This includes tracking student attendance, library management, and administrative procedures in academic settings.

In the future, this model has the potential for widespread adoption in school systems such as automated attendance, and library book lending management. Applying this model can help students and faculty minimize the time spent on manual attendance recording in class or expedite the borrowing and returning of library materials.

II. Literature Review

In the realm of object recognition and image classification, advances in methodologies such as YOLOv10, Data Augmentation, and Tesseract OCR have provided significant enhancements in accuracy and processing efficiency. Each approach contributes uniquely, fostering the development of more robust and versatile systems suited for a variety of applications.

A. YOLOv10

YOLOv10, the latest iteration of the YOLO architecture, has gained recognition for its remarkable real-time object detection capabilities. This version

Manuscript received November 17, 2024; revised December 25, 2024; accepted March 12, 2025; published June 5, 2025.

incorporates advanced techniques, including Feature Pyramid Networks (FPN) and Path Aggregation Networks (PANet), which enable improved detection of small objects while ensuring optimal performance [1]. By striking a balance between speed and accuracy, YOLOv10 is an ideal solution for various applications, such as security surveillance and industrial automation [1].

B. Data Augmentation

Data Augmentation is pivotal for enhancing the performance of deep learning models, particularly in the context of object detection. By generating variations through techniques such as rotation, flipping, cropping, and color adjustments, the dataset is expanded, which helps mitigate overfitting and enhances the model's ability to generalize [1]. This practice is instrumental in improving the performance of networks like YOLO, as it offers a richer and more diverse set of training samples [1].

C. Tesseract OCR

Tesseract OCR is a prominent Optical Character Recognition (OCR) tool widely used for tasks involving the conversion of image-based text into digital format. With continuous development and integration with modern frameworks like TensorFlow, Tesseract now offers precise handling of complex text and supports multiple languages [1]. Integrating Tesseract OCR into object detection or image analysis systems like YOLO enhances recognition capabilities, not only for images but also for text embedded within images, offering a more comprehensive solution for applications such as document or receipt recognition.

III. Materials and Methods

A. Method for Detecting and Classifying Student ID Cards Using YOLOv10

The proposed system utilizes the YOLOv10 model trained on an extensive dataset of student ID cards. This training equips the model with the capability to accurately locate and classify student ID cards in both images and video streams, making it suitable for various real-time applications. YOLOv10's advanced architecture, which includes features like NMS-free training and dual-head detection, greatly enhances its efficiency and performance [2].

1) Advanced model components

The architecture of YOLOv10 is built upon an enhanced version of CSPNet, which improves gradient flow and reduces computational redundancy [3]. Additionally, it incorporates a Path Aggregation Network (PAN) for effective multi-scale feature fusion [3]. We focus on processing speed while maintaining accuracy so we choose YOLOv10-N for object detection, particularly for use cases such as student card identification, which offers several advantages rooted in its specific design and performance capabilities. This compact model enhances speed and accuracy and caters to the computational limitations often associated with real-time applications. Utilizing pre-trained weights considerably shortens the training time required to achieve optimal performance. Since YOLOv10-N has already been trained on a diverse dataset like COCO, it has developed a foundational understanding of various image features, significantly reducing the time needed to converge when trained on the new student ID dataset [4].

2) Image processing pipeline for Student ID card isolation

We process individual student images. Each image, with dimensions 640×640, denoted as I contains $m + n$ objects, where $n = 1$ represents the student ID card, and m represents other objects (where m is typically 1 in this controlled scenario, indicating a single non-ID object). These objects are defined as $\{x_i, v\}$, where x_i denotes a non-ID object, and v denotes the student ID card. Our objective is to detect the student ID card (v) within the image.

Each detected student ID card is identified using a bounding box (x_b, y_b, w_b, h_b), where (x_b, y_b) represents the top-left corner coordinates, and (w_b, h_b) represents the width and height of the box, respectively. The model is trained on images with labeled bounding boxes corresponding to the student ID cards. After training, the model can predict the bounding box around the student ID card in a new image I.

To extract the student ID card from the image, we use the predicted bounding box coordinates. Given the image I and the predicted bounding box (x_b, y_b, w_b, h_b), the cropped ID card image, I_c, is obtained as follows:

$$I_c = I\,[y_b : y_b + h_b, x_b : x_b + w_b] \tag{1}$$

This operation extracts the rectangular region defined by the bounding box from the original image (Fig. 1), effectively isolating the student ID card. This cropped image I_c can then be used for further processing.

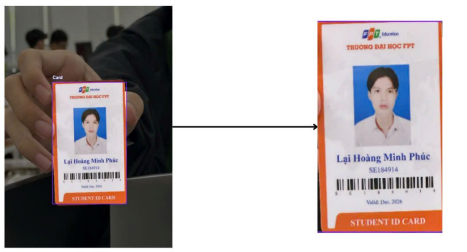

Fig. 1. ID card detection and cropping.

B. Optimizing Image Rotation with Orientation and Script Detection (OSD)

Correctly orienting a student ID card image is a crucial preprocessing step that significantly enhances the accuracy of subsequent information extraction. Instead of rotating the entire original image, an optimized method can be proposed, leveraging the skew angle provided by .image_to_osd alongside text-region image processing techniques. This optimized approach minimizes unnecessary image alterations while ensuring that the critical data remains accessible for accurate processing.

Orientation and Script Detection (OSD) is a notable feature of Tesseract OCR that identifies the orientation of

text within an image, accounting for any rotation or skew present in the layout. By determining the correct orientation, OSD allows the OCR engine to adjust the image accordingly, thereby improving its effectiveness in reading the text accurately [5]. This capability proves particularly beneficial when addressing documents and images where text may appear at various angles due to the method of capture.

The core functionality of OSD includes the analysis and classification of orientation angles, ranging from 0 to 180 degrees. Tesseract utilizes this analysis to correct the image alignment, ultimately ensuring that the text is presented in a horizontally readable format [6]. Additionally, the script detection component of OSD enables Tesseract to recognize different writing systems, which is vital for multilingual document processing [6].

1) Grayscale conversion

The image processing workflow initiates with the conversion of the original student ID card image into grayscale. This step is pivotal as it helps minimize unnecessary noise associated with colors, thereby enhancing the efficiency of text detection [7]. Following the grayscale conversion, the area containing text is identified and enlarged. By enlarging the text region before calculating the rotation angle, the approach notably enhances the accuracy of text detection while simultaneously reducing the impact of irrelevant areas in the image [7].

2) Skew angle detection and adjustment

Next, we utilize pytesseract.image_to_osd to analyze the grayscale image and detect the skew angle of the text (the 'rotate' value). Once the skew angle is identified, the system returns the current orientation of the text. To correct the image orientation, we rotate the image by subtracting the detected skew angle from the current image angle.

The rotation adjustment formula is:

$$\text{Required Rotation} = \text{Current Angle} - \text{Detected Skew Angle} \quad (2)$$

After applying this correction (Fig. 2), the image is rotated back to a horizontal alignment, ensuring the text is displayed in the correct orientation.

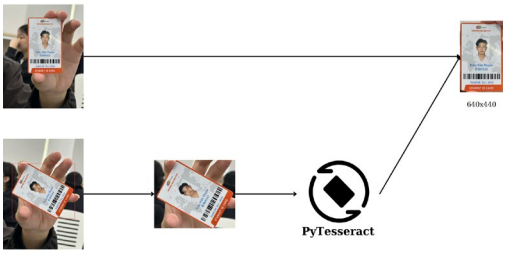

Fig. 2. ID card with rotation correction using PyTesseract.

3) Results and benefits

This method achieves three primary objectives:

- **Improved performance**: Enlarging the text region before skew angle detection speeds up the process, as only a smaller portion of the image is analyzed instead of the entire image.
- **Enhanced accuracy**: Rotating the enlarged text region improves the precision of both text detection and skew correction.
- **Simplified data processing**: Grayscale conversion and focusing on the text-containing region reduce the amount of data to be analyzed, optimizing the overall system performance.

This structured approach not only enhances the efficiency of text recognition tasks but also contributes to more reliable results in processing images of student ID cards and similar documents.

C. Text Area Detection

Following preprocessing, the next phase of our investigation involves accurately locating and segmenting regions of interest within standardized student ID card images. Specifically, we focus on the extraction of three essential information fields: student portrait, full name, and identification number. The precise delineation of these regions is paramount for subsequent information retrieval and automated processing.

1) Standardization of image dimensions

First, all student ID card images are resized to a resolution of 640×440 pixels (height × width) to standardize input dimensions and ensure consistent performance during subsequent Optical Character Recognition (OCR).

Ensuring that images are consistently resized contributes to the overall reliability and accuracy of the OCR process. Standardizing the input dimensions helps in achieving better clarity and recognition of characters, thereby enhancing the system's performance [8]. Consequently, this methodology aids not only in accuracy but also in achieving uniformity across different images processed by the OCR system [8].

2) Manual region definition

The manual region definition process is facilitated by the consistent layout of FPT University student ID cards (Fig. 3). Textual information is reliably located beneath the portrait, establishing a fixed structure that enhances the reliability and efficiency of information extraction. This consistency minimizes the need for complex algorithms to accommodate varying layouts, enabling straightforward implementation.

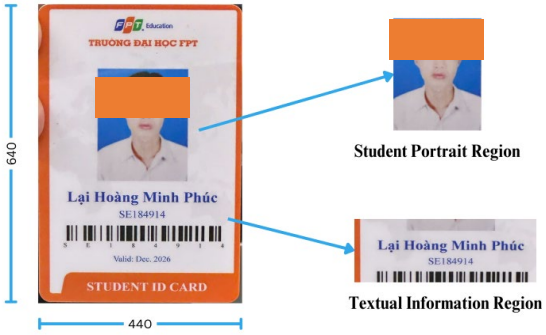

Fig. 3. ID card cropping.

By defining regions through specific pixel coordinates (Table I), this method simplifies processing and improves accuracy. This precise definition allows processing algorithms to operate efficiently, focusing on the designated areas without unnecessarily scanning the entire image.

TABLE I. REGION COORDINATES ON STUDENT ID CARD

	x-min	y-min	x-max	y-max
Student Portrait Region	130	130	320	370
Textual Information Region	10	350	430	500

This precise definition allows the processing algorithms to operate efficiently, focusing on the specified areas without unnecessarily scanning the entire image.

D. Preprocessing for Enhanced OCR Accuracy

After isolating the Regions of Interest (ROIs)—namely, the student portrait and the textual information area—these regions are extracted from the original ID card image for further processing. Image quality can vary considerably due to factors inherent in the acquisition process, such as blur, noise, and uneven illumination. Therefore, a series of preprocessing steps is essential for enhancing the quality of the extracted information and improving the performance of downstream tasks such as Optical Character Recognition (OCR). These preprocessing steps are detailed below:

1) Grayscaling

The Region of Interest (ROI) is converted to grayscale to reduce dimensionality while preserving essential structural information. This conversion simplifies subsequent processing, enhancing the efficiency of various image processing algorithms. The grayscale transformation is typically achieved using a weighted sum of the red (R), green (G), and blue (B) color channels, represented by the equation: $Y = 0.2126R + 0.7152G + 0.0722B$, where Y represents luminance [9, 10]. The coefficients are derived from the human visual system's sensitivity to different wavelengths, with green receiving the highest weight due to its greater perceived brightness [9, 10].

2) Blurring

A Gaussian blur is applied to the grayscale image to attenuate noise and smooth high-frequency details. This operation effectively reduces the impact of small artifacts or imperfections that might interfere with character recognition [11, 12]. The degree of smoothing is controlled by the Gaussian kernel size. While larger kernels provide more aggressive smoothing, they also risk blurring critical edges; thus, careful selection of this parameter is essential.

3) Sharpening

After blurring, a sharpening technique, such as unsharp masking (Fig. 4), is employed to enhance edge definition and improve character clarity within the text region. This sharpening step compensates for the blurring introduced previously and accentuates fine details, thus improving the accuracy of subsequent Optical Character Recognition (OCR). Specifically, a 3×3 sharpening kernel, defined as np.array([[−1, −1, −1], [−1, 9, −1], [−1, −1, −1]]), is applied using the cv2.filter2D function. The −1 argument for depth ensures the output image has the same depth as the input image, while the kernel enhances edges by increasing the contrast between neighboring pixels [13].

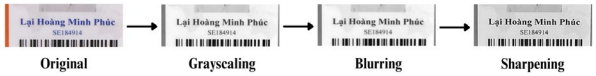

Fig. 4. Image preprocessing pipeline.

E. Text Extraction from Student ID Cards Using Optimized Tesseract OCR

After evaluating various Vietnamese-capable OCR models, including VietOCR, we found that VietOCR's limitation in recognizing text beyond a single line did not meet the requirements of our multi-line student ID card data extraction. This shortcoming led us to select Tesseract OCR for its superior performance, particularly its ability to recognize multi-line Vietnamese text accurately. Tesseract's integration of Long Short-Term Memory (LSTM) recurrent neural networks significantly enhances its recognition accuracy and generalization capabilities across various fonts and linguistic variations. This advanced architecture allows Tesseract to learn from training data and adapt to different styles, making it more robust and suitable for real-world applications like ours.

We leverage Pytesseract, a Python wrapper for Tesseract, to perform text recognition on the preprocessed images, optimizing key configuration parameters to improve further accuracy for the specific characteristics of student ID cards.

In our investigation into optimal Optical Character Recognition (OCR) models for Vietnamese text recognition, two leading candidates, VietOCR and Tesseract OCR, underwent rigorous evaluation. VietOCR, while exhibiting commendable processing speed and accuracy, particularly for handwritten Vietnamese script, presented a critical constraint: its capacity for effective processing is limited to single lines of text. Applying VietOCR to student ID cards would necessitate manual region definition for information fields like name and student ID. While the consistent layout of FPT University student ID cards facilitates this process, as textual information is reliably located beneath the portrait, manual segmentation within the confined space of the card introduces a significant risk of errors. Specifically, separating closely positioned fields like name and student ID using fixed coordinates could lead to information loss, especially with variations in text length. While more sophisticated image processing algorithms could potentially mitigate this, such approaches would introduce undesirable performance overhead. This limitation of VietOCR significantly influenced our decision to adopt Tesseract OCR, which offers superior performance, notably its capacity for concurrently processing multiple lines of Vietnamese text, thereby eliminating the need for manual region definition and the associated risks.

Incorporating Long Short-Term Memory (LSTM) recurrent neural networks substantially augments Tesseract's recognition accuracy and generalization

capabilities across diverse fonts and linguistic variations. This sophisticated architecture empowers the model to learn from the training corpus and adapt to a wide array of textual formats, enhancing its robustness and suitability for practical applications, including the present study.

We leverage Pytesseract, a Python wrapper for Tesseract, to perform text recognition on the preprocessed images, optimizing key configuration parameters to further improve accuracy for the specific characteristics of student ID cards.

F. Data Extraction Using Regular Expressions

The extracted text from Fig. 5 often requires further refinement due to the presence of extraneous characters. To isolate the pertinent information, namely the student's name and ID number, we utilize Python's re library, which provides robust regular expression operations. Student ID numbers at FPT University adhere to a specific structure, consisting of a campus code (H, S, C, N, D), and a major code (S, E, etc.), followed by a six-digit numeric sequence. Exploiting this predictable format, we implement a targeted extraction strategy. Using a regular expression, we identify the line containing the two-character campus and major codes, along with the six-digit sequence. This matched string is then extracted as the student ID. Subsequently, we extract the student's name from the preceding line(s), accommodating potential intervening newline characters.

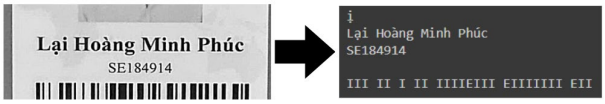

Fig. 5. Text extraction from student ID cards.

Extracting student information from Fig. 5 presents challenges due to the presence of extraneous characters in the OCR output. Accurate identification of student names and ID numbers requires a robust approach. We utilize Python's re-library and regular expressions, exploiting the structured format of FPT University student identification numbers. These IDs consist of a two-character campus code (H, S, C, N, D), and a two-character major code (e.g., SE), followed by a six-digit sequence. Our initial strategy involved matching this specific pattern to extract the ID. Subsequently, the student's name is extracted from the preceding lines, accommodating potential newline characters.

However, preliminary results revealed limitations in this approach. OCR errors frequently introduce extraneous whitespace within the ID string and often misinterpret the digit "1" as "I", "l", or "L" (Fig. 6). To address these issues, we refined our regular expression. The revised expression now disregards whitespace within the ID string, providing greater flexibility in matching (Fig. 7). Furthermore, it incorporates an alternative pattern that accepts "I", "l", or "L" in place of the first digit of the six-digit sequence, significantly improving the accuracy of ID extraction in the presence of this common OCR error. This enhanced method demonstrates increased robustness and accuracy in extracting student information from the noisy OCR output.

Fig. 6. Unwanted character removal.

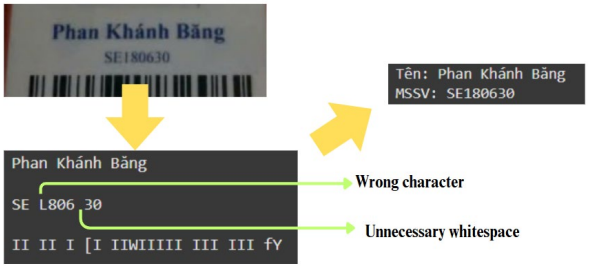

Fig. 7. Limitations in regular expressions.

We tested our extraction method on 150 images degraded by brightening, darkening, and blurring (the code snippet provided illustrates these transformations) (Fig. 8). We achieved an 86% success rate (129/150 images). For these successful extractions, the average name Character Error Rate (CER) and Word Error Rate (WER) were 0.0011 and 0.0039, respectively. The average ID CER and WER were even lower, at 0.0008 and 0.0067, respectively. ID extraction errors were minimal, typically involving only a single misidentified character. Name extraction errors frequently occurred in Vietnamese diacritical marks. The model performed well on blurred and darkened images but struggled with overly bright images, sometimes yielding spurious information. The robustness to blurring and darkening, combined with the remarkably low ID error rate, highlights the method's effectiveness under challenging conditions.

```
For all processed images:
  - Total images: 150
  - Images with extracted information: 129
  - Percentage of images with no information: 14.00%
  - Percentage of images with information: 86.00%

For images with extracted information only:
  - Average Name CER: 0.0011
  - Average Name WER: 0.0039
  - Average ID CER: 0.0008
  - Average ID WER: 0.0067
```

Fig. 8. Name and ID extraction accuracy on degraded images.

G. Evaluation Metrics

- *Mean Average Precision (mAP):* mAP is the primary metric for object detection tasks, calculating the average precision across various Intersections over Union (IoU) thresholds. It provides a comprehensive measure of the model's capability to detect objects accurately.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \quad (3)$$

- *Precision:* This metric measures the ratio of true positive detections to the total positive detections(true positives + false positives). A high precision indicates that most of the detected objects are correctly identified.

$$P = \frac{TP}{TP+FP} = \frac{TP}{ALL\ detection} \quad (4)$$

- *Recall:* Recall assesses the ability of the model to identify all relevant objects in an image. It is calculated as the ratio of true positive detections to the total actual positives (true positives + false negatives).

$$R = \frac{TP}{TP+FN} = \frac{TP}{ALL\ ground\ truths} \quad (5)$$

- *F1-Score:* The F1-Score combines precision and recall into a single metric through the harmonic mean, balancing the two for better performance understanding. This is especially useful when detecting objects in imbalanced datasets.
- *Intersection over Union (IoU):* IoU measures the overlap between the predicted bounding box and the ground truth bounding box, serving as a basis for determining true positive detections. A higher IoU indicates better localization performance, which means a better understanding of evaluation metrics in the object or student ID card detection task.
- *Character Error Rate (CER):* CER measures the rate of character-level errors in a candidate text, making it useful for identifying mispronunciations and phoneme errors [14]. It is calculated by dividing the total number of incorrect characters by the total number of characters in the reference text and is expressed as a percentage [15].
- *Word Error Rate (WER):* WER measures the accuracy of a candidate text by considering word-level errors, including substitutions, deletions, and insertions [14]. It is calculated by dividing the total number of word errors by the total number of words in the reference text [15].

H. Preparation

1) Dataset

The dataset consists of more than 1800 images, of which 900 are FPT University student ID cards, collected from various sources, including the school's official website and asking students from previous and future student ID cards to take a survey and share student ID cards, ensuring compliance with the school's image usage policy. The data includes ID cards of previous and current students, new and old ID cards, with various sizes and distances, to ensure the representativeness and generalizability of the model, and 900 other images of various types.

To collect the dataset, we designed a volunteer registration form for data contribution in Google Forms (Fig. 9) and shared it with the 9.5 AI community group. We commit that the whole dataset provided by contributors will be utilized just for study and model development, with no ulterior intents.

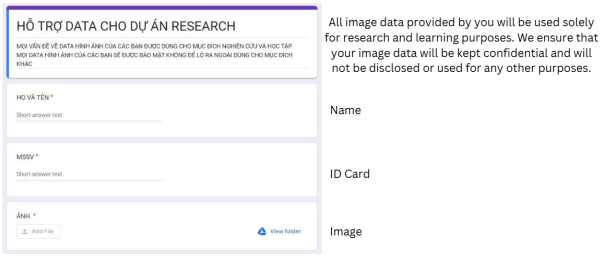

Fig. 9. Data support request form.

To enhance the diversity of the data and enhance model training, we apply some Data Augmentation techniques, including Image rotation: Which helps the model recognize student ID cards at different angles. Add blur noise: Helps the model process low-quality images and increases robustness.

Rotating the image helps the model recognize cards at different angles while adding blur noise (Fig. 10) helps the model handle low-quality images. We choose to improve the model's capacity to manage changes in lighting and color circumstances using brightness and color augmentation. Although the adjustments to these criteria on FPT student cards are currently minor, adding these improvements will help to future-proof the model. This phase ensures that the model stays successful even under poor lighting circumstances, broadening its applicability outside classrooms where lighting is normally good.

After collecting the student card image data of FPT University, we used Roboflow to label the card images. We used bounding boxes to label the student cards. After collecting and processing the images, we had to label a dataset of more than 1800 images within 1 day. We divided the dataset into 80% train, 10% valid, and 10% test.

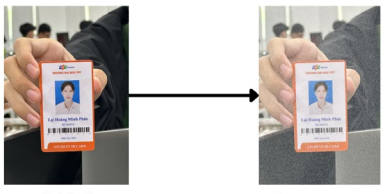

Fig. 10. Blur noise student ID card.

2) Training

Using the pre-trained weights of YOLOv10n trained on the MS COCO dataset [16].

To effectively detect elements within student ID cards, we fine-tuned a YOLOv10n object detection model pre-trained on the MS COCO dataset [16]. This approach leveraged the model's existing knowledge while adapting it to our specific task. The fine-tuning process involved training the model for 30 epochs with a batch size of 16 and input images resized to 640×640 pixels. Crucially, we

carefully selected hyperparameters to optimize performance and prevent overfitting. An initial learning rate of 0.00002 allowed for fine-grained adjustments, while a weight decay of 0.0005 mitigated overfitting by regularizing the model's weights. Early stopping with a patience of 5 prevented prolonged training on a plateauing validation performance. Furthermore, we strategically disabled mosaic data augmentation during the final 5 epochs, recognizing the significant difference between our ID card images and the MS COCO dataset. This tailored approach allowed the model to focus on the unique visual features present in ID cards. Through manual hyperparameter tuning and validation set evaluation, this specific configuration proved optimal, yielding the best mAP-50 score and demonstrating the positive impact of these carefully chosen parameters on the model's ability to accurately and efficiently detect elements within student ID cards.

IV. Result and Discussion

A. Evaluate the Model on the Validation Set

The current results section provides detailed evaluation metrics, but to highlight the system's performance, adding charts or images will help visualize key outcomes. Below is a proposed addition to this section:

As shown in Table II, the model demonstrates impressive performance across both the "Card" and "Other" classes. Specifically, the "Card" class achieves Recall (0.978) and very high precision (0.908), indicating that the system can detect most student ID cards in the dataset accurately and comprehensively.

TABLE II. Evaluation Results on the Validation Set

Class	Instances	P	R	mAP50	mAP50–95
all	181	0.889	0.89	0.946	0.805
Card	90	0.908	0.978	0.99	0.934
Other	96	0.989	0.869	0.803	0.675

B. Evaluate the Metrics

In Fig. 11, the precision-confidence curve evaluates model performance by illustrating the relationship between predicted bounding boxes' precision (Y-axis) and confidence (X-axis). Confidence (0–1) represents the model's certainty that a bounding box contains an object. At the same time, precision is the ratio of correctly predicted bounding boxes to the total number of predicted bounding boxes exceeding a given confidence threshold. The plot includes separate curves for the "Card" class, the "Other" class (encompassing all other classes), and an average curve for all classes ("all classes"). For instance, "all classes 1.00 at 0.893" signifies that at a confidence threshold of 0.893, the average precision reaches nearly 1.00, meaning almost all bounding boxes with confidence 0.893 or higher are correct. Observing the plot reveals that the model performs better at detecting "Card" than "Other" because the "Card" curve exhibits higher precision across most confidence levels. This information allows for adjusting the confidence threshold to balance precision and recall (the ability to detect all objects) for individual classes and overall.

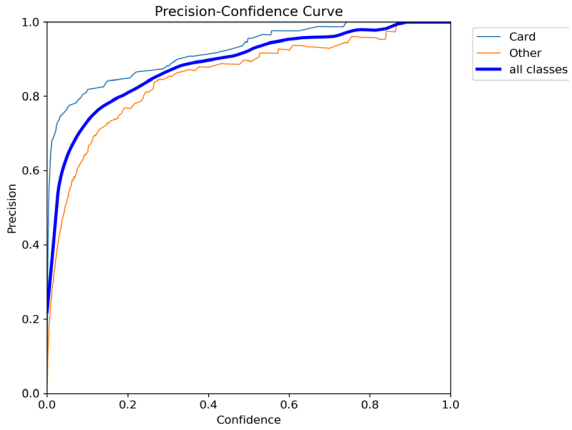

Fig. 11. Precision confidence curve.

In Fig. 12, The Recall-Confidence Curve helps assess a YOLOv10 model's performance by showing the relationship between confidence (X-axis) and recall (Y-axis). Confidence from 0 to 1 represents the model's certainty that a bounding box contains an object. Recall, calculated as the ratio of true positives to the total number of actual objects (true positives + false negatives), reflects the model's ability to find all true positives at a given confidence threshold or higher. The curve includes lines for the "Card" class, "Other" classes (all others combined), and "all classes". "All classes 1.00 at 0.000" signifies perfect recall (1.00) at a 0 confidence threshold, meaning all bounding boxes are considered, including all true positives, though this would likely yield low precision. Analyzing this curve allows for selecting an appropriate confidence threshold to balance recall and precision based on the application's needs.

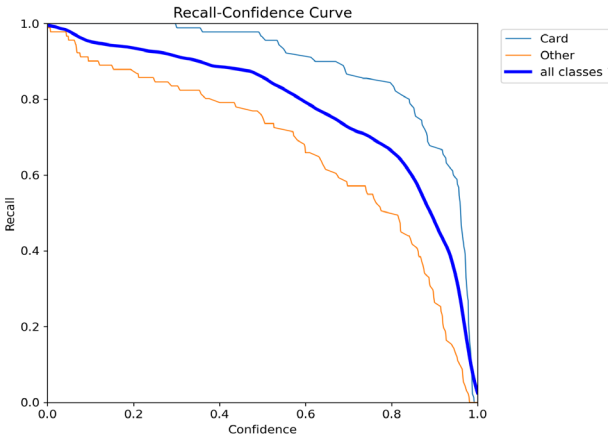

Fig. 12. Recall-confidence curve.

In Fig. 13, this F1-Confidence Curve visualizes the relationship between the F1-Score (Y-axis) and the confidence threshold (X-axis) for an object detection model, likely YOLOv10. The F1-Score, a balanced measure of precision and recall, provides a single metric to assess performance. Confidence, ranging from 0 to 1, indicates the model's certainty in its bounding box predictions. Separate lines depict the F1-Score's variation with confidence for the "Card" class, the "Other" classes

(all other classes combined), and the overall performance ("all classes"). The annotation "all classes 0.89 at 0.338" signifies that the highest average F1-Score (0.89) across all classes is achieved at a confidence threshold of 0.338. This curve helps determine the optimal confidence threshold to maximize the F1-Score, balancing precision and recall. As seen in the graph, the "Card" class generally achieves a higher F1-Score than the "Other" classes across different confidence thresholds. This visualization assists in understanding the model's performance and selecting the best confidence threshold for specific application requirements.

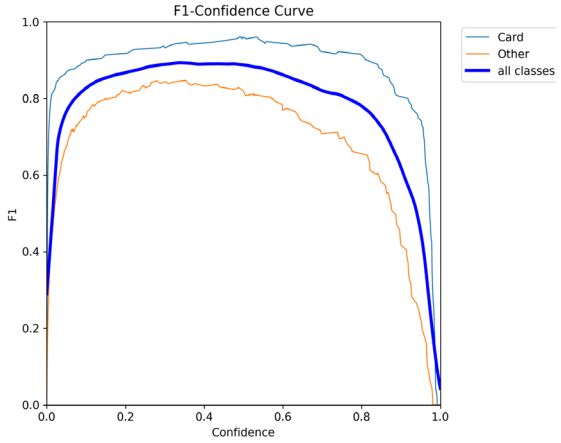

Fig. 13. F1-confidence curve.

C. Discussion

Despite possessing numerous notable advantages, this model still has some shortcomings that have not been fully addressed. One of the major limitations is the impact of uneven lighting conditions. When the light is too dim, the model struggles to identify critical details, while overly bright light causes reflections that blur the information on the card. This directly affects the model's performance, resulting in outcomes that fall short of the desired accuracy. Additionally, the physical condition of the student ID card is another crucial factor that cannot be overlooked. Cards that are scratched, torn, or missing essential information such as names or ID numbers pose significant challenges to the model. In such cases, the recognition capability may be severely compromised, potentially leading to complete inaccuracies. To improve, it is necessary to optimize the algorithms, enhance image processing capabilities under adverse conditions, and train the model with a more diverse dataset, including damaged cards. These limitations highlight that the model still requires further refinement to better meet real-world demands. We will continue to enhance this model in the future to address these drawbacks effectively.

This paper proposes a robust solution for automated information extraction from student ID cards, which uses the YOLOv10 model for detection and Tesseract OCR for text recognition. The system obtained an amazing 94.6% detection accuracy and 95% text extraction accuracy, demonstrating its efficacy in educational contexts where consistent and efficient attendance monitoring is crucial.

Fig. 14 shows that the model operates best under stable lighting circumstances often observed in schools, which is consistent with its intended purpose. Notably, the model performed well under low-light conditions, demonstrating its robustness in suboptimal scenarios. However, it was discovered that the system's accuracy decreases in areas with excessively bright lighting, such as direct sunshine or uneven illumination in the late afternoon. These issues are most likely caused by overexposure, which reduces the clarity of card elements and text.

Fig. 14. Bright and Dark testing.

To address these constraints, future enhancements will focus on increasing the model's robustness under different illumination circumstances. This includes the use of advanced data augmentation techniques and adaptive preprocessing approaches to properly manage both bright and dim situations. To avoid errors and ensure constant performance, we propose deploying the system in regulated lighting environments.

When applying YOLOv10 for the recognition and classification of student ID cards and other similar card types, several distinct challenges emerge that exacerbate the model's inherent limitations [17]. The detection and classification of card-based documents involve recognizing not only the card as a whole but also extracting fine details such as text fields, logos, and security features which are often small and intricately designed. Due to its grid-based detection approach, YOLOv10 may struggle to provide the necessary spatial resolution to accurately capture small text elements or nuanced design features that are critical for identity verification [18]. Furthermore, card images usually contain a variety of background patterns, reflections, and lighting variations which can impede the model's performance by introducing noise and distractors in the detection process [19]. In cases where the student IDs or other cards possess intricate designs or fine details, the trade-offs made in YOLOv10 to ensure real-time inference can result in missed detections or misclassifications. Additionally, the challenge is compounded by the fact that many ID cards have non-standardized layouts across different institutions or countries, placing further strain on the model's generalization capabilities.

Our regex-based text extraction method prioritizes efficiency and accuracy. Unsuccessful information retrieval is primarily associated with overexposure, a condition less common in our target environment. The method's key strength is its robustness against darkening, blurring, and noise, which often challenge traditional OCR

systems. This resilience makes our approach well-suited for real-world applications.

V. Conclusion

This paper presents an automated model for detecting and extracting information from student ID cards by combining YOLOv10 and Optical Character Recognition (OCR). The proposed method utilizes YOLOv10 for student ID card detection and employs Tesseract OCR for rotation handling without introducing black borders on the card. Subsequently, the region containing the information on the card is cropped, a technique that has proven effective in experimental trials. The model achieves an impressive accuracy of 98%, indicating its reliability in practical applications. This system holds potential for future expansion and implementation in real-world environments, enhancing automation and accuracy in processing information from student ID cards.

In future development, this system can be extended to handle a wider range of identification documents, such as passports and driver's licenses, and can be integrated into broader administrative systems across educational institutions and other industries. Additionally, further optimizations could focus on improving robustness under challenging conditions such as varying lighting, low image resolution, and more complex backgrounds. These improvements would enhance the system's adaptability and reliability in real-world applications, especially in high-security or large-scale administrative environments.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Toan Quy Nguyen, Tuan Minh Le, and Linh Van Nguyen, along with Phuc Hoang Minh Lai, conceptualized the project and defined its research direction. Toan Quy Nguyen, Tuan Minh Le, and Linh Van Nguyen conducted literature reviews and explored various models, including VietOCR, PaddleOCR, and YOLO versions 8, 9, and 10. The team divided the experimentation tasks: Toan Quy Nguyen and Tuan Minh Le focused on YOLOv8 and YOLOv10, Khang Vo Hoang Nguyen and Thuan Van Tran worked on YOLOv9, while Linh Van Nguyen and Phuc Hoang Minh Lai evaluated OCR models. Phuc Hoang Minh Lai, Khang Vo Hoang Nguyen, and Thuan Van Tran were responsible for data collection, preprocessing, and augmenting the dataset to enhance its diversity and robustness. The entire team collaborated in labeling and cross-checking the dataset to ensure accuracy and consistency. Thu Vo Minh Le provided supervision, offered methodological insights, and guided the research direction. All authors contributed to writing the manuscript, reviewed the final draft, and approved the submitted version.

References

[1] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, and J. Han, "YOLOv10: Real-time end-to-end object detection," arXiv preprint, arXiv:2405.14458, 2024.

[2] Faceonlive. (May, 2024). Understanding YOLOv8 for ID card detection and text extraction. [Online]. Available: https://faceonlive.com/understanding-yolov8-for-id-card-detection-and-text-extraction

[3] Ultralytics. (September, 2024). YOLOv10. In Ultralytics Documentation. [Online]. Available: https://docs.ultralytics.com/models/yolov10

[4] C. Y. Lin, M. Wu, J. A. Bloom *et al*., "Rotation, scale, and translation resilient public watermarking for images," *IEEE Trans. Image Process*., vol. 10, no. 5, pp. 767–782, May 2001.

[5] J. Singh and B. Bhushan, "Real-time Indian license plate detection using deep neural networks and optical character recognition using LSTM tesseract," in *Proc. 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, 2019.

[6] R. Smith, D. Antonova, and D. S. Lee, "Adapting the Tesseract open-source OCR engine for multilingual OCR," in *Proc. the International Workshop on Multilingual OCR*, 2009. doi: 10.1145/1577802.1577804

[7] A. Cichocki and R. Unbehaved, *Neural Networks for Optimization and Signal Processing*, 1st ed. Chichester, U.K.: Wiley, 1993, ch. 2, pp. 45–47.

[8] Microsoft Learn. (April, 2024). Computer vision OCR size matters? Microsoft Q&A. [Online]. Available: https://learn.microsoft.com/en-us/answers/questions/1656752/computer-vision-ocr-size-matters

[9] Formula to determine perceived brightness of RGB color. [Online]. Available: https://stackoverflow.com/questions/596216/formula-to-determine-perceived-brightness-of-rgb-color

[10] Synopsys. (2020). Color space conversions for display. [Online]. Available: https://www.synopsys.com/blogs/chip-design/color-space-conversions-display.html

[11] Gaussian Blur. Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Gaussian_blur

[12] Gaussian Blur. (2012). W3C HTML5 Training, Tehran. [Online]. Available: https://www.w3.org/Talks/2012/0125-HTML-Tehran/Gaussian.xhtml

[13] Y. Sharma. (2021). Python OpenCV filter2D() function—A Complete Guide. [Online]. Available: https://www.askpython.com/python-modules/opencv-filter2d

[14] WER, CER, and MER. [Online]. Available: https://docs.kolena.com/metrics/wer-cer-mer/

[15] Tamanna. Deciphering Accuracy: Evaluation Metrics in NLP and OCR—A Comparison of Character Error Rate (CER) and Word Error Rate (WER). [Online]. Available: https://medium.com/@tam.tamanna18/deciphering-accuracy-evaluation-metrics-in-nlp-and-ocr-a-comparison-of-character-error-rate-cer-e97e809be0c8

[16] T. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common objects in context," arXiv preprint, arXiv:1405.0312, 2014.

[17] THU-MIG. (2024). Problems with detecting smaller objects or objects in the distance. [Online]. Available: https://github.com/THU-MIG/yolov10/issues/100

[18] S. Gharnit. (2025). YOLO model for real-time object detection: A full guide. [Online]. Available: https://www.viam.com/post/guide-yolo-model-real-time-object-detection-with-examples

[19] YOLOv10 vs RTDETRv2: A technical comparison for object detection. (2024). [Online]. Available: https://docs.ultralytics.com/compare/yolov10-vs-rtdetr/