Journal of Innovative Image Processing (ISSN: 2582-4252)
www.irojournals.com/iroiip/

A Deep Learning–based Framework for Certificate Information Extraction and Authentication

Doan Van Thang[1*], Nguyen Ngoc Dung[2]

Faculty of Information Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam.

E-mail: [*1]vanthangdn@gmail.com, [2]nguyenngocdung.fit@gmail.com

Abstract

This paper presents a deep learning-based end-to-end system for the automatic extraction of key information from structured certificate documents. The model was trained with 1,784 manually labeled certified corpus images. The research finds locations in the dataset utilizing the YOLO models v11 and v12. While YOLOv11 has precision = 0.987, recall = 0.996, mAP@50 = 0.981, and mAP@50–95 = 0.678, YOLOv12 has precision = 0.992, recall = 0.998, mAP@50 = 0.986, and mAP@50–95 = 0.690. It can be seen from the experimental results that YOLO v12 excels in detecting objects (19.1 ms vs. 224.4 ms per image). In order to realize the capability to extract and verify good certificate information, the research then proposes an integrated object detection, optical character recognition (OCR), and database comparison process. In the future, instance segmentation, multimodal learning, and personalized OCR enhancement can be employed to further improve the system's performance on different categories of documents.

Keywords: Information Extraction, Object Detection, Deep Learning, OCR.

1. Introduction

Certifications are a classic example of the solid document automation that has become unavoidable in light of the growing dependency on computer systems. Certifications are seldom received in pristine digital form in real-world scenarios; rather, they are captured in uncontrolled environments, which leads to distorted layouts, low lighting, soiled backgrounds, and degraded text quality. The deskewing, gamma correction, Gaussian smoothing, and denoising preprocessing pipeline is employed prior to optical character recognition (OCR) to combat these degradations. Preprocessing is necessary here since traditional rule-based OCR systems require pre-specified templates and cannot generalize to documents containing random print quality, multilingual text, or heterogeneous layouts.

The present study explores deep learning-based object detection for certificate field localization as a remedy to such problems. To enhance the detection of near-perceptually low-contrast or tightly spaced textual areas, subsequent YOLO versions (YOLOv11 and YOLOv12) include architectural innovations such as attention, cross-scale feature fusion, and sophisticated loss functions. In contrast to the earlier rule-based approach, detectors based on these models are well-suited for certificate examination in real-world settings and accommodate bilingual

DOI: https://doi.org/10.36548/jiip.2025.3.024
Received: 28.07.2025, received in revised form: 03.09.2025, accepted: 20.09.2025, published: 30.09.2025

annotation (e.g., Vietnamese–English) as well as layout variation. In structured document datasets, where the problems are overlapping areas, visually similar fields, and multilingual text, relatively few published analyses of YOLO's more recent versions have been carried out, although it has achieved high performance on overall detection issues.

We collected 1,784 certificates to fill this gap and tagged them with eight semantic fields in both Vietnamese and English. YOLOv11 and YOLOv12 were utilized to test the dataset with respect to inference latency, detection density, and mean average precision (mAP). A transformer-based model for document understanding named LayoutLMv3 was also tested to compare it in order to check trade-offs between detection accuracy and computational cost.

A complete information extraction pipeline comprises three integrated processes: (i) Tesseract OCR text recognition, (ii) region detection with YOLO, and (iii) post-processing normalization and cross-checking of database records. The architecture provides semantic validation and precise detection of identified fields.

The rest of the paper is organized as follows: Section II presents related work; Section III explains dataset collection and annotation; Section IV provides experimental setup and results; Section V explains system architecture and pipeline integration; Section VI explores limitations and deployment considerations; and Section VII concludes with guidance on future research.

2. Literature Survey

The YOLO (You Only Look Once) family of models has undergone rapid evolution over the past decade, with each version pushing the boundaries of real-time object detection. Early versions, such as YOLOv2 and YOLOv3, introduced key innovations including anchor boxes, batch normalization, and multi-scale detection, which significantly improved accuracy while enabling real-time inference. These contributions reframed object detection as a regression task and quickly attracted adoption across application domains. For example, AlQudah et al. [1] adapted YOLO for the detection of license plates in crowded traffic conditions, while Zeng et al. [2] used luminance attention to address low-light conditions. This kind of work reflects challenges that are analogous to those that exist with certificate images, including small font, layout irregularities, and scan quality degradation.

Subsequent work has focused on architectural optimizations. Chen [3] presented a well-organized summary of backbone networks, feature fusion approaches, and loss functions, whereas He et al. [4] contrasted YOLOv5 with YOLOv11 for industrial inspection and concluded that YOLOv11 performed best. Dwivedi et al. [5], Cong et al. [6], and Safaldin et al. [7] demonstrated the ability of YOLO to detect moving objects, and Ananth et al. [8], Katyayani et al. [9], Boussaad and Boucetta [10], and Atik et al. [11] implemented YOLO in challenging regions such as dense multiple-object scenes and aerial imaging. These collectively indicate that YOLO's advantage is not so much in achieving a one-off high mAP score but in its flexibility across tasks. At the same time as these advances, there has been a presence of literature with an applied bent.

Padmane [12] provided an overview of real-time detection systems generally, and Patil et al. [13] wrote a practitioner's guide to model selection and optimization. Ponika et al. [14] presented OpenCV-based implementations, and Sirisha et al. [15] employed statistical analysis for architectural design trade-off analysis. Models based on transformation have instead been

dominant in Document AI. LayoutLMv3 [16] and DocParser [17] are examples that leverage textual, visual, and even layout knowledge to enhance document comprehension. Even though powerful, these models are accompanied by enormous computation costs and thus are not ideal for fast certificate processing where speed is of the essence. Most recently, YOLOv12 (2025) incorporated Residual Efficient Layer Aggregation Networks (R-ELAN) [18] that improve the detection of small and densely packed objects.

Early accounts show that YOLOv12-N achieves improvements in mAP over YOLOv11-N with small increases in latency, for example, in application areas like fruit detection where both recall and precision were increased. Little has been shown, though, from our literature search in academic repositories regarding the use of YOLOv12 on document-centric use cases. Certificates, above other documents, are especially difficult due to their constrained layout, bilingual text, and institutionally internal structure. This scarcity spurs this work.

We comparatively analyze YOLOv11 and YOLOv12 end-to-end directly for ordered certificate analysis, constructing a 1,784 annotated certificate dataset with eight fields labeled in both Vietnamese and English. Besides field detection, we construct an end-to-end pipeline that fuses YOLO-based localization with OCR text reading and database verification. At the same time, we compare YOLO's lean performance against the contextual intelligence of two strong baselines, LayoutLMv3 [16] and DocParser [17], thereby offering speed, accuracy, and context-awareness trade-offs in certificate digitization in the real world.

3. Methodology

We found that our problem space had no suitable public datasets while doing this project. Though they were clean and nicely formatted, they didn't have the randomness and minor imperfections of real certificate photos. So, we created a customized dataset that is realistic and helpful. We constructed the dataset with four typical certificate templates to generate layout variations. Personal sensitive information was anonymized by manually blurring or masking private text, and original certificates were procured with the necessary authorization. The areas of interest that mapped to the eight semantic classes bilingual names, birthdates, registration numbers, certificate numbers, and reference IDs were labeled manually. Though it took a tremendous amount of resources, the manual and time-consuming annotation process yielded accurate and consistent labels. All 1,784 images were resized to 640 x 480 pixels to normalize the inputs and preprocess the training model.

Using the stratified sampling method, the 1,784 images were divided into three subsets: 70% for training, 15% for validation, and 15% for testing. Certificate layouts across subsets were evenly distributed, and the split was balanced in terms of classes. The following field detection and evaluation experiments are performed against Table 1. which provides annotated object classes and their descriptions.

Table 1. Object Classes and Annotation Descriptions

Class ID	Description
0	Recipient's full name in Vietnamese
1	Date of birth (Vietnamese version)

2	Certificate serial number
3	Registration number of issuance
4	Recipient's full name in English
5	Date of birth (English version)
6	Registration number (English)
7	Reference number (English)

Figure 1. Example of Annotated Bounding Boxes on a Certificate Image

As shown in Figure 2, the dataset was divided into three subsets:

- Training set: 1,257 images (70%)
- Validation set: 264 images (15%)
- Test set: 263 images (15%)

This balanced split ensures both generalization and fair evaluation of the models during the training and testing phases.

Dataset Split

Figure 2. The Dataset is Divided into Three Parts: Train, Test, and Validation

The class distribution was analyzed to examine balance across fields. The annotated dataset includes Name_vi (2,500 instances), DOB_vi (1,784), Serial (1,784), RegNo (1,600), Name_en (2,450), DOB_en (1,780), RegNo_en (1,600), and RefNo (1,550). Annotated batches were also visualized to identify and correct label noise, thereby ensuring annotation quality and consistency.

3.1 Annotation Visualization

During the process of building the dataset for the research, in the data pre-processing and model training stages, we performed labeling on both v11 and v12 versions of the YOLO model to check whether the data is accurate and diverse. Each batch is displayed as a 4×4 grid of random images, in which the information fields are marked with bounding boxes of their own colors and class IDs. We can easily observe whether the labels are missing, duplicated or incorrect. In addition, this visualization also helps to clearly see the diversity of the data: text can rotate in many directions, appear in both Vietnamese and English, and lie on different complex backgrounds.

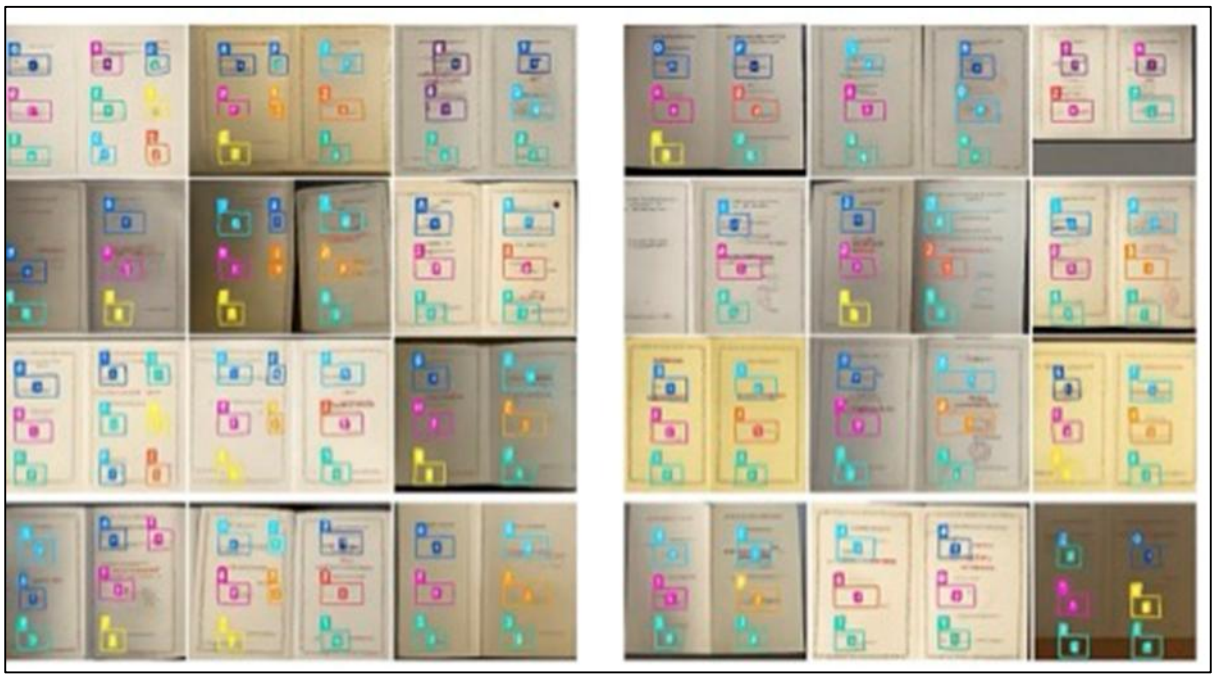

Figure 3. Sample Training Batches from YOLOv11 (Left) and YOLOv12 (Right)

3.2 Model Configuration and Training Details

YOLOv11 and YOLOv12 are newer YOLO versions that try to balance speed with accuracy, especially when objects are close together. Both follow the anchor-free, one-stage idea, using distribution focal loss (DFL) and objectness scores.

The latest versions of the YOLO family, YOLOv11 and YOLOv12, are designed to strike a balance between efficiency, speed, and accuracy for tasks involving dense object detection. Both use a one-stage, anchor-free detection framework and refine bounding box

localization and classification using Distribution Focal Loss (DFL) in combination with objectness scoring.

YOLOv11 maintains the RepNCSPELAN backbone introduced in YOLOv8, which is computationally optimized yet robust in representational capability. YOLOv12, on the other hand, presents architectural enhancements within the decoder phase, including attention-based feature aggregation for enhancing cross-scale information fusion. These changes allow for better detection of small and nearby objects, a feature especially useful for certificate images where bilingual text fields often appear close to each other and have high visual similarity.

Both models were trained on the Ultralytics pipeline with pre-trained weights (yolov11n.pt and yolov12n.pt). Training was conducted for 100 epochs with automatic mixed precision (AMP) turned on to speed up computation and minimize memory usage. Experiments were run on one GPU with eight dataloader workers to maximize throughput. Random seeds were set to maintain reproducibility across runs.

The number of epochs chosen, 100, was determined through empirical observation. Initial experiments revealed that learning curves started plateauing around epochs 80 and 90 with minimal improvements beyond these. As such, 100 epochs were used as a conservative stopping point to verify model convergence without excess computation. Hyperparameter details are presented in Table 2.

Table 2. Training Hyperparameters used for Model Training

Parameter	Values
Image size	640 × 480
Batch size	16
Learning rate	0.01
Optimizer	Adam / SGD
Scheduler	Auto
Warmup epochs	None
Augmentation	3
Mosaic	RandAugment
Erasing	1.0
Auto augment	0.4

3.3 Data Augmentation Strategy

During training, to improve stability, we applied several techniques as follows:

- Color jittering with HSV shift (h=0.015, s=0.7, v=0.4).
- Geometric transforms: random scaling (0.5), translation (0.1), horizontal flip (50%)
- Erasing and mosaic augmentations.

- RandAugment policy to improve performance under variable lighting and background noise.

These augmentations were carefully selected to simulate common distortions observed in real-world certificate scans, such as varying lighting conditions, ink differences, layout changes, and image noise. Mosaic and RandAugment, in particular, were effective in improving generalization to diverse layouts and languages. HSV jittering enabled the model to become robust to certificates printed with different color schemes and lighting artifacts.

3.4 Evaluation Metrics

In the object detection task, the model's performance was evaluated using the following metrics:

- Precision and recall across all classes.
- mAP@50 and mAP@50:95.
- Inference time (ms/image).
- Average number of boxes per image.

These metrics were chosen to evaluate both the accuracy and efficiency of the models. Precision and recall provide insight into detection reliability, while mAP reflects the quality of bounding box localization at different IoU thresholds. mAP@50 serves as a general benchmark, and mAP@50--95 highlights fine-grained detection performance. Inference time and box density were included to assess suitability for real-time and high-volume document processing scenarios. Metrics were computed using the Ultralytics built-in evaluation pipeline and confirmed via logs and plotted training curves.

4. Experimental Results

In this section, we present both quantitative and qualitative evaluation results comparing YOLOv11 and YOLOv12 on the custom certificate dataset.

4.1 Quantitative Results

We evaluate both YOLOv11 (shown in Figure 4) and YOLOv12 (shown in Figure 5) using standard object detection metrics including Precision, Recall, mAP@50, and mAP@50-95. The results are summarized in Table 3.

Table 3. Performance Comparison of YOLOv11 and YOLOv12 on Certificate Detection

Model	Precision	Recall	mAP@50	mAP@50:95
YOLOv11	0.987	0.996	0.981	0.678
YOLOv12	0.992	0.998	0.986	0.690

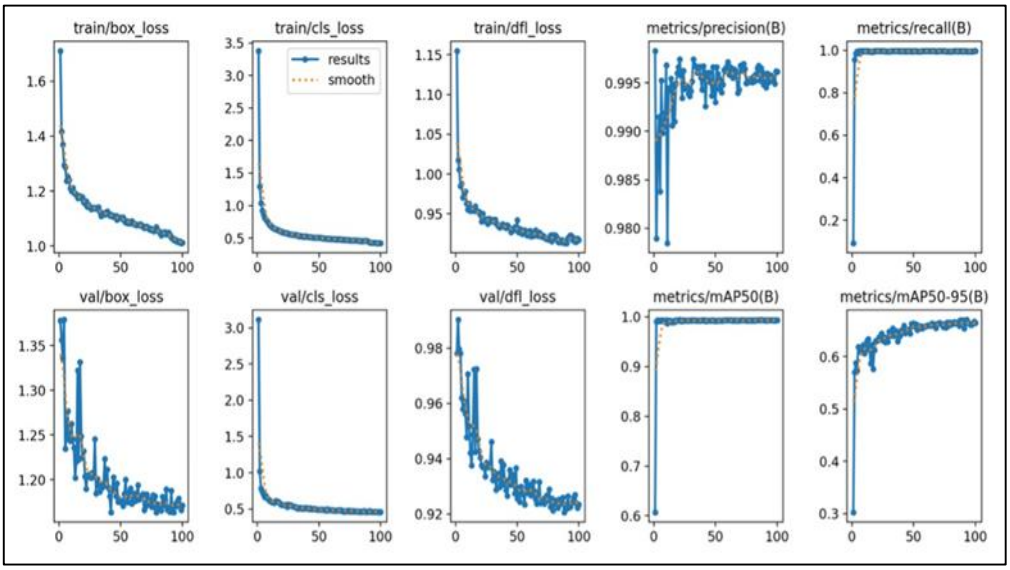

Figure 4. Training Result of YOLOv11

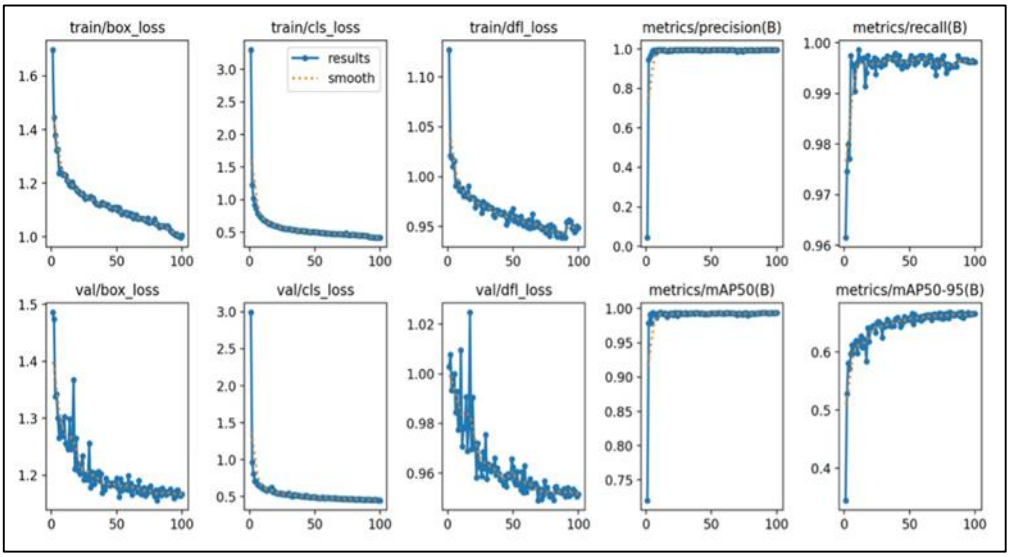

Figure 5. Training Result of YOLOv12

As seen from Table 3:

- YOLOv12 outperforms YOLOv11 across all metrics, especially in terms of mAP@50:95, indicating its better localization performance across a range of IoU thresholds.
- Both models achieve high recall and precision, reflecting accurate detection and minimal false positives or negatives.

We observed that the YOLOv12 model demonstrated superiority through its faster convergence, which was evident from the loss curves during training. In addition, throughout the training epochs, its performance on the validation set remained more stable.

4.2 Qualitative Results

To further compare the detection performance, we visualize bounding box predictions of both YOLOv11 and YOLOv12 on the same certificate image.

As illustrated in Figures 6(a) and 6(b), both models successfully detect most key information fields. In our tests, YOLOv12 provided tighter boxes and higher confidence scores, especially in the smaller text fields such as Ref No and Reg No.

- In the YOLOv11 result (Figure 6(a)), boxes 6 and 7 corresponding to Reg No and Ref No are detected with confidence scores of 0.82 and 0.69 respectively. The bounding boxes are present but slightly loose.
- In contrast, YOLOv12 (Figure 6(b)) produces higher confidence scores of 0.83 and 0.71 and tighter box boundaries in the same regions, suggesting better object localization.
- Moreover, YOLOv12 produces more consistent detections, regardless of whether the fields appear on the left or right page of the certificate.

These qualitative findings match what we observed in Section 4.1, and they highlight again how YOLOv12 manages to handle complex, tightly structured document layouts.

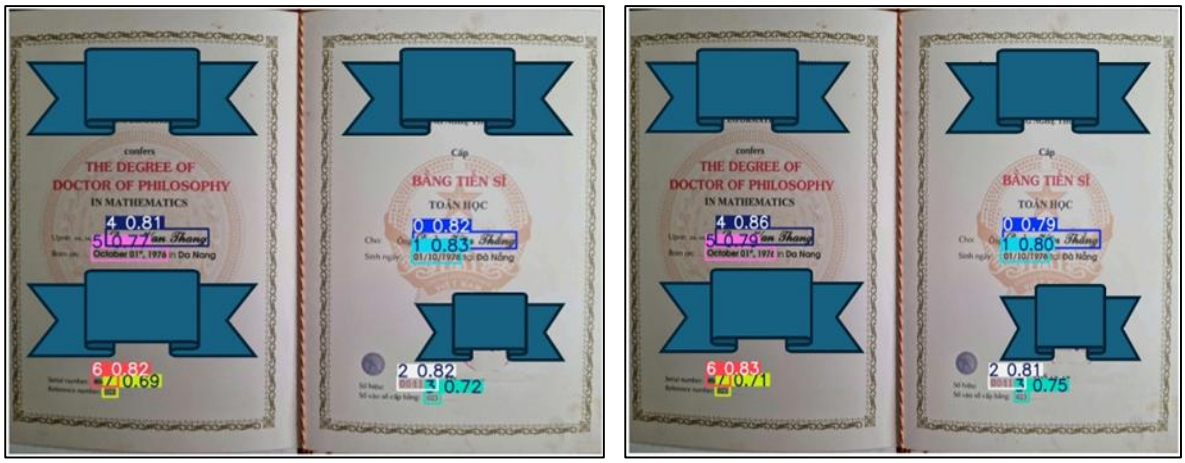

Figure 6. Results When Detecting Information: (a) YOLOv11, (b) YOLOv12

4.3 Inference Time and Detection Density

In addition to detection accuracy, we assess both models based on inference time and the average number of predicted boxes per image. Based on experimental results, it is shown that YOLOv12 outperforms YOLOv11 with higher accuracy, more stable behavior and faster computation speed, making it more suitable for real-time certificate processing systems.

Table 4. Inference Performance for Certificate Detection

Model	Inference Time (ms/image)	Avg. Boxes/Image
YOLOv11	22.4	8.2
YOLOv12	19.1	8.4

4.4 Training Curves

We examined the learning curves of both models over the 100 training epochs, as this gave us a clearer view of their training behavior. Figures 6(a) and 6(b) present the evolution of key metrics including box loss, classification loss, distribution focal loss (DFL), precision, recall, and mean average precision (mAP).

- YOLOv11 Figure 6(a) shows a steady decrease in all three loss components during training and validation, although its val/box_loss and val/cls_loss fluctuate slightly in the earlier epochs.
- YOLOv12 Figure 6(b) demonstrates a more stable and faster convergence. Loss values across training and validation reduce more smoothly, suggesting better generalization.
- In terms of precision, recall, and mAP metrics, YOLOv12 achieves quicker stabilization and higher overall values by epoch 30, while YOLOv11 reaches similar values but with more variation.

In line with the performance patterns discussed in previous sections, the training curves showed that YOLOv12 converged faster and attained greater stability than YOLOv11. We ran ablation experiments by altering the number of epochs (50 vs. 100), learning rate (0.001 vs. 0.01), and the addition of data augmentation in order to verify the selection of hyperparameters. YOLOv12 achieved mAP@50 = 0.986 and mAP@50–95 = 0.690, confirming that training for 100 epochs with a learning rate of 0.01 and augmentation yielded the best results.

Table 5. Ablation Results Showing the Effect of Epochs, Learning Rate, and Augmentation on the Performance of YOLOv12

Setting	Precision	Recall	mAP@50	mAP@50–95
Epochs = 50	0.987	0.994	0.972	0.642
Epochs = 100	0.992	0.998	0.986	0.690
LR = 0.001	0.981	0.990	0.968	0.635
LR = 0.01	0.992	0.998	0.986	0.690
Augmentation OFF	0.985	0.993	0.975	0.648
Augmentation ON	0.992	0.998	0.986	0.690

4.5 Confusion Matrix

To further evaluate model performance across all object classes, we analyze the confusion matrices for YOLOv11 and YOLOv12 on the test set. Figures 7 and 8 display the normalized confusion matrices.

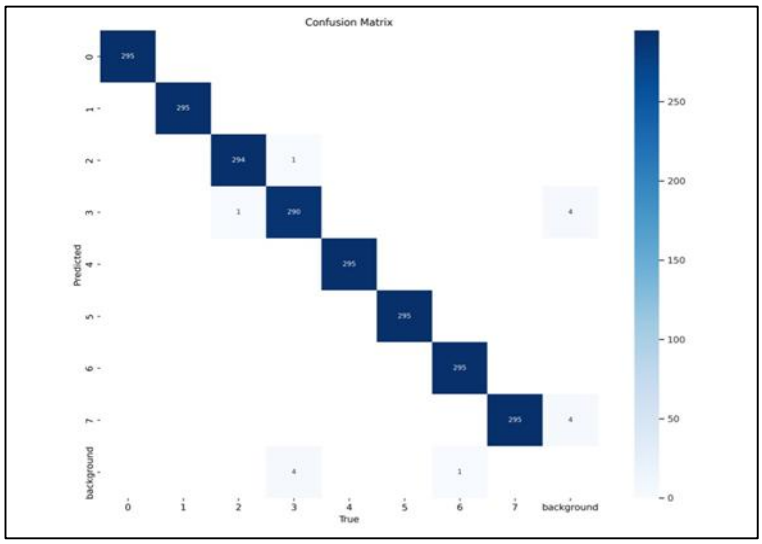

Figure 7. Confusion Matrices YOLOv11

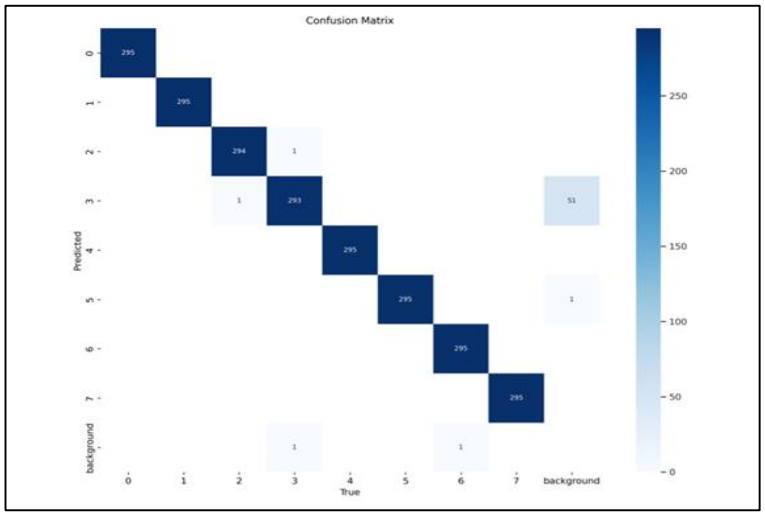

Figure 8. Confusion Matrices YOLOv12

- YOLOv11 (Figure 7) achieves nearly perfect classification across all classes, with minimal misclassifications. However, there is a slight confusion between class 2 Reg No and class 3 Ref No, and a notable number of false positives detected as background for class 6 and class 7 Reg No and Ref No.

- YOLOv12 (Figure 8) shows even better performance. It eliminates most background confusion seen in YOLOv11 and significantly reduces off-diagonal errors. Only minor confusion remains in class 3 and class 7, but overall accuracy per class is very high.

4.6 Baseline Comparison with Transformer-based Models

We also compared YOLOv12 with LayoutLMv3, a multimodal document understanding transformer-based architecture, to evaluate it on document-centric tasks more comprehensively. From the computational side, LayoutLMv3 was only fine-tuned on 300

certificate images, but the models were all compared under the same settings, including precision, recall, mAP@50, and mAP@50–95. LayoutLMv3's accuracy was similar to YOLOv12's, but its inference speed was almost an order of magnitude slower. This indicates an unequivocal trade-off: transformer models are very strong at semantic reasoning on more complex layouts but at a much higher computational cost. Alternatively, YOLOv12 appropriately identified well-structured certificate fields, such as registration and reference numbers, irrespective of layout design and scan quality, by achieving a decent balance between detection accuracy and run-time speed. These results show that even though transformer models like LayoutLMv3 offer more context modeling, their requirements for large annotated datasets, longer training time, and hardware-specific resources (i.e., GPUs/TPUs) make them difficult to implement in real-time or for large-scale certificate processing. However, YOLOv12 is more appropriate for certificate digitization pipelines where efficiency is essential due to its lightweight design and powerful detection capabilities.

Table 6. Performance Metrics Comparison

Model	Precision	mAP@50	mAP@50:95
YOLOv12	0.992	0.986	0.690
LayoutLMv3	0.985	0.982	0.682

Table 7. Qualitative Comparison of Detection Models

Model	Strengths	Limitations
YOLOv12	Fast, efficient, accurate for structured documents	Limited semantic context, purely visual
LayoutLMv3	Rich semantic modeling, strong contextual understanding	High latency, requires large labeled datasets

The experiment's findings demonstrated that LayoutLMv3 was more capable of semantic reasoning, especially when it came to context inference in intricate document structures. However, YOLOv12 was more advantageous for practical applications due to its reliable and effective structured field extraction in certificates. On the basis of their divergent styles, future hybrid models that combine the contextual understanding of transformer models with the real-time capabilities of YOLO can be visualized. In contrast to 22.4 ms for YOLOv11 and approximately 185 ms for LayoutLMv3, YOLOv12 inferred at 19.1 ms per image. This comparison shows how flexible YOLOv12 is for digitizing bulk certificates, where latency and throughput have an equal impact on accuracy.

4.7 Error Analysis

We carried out an error analysis to get a better idea of the common failure cases that showed up in the detection pipeline:

- False negatives in low-quality scans: Some small fields such as Ref No were occasionally missed in heavily degraded certificate images.

- Misclassification between Reg No and Ref No: Due to close spatial proximity and similar textual structure, minor confusion occurred between these two classes.
- Bounding box misalignment on rotated certificates: Certificates with slight rotations (5-10 degrees) sometimes led to bounding box drift, particularly for horizontally aligned fields.

In future work, we plan to focus more on making the system robust to rotations, and we also want to add semantic checks so that these kinds of errors can be reduced.

Beyond the confusion matrices, the errors were categorized to better understand their sources. Four main types were observed: Reg No ↔ Ref No confusion (3.2%), OCR misreads in low-quality or low-resolution scans (5.1%), false positives caused by logos or decorative stamps (1.8%), and bounding box drift in rotated or tilted certificates (2.5%). These findings are summarized in Table 8.

Table 8. Error Taxonomy and Frequency Observed in the Test Set

Error Type	Frequency (%)	Description
Reg No ↔ Ref No confusion	3.2	Confusion between two visually similar and adjacent fields
OCR misread (low-quality scans, diacritics)	5.1	Errors due to degraded image quality and Vietnamese text with accents
False positives (logos, decorative seals)	1.8	Background logos or stamps misclassified as text fields
Bounding box drift (rotated certificates)	2.5	Slightly tilted documents caused detection misalignment

Beyond the categorized errors shown in Table 8, an internal breakdown was carried out to quantify the relative contribution of different pipeline components. The results indicated that approximately 65% of residual errors originated from OCR misreads, 25% from detection mistakes, and 10% from post-processing inconsistencies. The dictionary-based correction and CAPTCHA/database verification steps reduced some OCR-related errors, however recognition of low-quality or multilingual text remains the main bottleneck. These findings highlight that while detection has improved considerably with YOLOv12, future work should focus on OCR enhancement and stronger semantic validation.

5. System Design and Information Retrieval Algorithm

5.1 System Workflow

Figure 9 outlines the main steps of our system for certificate information extraction. The pipeline is designed to reduce manual work by combining image preprocessing, object detection, OCR, and database matching.

- **Data preprocessing:** Many scanned certificates are not clean. Some are tilted, too dark, or have background noise. In this step, we correct the orientation, adjust brightness, and apply denoising so that the later models receive clearer input.

- **Object detection:** Once the images are cleaned, YOLO is applied to locate the key regions on the certificate. These regions include names (Vietnamese and English), serial numbers, Reg No, and Ref No. YOLO outputs bounding boxes around these fields.

- **Text extraction:** The detected regions are then passed to Tesseract OCR which converts the image areas into text. We use Tesseract because it supports multiple languages and can still work when the input quality is not ideal.

- **Post-processing:** The raw OCR output often contains extra symbols or inconsistent formatting. Post-processing removes unwanted characters, normalizes formats, and handles missing values. The result is structured text data.

- **Database matching:** In the last step, the processed text is matched with the certificate database to verify the information.

Overall, this workflow provides a working end-to-end solution. It can process certificates automatically and at scale, though challenges remain with very poor scans and non-standard templates.

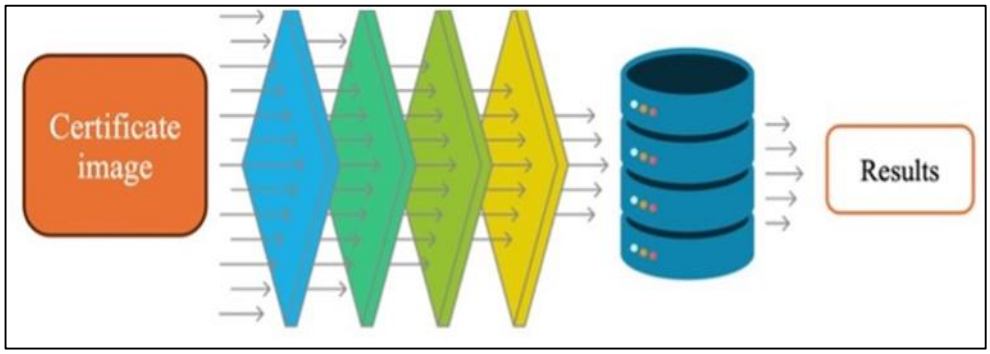

Figure 9. The Diagram of the Proposed Certificate Information Extraction System

5.2 Information Extraction and Search Algorithm

5.2.1 Algorithm Specification

Algorithm 1 shows the pipeline for certificate information extraction. It has five steps: preprocessing, detection, OCR, post-processing, and database matching. In preprocessing, the image is fixed if tilted, brightness is adjusted, and some noise is removed. Sometimes we also resize the image, the goal is just to make it clearer, not perfect.

YOLO is then used to find key regions, such as the name or registration number. Each region is cropped and sent to OCR, and the text is compiled. After a quick cleanup to remove odd symbols and fix formats, the result is compared with the certificate database. If there is a match, the system shows the extracted fields. The pipeline works on different image qualities and layouts, and combining detection with basic normalization helps reduce common OCR errors.

Algorithm 1: Certificate Information Extraction Pipeline

Input: Certificate image, Certificate database

Output: Extracted information, Retrieval result
Method:
Step 1: Image Preprocessing
Receive input image
Perform quality enhancement:
Deskew the image
Enhance brightness and contrast
Denoise the image
Resize to standard dimensions
Output: preprocessed image
Step 2: Object Detection
Load the trained YOLO model
Detect information regions
Label each region (e.g., Name, Reg No, Ref No)
Step 3: Text Extraction
for all detected regions do
Crop the region
Apply OCR to extract text
Aggregate extracted texts
Step 4: Post-Processing and Normalization
for all extracted texts do
Remove unwanted characters
Normalize the format
Handle missing or incomplete data
Create structured result
Step 5: Database Matching
if match found then
Return extracted information and retrieval result
else
Return extracted information with no match found

5.2.2 Formal Analysis of Correctness and Termination

Problem Setting and Specification

Input. A certificate image I and a finite certificate database D.

Output. A pair (Info, Result) where:

- Info is the normalized tuple of target fields (Name_vi, DOB_vi, Serial, RegNo, Name_en, DOB_en, RegNo_en, RefNo).
- Result ∈Match, NoMatch.

The pipeline consists of five sequential phases: *Preprocessing → Object Detection (YOLO) → OCR → Post-processing & Normalization → Database Matching*. Let ß be the finite set of detected regions (with class labels). For each b ∈ ß: OCR returns a string s_b, and Norm maps s_b to a normalized value according to the class label of b.

1) Partial Correctness (Correctness)

We argue phase-by-phase using invariants and pre/post-conditions.

a) Preprocessing

- **Invariant (content-preserving):** Deskew/denoise/illumination adjustments do not create or remove semantic characters; they only improve image quality.
- **Post-condition:** The resulting image I' is a valid input for detection/OCR and preserves the original textual content.

b) Object Detection (YOLO)

- **Pre-condition**: Input image is valid.
- **Post-condition:** Returns a finite set of bounding boxes with class labels from the target set (Name, RegNo, RefNo, …).
- **Loop-free phase:** No non-terminating loops; a fixed forward pass produces results.

c) OCR Loop over Detected Regions

- **Loop: for b in $\mathfrak{B}$:** crop → OCR → append.
- **Loop Invariant (Inv):** After processing the first regions, the partial structure Info correctly stores OCR outputs (possibly unnormalized) for exactly those regions; no information from processed regions is lost.
- **Initialization:** k = 0, Info_0 is empty — Inv holds.
- **Maintenance:** Processing the next region preserves Inv (we add one entry).
- **Termination/Exit:** When the loop finishes, OCR text exists for all $b \in B$.

d) Post-processing & Normalization

- **Loop:** for each extracted string: clean → normalize → handle missing → assemble.
- **Invariant:** After normalizing the first strings, Info contains normalized values in the correct format for those fields.
- **Post-condition:** Info is fully normalized and ready for matching

e) Database Matching and Return

- The matcher returns Match if there exists a record in D consistent with Info (according to the system's matching rules); otherwise it returns NoMatch.

Therefore (Partial Correctness): If detection returns correct regions and OCR+normalization behave according to their specifications on those regions, then the algorithm returns Match if and only if D actually contains a consistent record with the fields in Info (and NoMatch otherwise). Hence the output satisfies the output specification.

2) Termination

Define a natural variant that strictly decreases in every loop:

- **Detection:** A single forward pass finite computation, no loop.
- **OCR loop:** Iterates over the finite set $\mathfrak{B}$. At each iteration the number of unprocessed regions decreases by 1. Hence the loop terminates.

- **Normalization loop:** Iterates over the finite list of extracted strings; the number of unprocessed strings decreases by 1 each step. Hence it terminates.
- **Matching:** A finite query over a finite database (or a bounded-time indexed lookup). Thus it terminates.

Since the whole algorithm is a composition of finite computations and two loops over finite sets, the pipeline always terminates.

3) Time Complexity (High-Level)

Let $|\mathcal{B}|$ be the number of detected regions and the database size. The total time can be decomposed as:

$$\mathrm{T}(I) \approx T_{pre}(I) + T_{YOLO}(I) + \sum_{b \in \mathcal{B}}\big(T_{OCR}(b) + T_{norm}(b)\big) + T_{match}(Info, D)$$

- Detection cost T_{YOLO} usually dominates and depends on image resolution/model size; treat as one forward-pass cost.
- OCR + normalization is linear in $|\mathcal{B}|$.
- Matching is $\mathrm{O}(|D|)$ in the worst case (linear scan) or $\mathrm{O}(log|D|)$ with proper indexing/hash maps, plus constant-time field comparisons.
- Space complexity is linear in the number of regions and field strings maintained: $\mathrm{O}(|\mathcal{B}|)$, plus model parameters (fixed) and any indexing structures for D.

4) Conclusion

- **Correctness:** Given correct detections and faithful OCR+normalization, the algorithm's output pair (Info, Result) meets the specification.
- **Termination:** Guaranteed by the finiteness of the region set and database, and strictly decreasing loop variants.
- **Complexity:** Dominated by one detection forward pass and a linear pass over detected regions; matching can be sublinear with indexing.

5.3 System Evaluation

Figure 10. Main System Interface

To evaluate the practical effectiveness of the proposed certificate information extraction system, we implemented and tested it through a graphical user interface (GUI) that supports multiple processing modes. The system is organized into five functional tabs: Image Processing, Capture Live, PDF Processing, Enter Information, and ROI Processing. Each tab corresponds to a different method of input and verification, thereby enhancing system versatility and user accessibility.

In the Image Processing mode, users can upload certificate images from their local devices for automated analysis. Upon submission, the system performs preprocessing, object detection, OCR, and post-processing as described in previous sections.

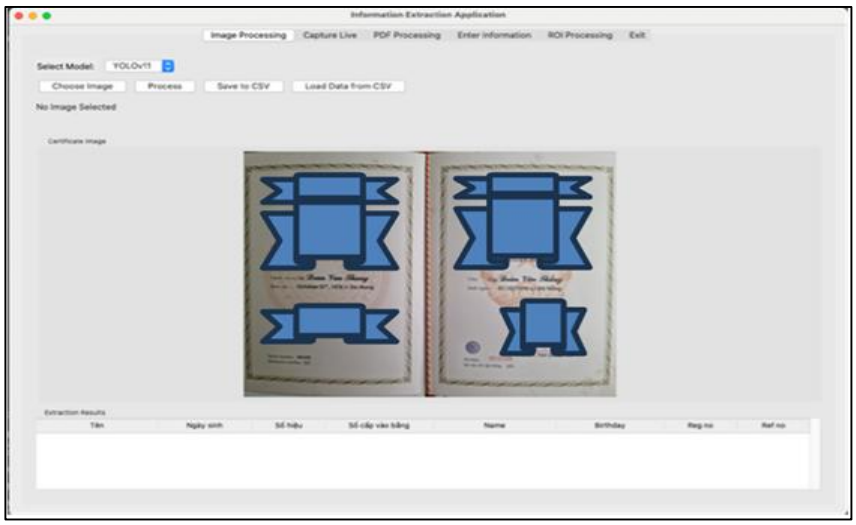

Figure 11. Certificate Upload Interface under the Image Processing Tab

The system simply verifies whether the extracted data matches anything in the certificate database and displays the record again. The primary fields, such as the recipient's name, registration number, and reference number, are listed in the result. In this manner, it is evident what was discovered and what is kept in the database (refer to Figure 12).

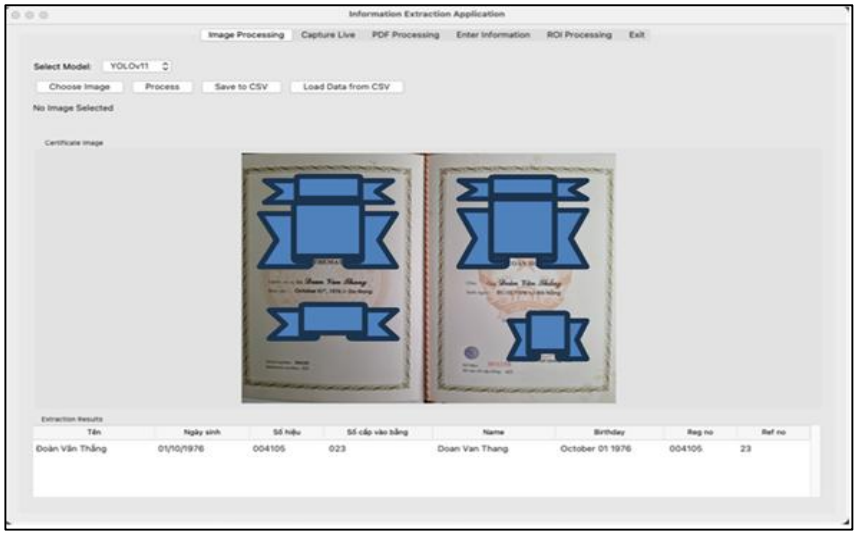

Figure 12. Displayed Result Showing Extracted and Matched Certificate Information

In contrast (Figure 13), when verifying a certificate that does not exist in the database, the system will still perform data extraction and notify the user of the failure to match.

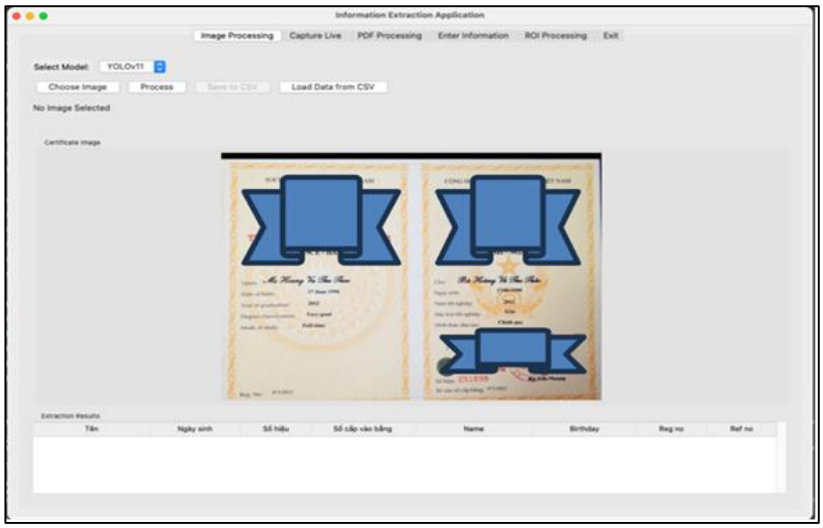

Figure 13. Certificate Image used in a Failed Matching Case

To maintain usability, the system also attempts approximate matching using key fields such as the Reg No, which can return a similar or partially matching entry from the database (refer Figure 14).

Figure 14. Notification of Approximate Match based on Reg No Field

These examples show that the system can provide useful feedback, whether the lookup works or fails. On the screen, the extracted fields and their statuses are shown clearly, making the process easy to follow and check again later. In practice, the system also works with many different kinds of inputs and continues to perform well even when the data is messy. From this evaluation, we see that the approach is practical and can be used for real certificate digitization and verification.

6. Discussion

We analyzed structured diplomas and developed a complete information extraction pipeline using YOLO, comparing versions 11 and 12. In practice, YOLOv12 showed clear advantages: its stronger feature aggregation and improved decoding step made it better at recognizing the small and closely packed elements that often appear in diploma layouts. This was reflected in higher mAP@50 and mAP@50:95 scores, particularly at stricter IoU thresholds, which confirm that YOLOv12 delivers more reliable results in real-world scenarios.

Comparing YOLO versions 11 and 12, we examined structured diplomas and created a comprehensive information extraction pipeline. In actuality, YOLOv12 demonstrated distinct advantages: it was better able to identify the tiny, densely packed elements that frequently show up in diploma layouts due to its stronger feature aggregation and enhanced decoding step. Higher mAP@50 and mAP@50:95 scores demonstrated this, especially at more stringent IoU thresholds, confirming that YOLOv12 produces more dependable outcomes in practical situations.

Additionally, during testing, we found that YOLOv12 not only converges faster during training but also provides more stable results. The approximately 15% reduction in inference time per image is indeed a boon, especially when processing certificates in real time. When looking at the confusion matrix, we also found that YOLOv12 has less background noise and less confusion in similar information fields, and a t-test confirmed that these improvements are statistically significant. Finally, compared to LayoutLMv3, YOLOv12 shows an advantage due to its good balance between speed and accuracy. However, in terms of semantic capabilities, LayoutLMv3 is more powerful but has a very high computational cost, making it difficult to apply in fast and large-scale processing scenarios.

The pipeline integrating YOLO with Tesseract OCR and database information matching shows a closed process from input image to structured data. However, if the scanned image is blurry, the information fields overlap, or the forms are not standard, there are still obstacles in the problem of information retrieval. Additionally, the current dataset is relatively small and specialized, so scaling it up and making it publicly available would be an important next step, both to improve generalizability and to contribute to the research community.

7. Conclusion

This paper presents a deep learning–driven framework for automated diploma information extraction, structured around two primary stages: object detection and data validation. Whether the database extracted from the diploma exists or if the pipeline system integrates YOLO and OCR and compares it with the database, is unclear. The experimental results show that YOLO version 12 is superior to version 11 in terms of accuracy (mAP@50, mAP@50:95), training speed, ability and latency to distinguish similar classes. The pipeline helps to reduce manual information and improve accuracy. However, there are still limitations and overlapping information fields with low-quality images, etc. The platform for large-scale diploma analysis is the final YOLO version 12, with potential applications in education, administration and law.

References

[1] Al-Qudah, Rabiah, and Ching Y. Suen. "Enhancing YOLO deep networks for the detection of license plates in complex scenes." In Proceedings of the Second International Conference on Data Science, E-Learning and Information Systems, (2019): 1-6.

[2] Zeng, Jie, Kan Wang, Xiong Hu, Yuanzhi Hu, Xi Liu, and Zhengqian Cheng. "YOLO series object detection networks optimized with luminance attention mechanism network." In Seventh International Conference on Traffic Engineering and

Transportation System (ICTETS 2023), vol. 13064, SPIE, (2024): 689-694.

[3] Chen, Bo. "Research overview of YOLO series object detection algorithms based on deep learning." Journal of Computing and Electronic Information Management 15, no. 3 (2024): 84–92. https://doi.org/10.54097/p81rtv77.

[4] He, Zijian, Kang Wang, Tian Fang, Lei Su, Rui Chen, and Xihong Fei. "Comprehensive performance evaluation of YOLOv11, YOLOv10, YOLOv9, YOLOv8 and YOLOv5 on object detection of power equipment." In 2025 37th Chinese Control and Decision Conference (CCDC), IEEE, (2025): 1281-1286.

[5] Dwivedi, Upendra, Kireet Joshi, Surendra Kumar Shukla, and Anand Singh Rajawat. "An overview of moving object detection using yolo deep learning models." In 2024 2nd International Conference on Disruptive Technologies (ICDT), IEEE, (2024): 1014-1020.

[6] Cong, Xiaohan, Shixin Li, Fankai Chen, Chen Liu, and Yue Meng. "A review of YOLO object detection algorithms based on deep learning." Frontiers in Computing and Intelligent Systems 4, no. 2 (2023): 17-20.

[7] Safaldin, Mukaram, Nizar Zaghden, and Mahmoud Mejdoub. "Moving object detection based on enhanced Yolo-V2 model." In 2023 5th international congress on human-computer interaction, optimization and robotic applications (HORA), IEEE, (2023): 1-8.

[8] Ananth, Aluri Dev, Abhiram Seemakurthi, Sasank Tumma, and Prasanthi Boyapati. "YOLO CNN Approach for Object Detection." In Algorithms in Advanced Artificial Intelligence, CRC Press, (2024): 481-486.

[9] Bhardwaj, Khushi, and T. Poongodi. "Deep learning approach for multi-object detection using yolo algorithm." In 2023 6th International Conference on Contemporary Computing and Informatics (IC3I), vol. 6, IEEE, (2023): 689-693.

[10] Boussaad, Leila, and Aldjia Boucetta. "YOLO Network-based URL Detection in Varied Conditions with Small-Sample Insights." International Journal of Informatics and Applied Mathematics 7, no. 1: 33-56.

[11] Atik, Muhammed Enes, Zaide Duran, and Roni Özgünlük. "Comparison of YOLO versions for object detection from aerial images." International journal of environment and geoinformatics 9, no. 2 (2022): 87-93.

[12] Padmane, Priyanka, Tushar Dasare, Prathamesh Deshkar, Nikhil Dasgupta, Ashish Kale, and Burhanuddin Hamdard. "A Review on Real Time Object Detection Using Deep Learning.".

[13] Patil, Suvarna, Soham Waghule, Siddhesh Waje, Prasad Pawar, and Shreyash Domb. "Efficient object detection with YOLO: a comprehensive guide." Int J Adv Res Sci Communi Technol 2022 (2024): 519-31.

[14] Ponika, Mannem, Kopalli Jahnavi, P. S. V. S. Sridhar, and Kavuri Veena. "Developing a YOLO based object detection application using OpenCV." In 2023 7th International Conference on Computing Methodologies and Communication (ICCMC), IEEE, (2023): 662-668.

[15] Sirisha, Uddagiri, S. Phani Praveen, Parvathaneni Naga Srinivasu, Paolo Barsocchi, and Akash Kumar Bhoi. "Statistical analysis of design aspects of various YOLO-based

deep learning models for object detection." International Journal of Computational Intelligence Systems 16, no. 1 (2023): 126.

[16] Huang, Yupan, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. "Layoutlmv3: Pre-training for document ai with unified text and image masking." In Proceedings of the 30th ACM international conference on multimedia, (2022): 4083-4091.

[17] Rausch, Johannes, Octavio Martinez, Fabian Bissig, Ce Zhang, and Stefan Feuerriegel. "Docparser: Hierarchical document structure parsing from renderings." In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 5, (2021); 4328-4338.

[18] Sapkota, Ranjan, and Manoj Karkee. "Improved yolov12 with llm-generated synthetic data for enhanced apple detection and benchmarking against yolov11 and yolov10." arXiv preprint arXiv:2503.00057 (2025).