Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Issue 1, Januari 2026
DOI : https://doi.org/10.33395/sinkron.v10i1.15383
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Implementation of YOLOv12 and PaddleOCR for Indonesian Bank Statement Table Extraction

Samuel Miracle Kristanto1)*, **Evan Tanuwijaya**2)
1)*2)Informatika, Fakultas Teknologi Informasi, Universitas Ciputra Surabaya, Surabaya, Jawa Timur, Indonesia.
1)*lsamuel01@student.ciputra.ac.id, 2)evan.tanuwijaya@ciputra.ac.id

Submitted :Nov 2, 2025 | **Accepted** : Nov 10, 2025 | **Published** : Jan 01, 2026

Abstract: The increasing reliance on digital financial documents has highlighted the need for automated methods to extract structured information from bank statements. Traditional optical character recognition (OCR) systems often fail to capture complex tabular structures, leading to incomplete or error-prone transaction records. To address this challenge, this research proposes a two-stage detection and recognition pipeline that combines YOLOv12 for table and structural element detection with PaddleOCR for text extraction, followed by automated Excel conversion. The objective is to develop an end-to-end approach capable of accurately identifying table regions and reconstructing their row-column structure into analyzable tabular data. The methods involve training a YOLOv12-n model in two stages: Stage 1 focuses on detecting entire table regions, while Stage 2 focuses on identifying row and column structures within the detected tables. A lightweight AdamW optimizer with conservative augmentation strategies was applied to preserve the geometric integrity of document layouts. Results show that Stage 1 achieved precision of 0.998, recall of 1.0, and mAP50-95 of 0.989, while Stage 2 achieved precision of 0.992, recall of 0.964, and mAP50-95 of 0.899, demonstrating strong localization and structural recognition. The conclusions confirm that the proposed two-stage pipeline is effective for financial document processing, with potential applications in digital banking, auditing, and automated record management. Future research may focus on expanding datasets and addressing domain-specific variability.

Keywords: Bank statements, Financial data extraction, PaddleOCR, Table detection, YOLOv12

INTRODUCTION

The acceleration of digital banking in Indonesia has heightened the need for efficient and accurate extraction of structured data from bank statements, which are fundamental for personal, business, and regulatory financial processes (Agrawal et al., 2021; H. Li, Huang, & Gu, 2021). Traditional manual processing (such as reviewing, validating, and recording transactions) remains common but is slow, susceptible to error, and unfit for large-scale, diverse statement formats (Trivedi et al., 2024; Vo-Nguyen, Nguyen, & Le, 2021).

Applying OCR alone is insufficient for these documents. While OCR can identify text, it does not capture the structural relationship between rows and columns, resulting in disorganized data that is difficult to analyze or automate downstream (Hou & Wang, 2025; Prasad, Gadpal, Kapadni, Visave, & Sultanpure, 2020; Smock, Pesala, & Abraham, 2021). Explicit detection of table regions and their structure is critical: transaction metadata, such as dates and balances, only has meaning when preserved in context. State-of-the-art studies confirm that pipelines which prioritize table and line extraction before text recognition result in far more accurate and usable outputs for financial document workflows (Hou & Wang, 2025; Trivedi et al., 2024).

Exporting to spreadsheet formats like Excel is not just a technical feature, but a business necessity. Excel remains the default tool for financial reconciliation, reporting, and data analysis in both small businesses and large enterprises. Automatic Excel export enables rapid review, fraud detection, and integration with accounting systems, streamlining operational and auditing cycles (Trivedi et al., 2024; Vo-Nguyen et al., 2021).

Recent advances in object detection have made automated document analysis increasingly feasible. The YOLO (You Only Look Once) family of detectors, particularly the latest YOLOv12, offers an effective balance

*name of corresponding author

between inference speed and localization accuracy for detecting tables and fine-grained structural elements in complex financial documents (Tian, Ye, & Doermann, 2025; Timothy & Tanuwijaya, 2024). Building upon earlier iterations, recent document-focused adaptations of YOLO have achieved strong results in document layout analysis (DLA) tasks. For example, DocLayout-YOLO (Zhao et al., 2024) adapts YOLO architectures for structured document segmentation, attaining state-of-the-art performance in detecting tables and text blocks. Likewise, YOLOv10-CBRC demonstrated high-precision layout detection in complex cultural heritage documents (Wu et al., 2025), while Beyene & Dancy (2025) incorporated YOLOv10 into a layout-aware OCR framework for unsupervised structural segmentation of archival materials. These developments collectively establish the YOLO family as a robust foundation for structured document understanding, motivating the adoption of YOLOv12 in this study for its enhanced attention-based feature aggregation and spatial reasoning optimized for dense tabular layouts. Complementing this, PaddleOCR is employed for accurate text extraction across varied and low-quality scanned typography (Du et al., 2020), ensuring a complete end-to-end pipeline for Indonesian bank statement processing.

This research proposes a dedicated, two-stage extraction pipeline for Indonesian bank statements: YOLOv12 first segments tables, rows, and columns, while PaddleOCR extracts cell-level text. To optimize performance across heterogeneous templates, a class-merging strategy is used, and the outputs are benchmarked for both detection accuracy and Excel usability. By focusing on structure-first extraction and Excel-ready exports, this work bridges the gap between academic solutions and the real expectations of financial professionals, advancing both the technology and its practical relevance for the banking industry.

LITERATURE REVIEW

Extracting structured data from bank statements demands high accuracy in both table detection and internal structure recognition to maintain the integrity of transactional information such as dates, descriptions, amounts, and balances (Prasad et al., 2020; Smock et al., 2021). CascadeTabNet introduced a two-stage CNN that detects tables and segments their structure, achieving an F1 score of 0.91 for table region detection on PubTables-1M, though it is computationally intensive and unsuitable for real-time applications (Prasad et al., 2020). Vo-Nguyen et al. (2021) later proposed an image-based method for detecting tables in bank statements using conventional models such as Faster R-CNN, achieving reliable detection on simple layouts but limited performance on multi-column statements due to lack of structural segmentation.

TabSniper advanced this field by introducing a transformer-based DETR model trained on BankTabNet, a financial statement dataset. It reached 0.93 mAP@50 for joint table and cell structure recognition but required over 250 ms per image inference time, reducing practicality for large-scale automation (Trivedi et al., 2024). Similarly, TABLET (Hou & Wang, 2025) employed an encoder-only transformer for end-to-end table recognition, achieving 0.94 structural F1 across diverse layouts but facing the same computational overhead typical of transformer models. Meanwhile, GFTE (Y. Li et al., 2020) combined CNN-based spatial encoding with rule-based parsing, reaching an F1 of 0.89 on financial tables and moderate inference speed.

To overcome these limitations, single-stage detectors such as the YOLO family have gained attention for document layout analysis (DLA). YOLOv12, incorporating attention-based spatial reasoning, achieves 48.0 mAP on COCO with an inference speed of 2.61 ms per image, demonstrating a favorable trade-off between precision and speed (Tian et al., 2025). Document-specific variants further validate YOLO's applicability to layout tasks. DocLayout-YOLO (Zhao et al., 2024) extends YOLO with document-specific pretraining (DocSynth-300K) and a global-to-local receptive module, achieving 0.91 mAP with reduced latency compared to transformer-based methods. Similarly, YOLOv10-CBRC (Wu et al., 2025) achieved 0.96 precision and 0.94 recall on complex historical document layouts, outperforming YOLOv8 by roughly 3% in layout detection accuracy. In parallel, Beyene & Dancy (2025) integrated YOLOv10 into a layout-aware OCR framework for unsupervised evaluation of archival documents, confirming the model's adaptability to structural segmentation tasks.

These developments demonstrate the rapid evolution of YOLO-family models from generic object detection to specialized DLA pipelines. Given this trajectory, YOLOv12 represents a natural progression, combining the speed and simplicity of YOLOv10 with enhanced feature attention and grid-awareness. Its architecture is particularly suited for detecting rectangular table boundaries and fine-grained structures in dense financial documents, making it well-aligned for this study's Stage 1 (table detection) and Stage 2 (row/column recognition) pipeline.

On the OCR side, PaddleOCR was chosen over alternatives due to its hybrid design combining DBNet for text detection and CRNN for recognition, providing superior robustness on noisy, skewed, or multilingual scanned documents (Du et al., 2020; Mursari & Wibowo, 2021). Comparative studies show that PaddleOCR achieves 5-8% higher word-level accuracy and up to 2 times faster inference than Tesseract under low-quality or rotated text conditions (Kannaopat U, 2025). Its ability to run securely in offline mode also supports privacy-preserving deployments required in financial applications.

Although research specifically addressing Indonesian bank statements remains limited, early OCR studies (Mursari & Wibowo, 2021) emphasize the need for custom detection pipelines to handle unique local bank layouts. Transformer-based systems such as TabSniper demonstrate strong detection accuracy but lack the real-time efficiency required for banking workflows. The integration of YOLOv12 and PaddleOCR in this study bridges this gap, offering a faster and more scalable solution for structured financial data extraction.

METHOD

This research follows a sequential pipeline starting from data collection and annotation through to structured output in Excel format. The bank statements were preprocessed into consistent image format and annotated into three classes: table, row, and column. Then in the first stage, a YOLOv12 model is trained to detect and crop table regions from entire bank statement pages. These cropped regions are then passed to a second YOLOv12 model specialized for detecting rows and columns, enabling explicit table structure recognition. The detected structural grid guides PaddleOCR in performing text recognition at the cell level. Finally, the recognized text is aligned with the detected table schema and exported into structured Excel files, ensuring usability for financial reconciliation and auditing tasks. The complete research pipeline is illustrated in Figure 1.

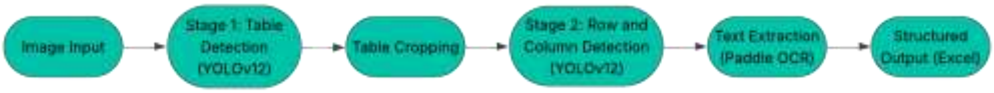
Image Input
Stage 1: Table Detection (YOLOv12)
Table Cropping
Stage 2: Row and Column Detection (YOLOv12)
Text Extraction (Paddle OCR)
Structured Output (Excel)

Fig. 1 Table Extraction Pipeline

Data Collection and Annotation

The dataset used in this study was curated from 245 personal Indonesian bank statements, collected by the first author between May 2022 and July 2025. The dataset encompasses diverse layout templates from multiple Indonesian financial institutions, including Bank Central Asia (BCA), Bank Rakyat Indonesia (BRI), and Bank Mandiri, each featuring distinct grid structures, header placements, and typography styles. This diversity was intentionally incorporated to improve model robustness against cross-bank format variation and enhance generalization to unseen layouts.

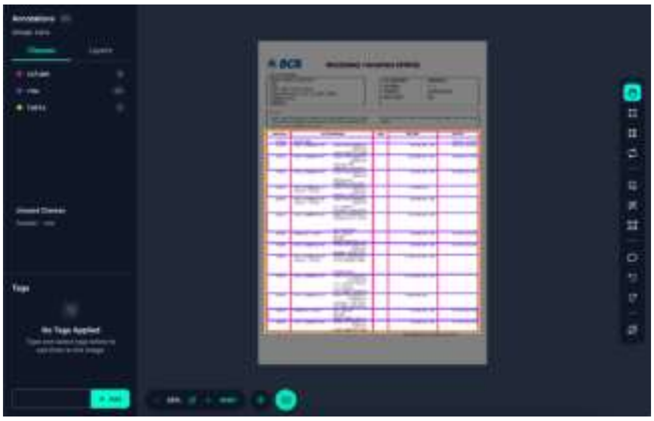

Fig. 2 Dataset Annotating Process with Roboflow

Each statement was rasterized into page-level images, which served as the basis for annotation. Annotation was conducted using Roboflow, where three primary categories were defined: table, row, and column. These classes were chosen to balance granularity with practicality, enabling the system to model document structure while avoiding excessive fragmentation of label space. A sample of the annotation interface is provided in Figure 2, where bounding boxes are visible over transaction grids and table structures. To ensure label integrity prior to training, all YOLO-format boxes were validated and minimally cleaned: class IDs were read from the dataset YAML without hard-coding, coordinates were clamped to the valid [0,1] range, zero-area boxes were discarded, and missing image/label pairs were reported and reconciled.

To maintain experimental rigor, a document-level split was enforced. Entire statements were partitioned into 80% training, 10% validation, and 10% testing subsets. This partitioning strategy is a widely adopted standard in deep learning for balancing the training pool against unbiased evaluation (Bai et al., 2021).

*name of corresponding author

Stage 1: Table Detection

Preprocessing: Constructing a Table-Only Dataset

Since the goal of Stage 1 is whole-table detection, the dataset was programmatically re-organized into a table-only subset. This involved copying each image and creating a corresponding YOLO-format label file containing only table instances, all remapped to class ID 0 (“table”). This transformation ensured that the Stage 1 detector learned solely the localization of table boundaries without being confounded by row/column labels, which are reserved for Stage 2. The resulting dataset hierarchy (train/valid/test) was linked to a dedicated Ultralytics YAML configuration, defining one class (“table”) and three splits.

Exploratory Data Analysis (EDA)

Before model training, an exploratory data analysis (EDA) was performed on the full-page dataset. This provided insight into the statistical properties of the data, guiding hyperparameter selection and model design. The conducted analyses are table bounding-box area (% of image) to measure the fraction of each page occupied by a detected table, revealing that tables consistently occupy 40–65% of the image area and then aspect ratio distribution (w/h) to reveal clustering around landscape-oriented grids, consistent with transaction records.

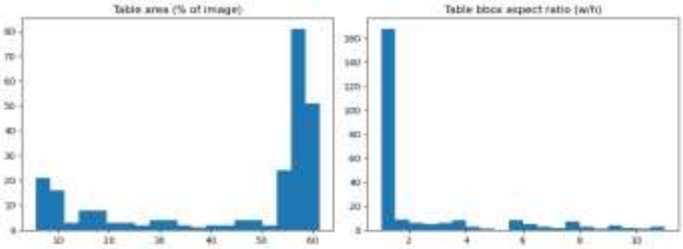

Fig. 3 Stage 1 EDA Histograms

Representative histograms are provided in Figure 3, while qualitative overlays of ground-truth table boxes on random training samples are shown in Figure 4. Together, these checks validated that the dataset was structurally consistent and suitable for training a dedicated table detector.

Fig. 4 Ground-Truth Table Bounding Box Overlay Sample for Training

Data Augmentation

Document tables have rigid, axis-aligned structure; rotations, shear, and perspective can corrupt the semantics encoded by row/column orthogonality. Prior document-layout work (PubLayNet; TableBank) shows that geometric misalignment is a primary failure mode for table detectors, so augmentation was restricted to mild, geometry-preserving transforms that mimic realistic scanning noise while avoiding structural drift (M. Li et al., 2020; Zhong, Tang, & Yepes, 2019). The final policy is intentionally conservative to stabilize training on a small, homogeneous dataset of bank statements. As shown in Table 1, this restrained policy aligns with broader document-analysis practice: preserve grid integrity, introduce only realistic variability.

Table 1
Stage 1 Data Augmentation Strategy

Augmentation Type	Value	Reasoning	Reference
Translation	0.02	Models slight page/frame misalignment common in scanning without shifting tables off canvas.	(M. Li et al., 2020; Zhong et al., 2019)
Scaling	0.15	Captures moderate variation in printed table size while keeping line thicknesses readable at 640 px.	(M. Li et al., 2020)

Training Configuration

A YOLOv12-n detector was used for Stage 1 to balance speed and accuracy on a single-class task (table). Settings follow standard YOLO practices with adjustments for document pages and the dataset's size, as shown in Table 2. Rectangular batching and a lower initial learning rate were chosen to improve curve smoothness and convergence stability on small data.

Table 2
Stage 1 Training Configuration

Parameter Type	Value	Reasoning	Reference
Image size	640×640	Default YOLO resolution; adequate to retain table lines while keeping throughput high.	(Terven & Cordova-Esparza, 2024)
Batch size	16	Usually used for standard YOLO training for document images of this resolution.	(M. Li et al., 2020)
Optimizer	AdamW (lr_0 = 0.003, lrf = 0.1, momentum = 0.937, weight_decay = 1×10^{-4})	Lower start LR and lighter regularization stabilize learning on small data; AdamW is robust for detection backbones.	(Terven & Cordova-Esparza, 2024)
Epochs	30, patience 10	Enough steps for convergence with the lower LR; patience guards against overfit.	-

Table Cropping for Stage 2

The trained Stage 1 detector was then applied to generate cropped table regions as inputs for Stage 2 (row/column detection). Cropping was implemented with a confidence cascade where primary predictions were accepted if detected with conf ≥ 0.50 (IoU = 0.50); if no boxes were detected, a fallback conf = 0.35 was used; if still empty, the system fell back to ground-truth boxes to maintain alignment for Stage 2 training. This design ensured maximal coverage of training/validation/test splits while reflecting real-world detector performance. Cropped outputs were stored separately into ground-truth crops (training/validation) and predicted crops (Stage 2 testing and end-to-end evaluation), enabling a clean distinction between supervised learning material and deployment scenarios.

Stage 2: Row & Column Detection

Preprocessing and label projection

The Stage 2 dataset is constructed by projecting full-page structure labels into the coordinate frame of each table crop. For every full-page image in each split, the pipeline identifies the corresponding table box, reads all structure boxes (column and row), keeps only those with non-zero intersection with the table box, clips them to the crop bounds, shifts them into crop-local coordinates, and writes normalized YOLO labels.

Exploratory Data Analysis (EDA)

Stage 2 operates on table crops, not full pages. Table crops are generated from the Stage 1 ground-truth boxes for every split, so the original 80/10/10 on train/validation/test partition is preserved naturally at the crop level. Each crop inherits only the structure annotations that intersect its table extent, so the label space is reduced to only column and row class. Very small fragments are removed using an area filter of 0.0002 of crop area, which prevents spurious slivers along grid lines from dominating training.

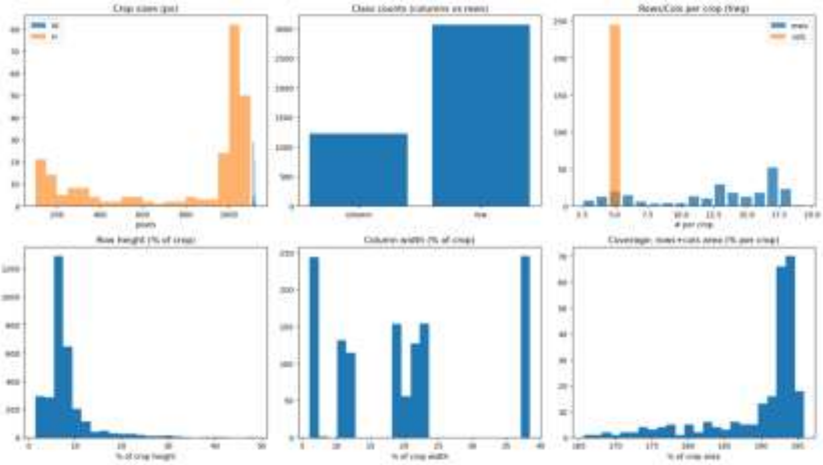

Fig. 5 Stage 2 EDA Histograms

An exploratory data analysis (EDA) pass was conducted on the derived table-crop dataset to characterize its structure and inform training choices. As shown in Figure 5, crop resolution statistics, per-class instance counts, and per-image object counts are summarized together with diagnostics specific to tabular data. These include the distribution of row heights as a percentage of crop height, the distribution of column widths as a percentage of crop width, the overall labeled area coverage (rows and columns) per crop, and a generic box aspect-ratio histogram. The histograms confirm that rows consistently appear as short horizontal bands, while columns are tall vertical bands, but that both counts and widths/heights vary considerably across different bank statement templates. This justified the use of a higher input resolution during training and the adoption of conservative, geometry-preserving augmentation to avoid corrupting line integrity. To further validate dataset quality, representative samples with overlaid ground-truth row and column bounding boxes are presented in Figure 6.

Fig. 6 Ground-Truth Row/Column Bounding Box Overlay Sample for Training

Data Augmentation

Table structure detection on rows and columns is highly sensitive to distortions that disrupt alignment, since both horizontal and vertical band consistency directly encode semantics. Prior works emphasize that geometry-preserving augmentation is essential for table structure recognition, while still introducing limited variability to improve generalization. Therefore, Stage 2 augmentation was carefully restricted to conservative transformations suited to narrow bands and fine-grained grid lines, as summarized in Table 3.

Table 3
Stage 2 Data Augmentation Strategy

Augmentation Type	Value	Reasoning	Reference
Translation	0.03	Simulates subtle shifts of rows/columns within crops, reflecting realistic OCR scanning misalignments.	(M. Li et al., 2020; Zhong et al., 2019)

*name of corresponding author

Scaling	0.2	Accounts for variation in row height and column width across different bank templates; avoids excessive rescaling that would distort band ratios.	(M. Li et al., 2020)
Horizontal Flip	0.1	Provides symmetry augmentation; improves generalization in models trained on limited layouts, while vertical flips were excluded to avoid semantic inversion.	(Xu et al., 2022; Zhong et al., 2019)

Training Configuration

For Stage 2, a YOLOv12-n model was again adopted, but training parameters were adjusted to accommodate the higher input resolution and denser object counts characteristic of row/column datasets. Table 4 outlines the hyperparameters and their rationale.

Table 4
Stage 2 Training Configuration

Parameter Type	Value	Reasoning	Reference
Image size	1024×1024	Higher resolution retained thin lines and narrow columns critical for structure parsing; also aligns with prior document detection benchmarks.	(Terven & Cordova-Esparza, 2024)
Batch size	8	Reduced to accommodate larger input resolution on device memory, while maintaining stable gradient updates.	(M. Li et al., 2020)
Optimizer	AdamW ($lr_0 = 0.01$, $lrf = 0.1$, momentum = 0.937, weight_decay = 5×10^{-4})	Stable optimizer for detection tasks, reported effective in YOLO-based document detectors.	(Hou & Wang, 2025; Terven & Cordova-Esparza, 2024)
Epochs	150, patience 30	Longer training than Stage 1 enabled convergence for high-density crops; early stopping prevented overfitting.	-
Loss weights	box = 7.5, cls = 0.5, dfl = 1.5	Prioritized localization over classification, consistent with findings in CascadeTabNet and CDeC-Net for document tables.	(Agarwal, Mondal, & Jawahar, 2020; Prasad et al., 2020)

Text Extraction with PaddleOCR

Following structure detection, each detected row-column cell was cropped into an individual bounding box and passed to the OCR engine, and when multiple text lines were recognized within the same cell, they were concatenated while the recognition confidence was averaged across tokens to provide a stable reliability estimate.

Since PaddleOCR is an extensively benchmarked third-party OCR framework (Du et al., 2020), this study adopts its pretrained model without additional re-evaluation, focusing instead on how detection accuracy influences text extraction quality in the overall pipeline.

Structured Output Mapping

Once text was extracted, the outputs were mapped into a structured table guided by the bounding boxes predicted in the structure detection stage. Rows were ordered from top to bottom, and columns from left to right, following the centroids of the bounding boxes. To improve consistency, a lightweight merging strategy was applied to remove duplicate or fragmented detections by combining overlapping boxes based on IoU thresholds and small positional gaps. In templates where the number of columns was fixed, column centers were normalized using clustering to correct spacing irregularities, while closely spaced horizontal bands were merged to avoid splitting single rows into multiple segments.

Excel Conversion

The recognized content was then converted into spreadsheet format. Each bank statement page was exported into an .xlsx file with sequential column labels, and a combined workbook was also generated to consolidate results across multiple pages into a single sheet. This combined output included provenance information such as page identifiers to preserve traceability. For columns containing financial values such as debit, credit, or balance, additional normalization was performed to ensure compatibility with standard accounting software. Non-breaking spaces and thousand separators were removed, decimal formats were standardized from 1.234,56 to 1234.56, and values were coerced into numeric types wherever possible. These post-processing steps ensured that the extracted data was not only human-readable but also ready for machine-based reconciliation, auditing, and further financial analysis.

Ethical and Privacy Considerations

This pipeline was designed with privacy and data security as core principles. All processing occurs entirely offline, ensuring that no document images or extracted text are transmitted, stored, or shared beyond the user's local environment. The YOLOv12 and PaddleOCR models operate in a stateless manner, meaning they do not learn from or retain any information after inference. Each input document is processed independently, and once the extraction is complete, the system produces only the structured Excel output without saving intermediate data. These safeguards ensure that sensitive financial information, such as account details, transaction history, or personal identifiers, remains fully private and under the user's control, aligning with responsible data-handling and ethical AI deployment standards.

RESULT

Stage 1 Model Performance Evaluation

The YOLOv12-n model trained for single-class detection of tables demonstrated strong performance in localizing tabular regions with minimal false positives or missed detections. The detector achieved a precision of 0.998 and recall of 1.0, showing that it was able to identify nearly every table instance without introducing spurious detections. The mean Average Precision further supports this result, with mAP@50 = 0.995 and a stricter mAP@50–95 = 0.989, confirming robust generalization across different overlap thresholds. Together, these metrics indicate that the Stage 1 detector could effectively isolate table regions as a preprocessing step before moving into row and column structure detection in Stage 2.

Figure 7 presents the Stage 1 loss and performance metric curves. Training and validation losses for bounding box regression, classification, and distributional focal loss all showed steady declines, confirming stable convergence without significant overfitting. Similarly, the precision, recall, and mAP curves improved rapidly in the early epochs and remained stable thereafter, underscoring the robustness of the trained detector.

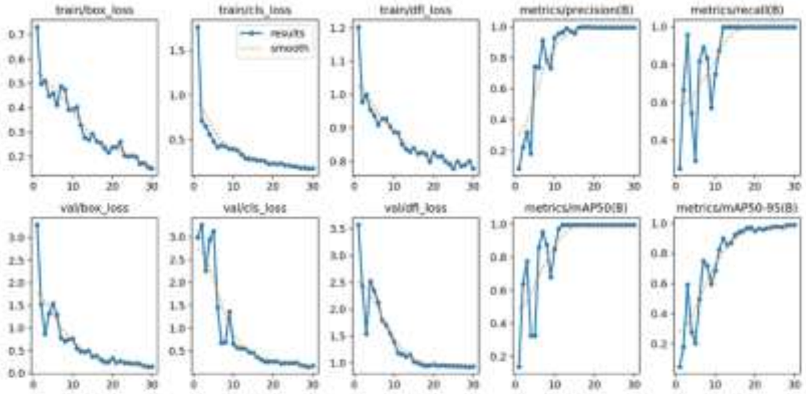

Fig. 7 Stage 1 Loss and Performance Metric Curves

Figure 8 shows the F1 score progression across epochs, where initial fluctuations were expected due to the limited single-class setup but stabilized quickly as training progressed. By later epochs, the detector consistently converged toward high F1 values, illustrating its ability to balance precision and recall effectively. The average detection confidence across test predictions was 0.72, which aligns with the high precision score and reflects the model's reliability in assigning strong confidence values to true table detections.

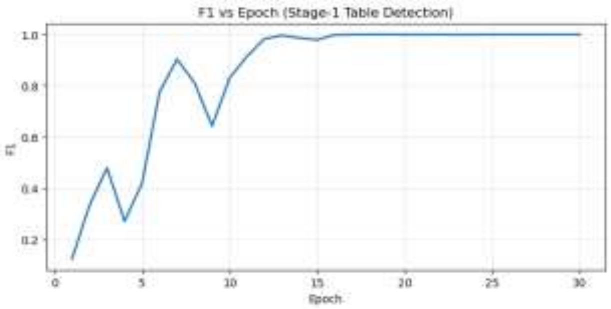

Fig. 8 Stage 1 F1 vs Epoch Curve

Figure 9 illustrates the Stage 1 detection outputs across the Indonesian bank templates (BCA, Mandiri, and BRI), highlighting the model's ability to generalize across distinct statement formats. Despite noticeable variation in header design, typography, and table grid density, the YOLOv12-n detector consistently localized the table regions with high precision, maintaining strong bounding alignment even on narrow or multi-page layouts.

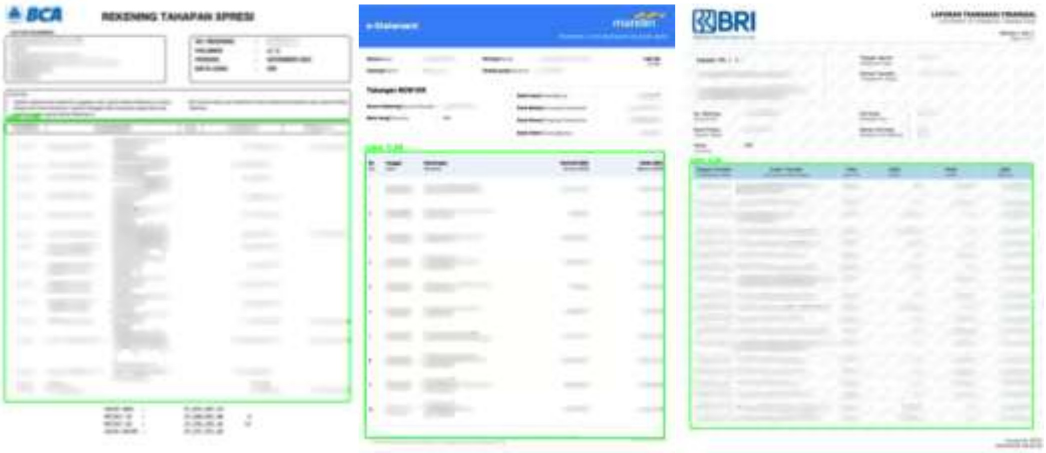

Fig. 9 Stage 1 Test Output Samples

Stage 2 Model Performance Evaluation

Stage 2 results (Table 5) further illustrate the pipeline's ability to capture table structure. Precision remained strong across both row and column classes, with recall slightly lower for columns, suggesting that tall, narrow bounding boxes were more challenging for the model compared to horizontal row segments. The mAP scores across IoU thresholds highlight that the detector not only performed well under lenient overlap criteria but also maintained reasonable quality under stricter conditions.

Table 5
Stage 2 Training Results

Class	Images	Instances	Box(Precision)	Box(Recall)	Box(mAP50)	Box(mAP50-95)
all	24	410	0.992	0.964	0.985	0.899
column	24	120	0.996	1	0.995	0.973
row	24	290	0.988	0.928	0.975	0.824

Training dynamics are further illustrated in Figure 10, where loss curves consistently decline across all components (box, classification, DFL) and validation metrics rise steadily, underscoring stable optimization.

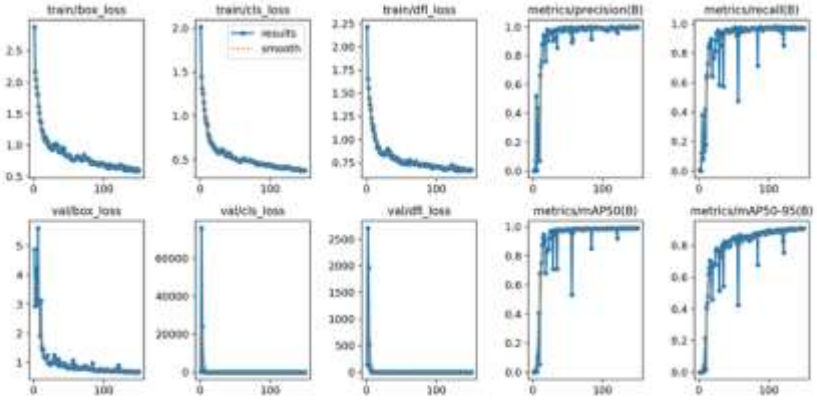

Fig. 10 Stage 2 Loss and Performance Metric Curves

Figure 11 plots F1 against training epochs, showing rapid improvement within the first 20 epochs and stable convergence near the maximum value thereafter. This suggests that the model quickly learned discriminative features for row and column segmentation and avoided overfitting.

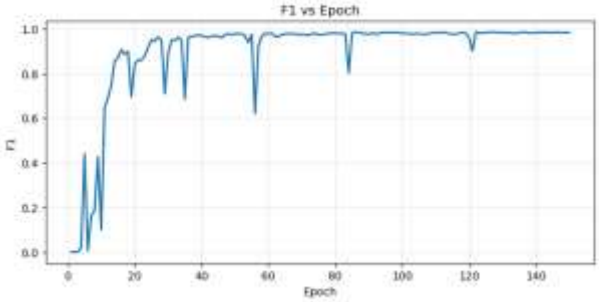

Fig. 11 Stage 2 F1 vs Epoch Curve

Figure 12 provides a breakdown of precision per class, with near-perfect performance for both rows and columns, reinforcing the model's accuracy in differentiating structural elements. It also shows the average confidence levels, with rows slightly lower than columns, indicating the model was somewhat less certain in consistently labeling rows, which is a likely consequence of variability in row height across templates.

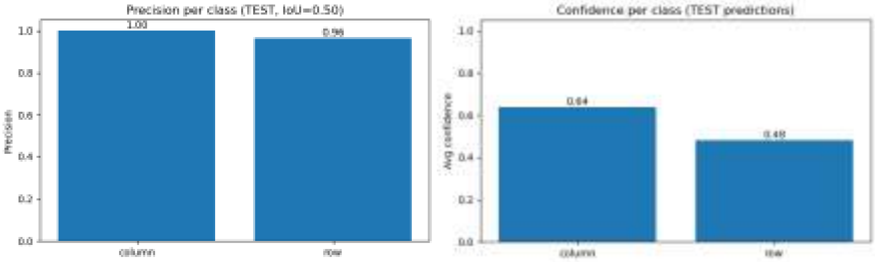

Fig. 12 Stage 2 Precision per Class & Confidence per Class Histogram

As shown in Figure 13, qualitative outputs of Stage 2 detection demonstrate clear bounding box overlays aligned with table rows and columns, visually confirming the detector's precision.

Fig. 13 Stage 2 Test Output Samples

PaddleOCR & Excel Conversion Evaluation

As shown in Figure 14, the integration of PaddleOCR and Excel conversion successfully bridged raw document images with structured digital records. Transaction dates, descriptions, debit/credit amounts, and balances were automatically extracted and arranged into their respective columns, producing a machine-readable format suitable for further analysis. While occasional OCR noise or formatting inconsistencies may still occur, the pipeline demonstrates strong potential for automating financial document digitization, eliminating much of the manual effort typically required for data entry. This highlights the practical utility of combining detection and OCR models into an end-to-end workflow for real-world banking data processing.

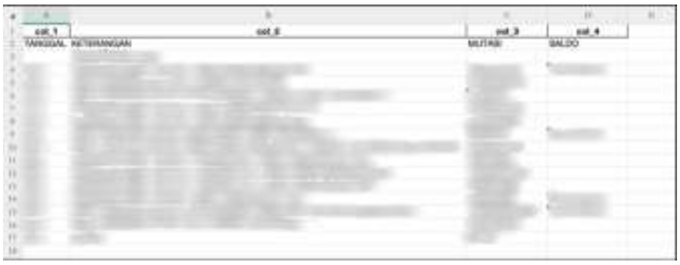

Fig. 14 Excel Conversion Result Sample

DISCUSSIONS

This study evaluated a two-stage YOLOv12-based pipeline for financial table extraction, focusing first on full-table detection and then on row/column structure recognition. In Stage 1, the model achieved nearly perfect localization with precision = 0.998, recall = 1.0, mAP@50 = 0.995, and mAP@50–95 = 0.989, confirming that YOLOv12-n is highly effective for rigid-layout detection tasks where orthogonality carries semantic meaning. These results exceed the F1 ≈ 0.91 reported by CascadeTabNet (Prasad et al., 2020) and align with the structural-localization emphasis of CDeC-Net (Agarwal et al., 2020). The smooth convergence curves indicate that, with moderate augmentation and optimized anchor scaling, YOLOv12 maintains the stability observed in recent YOLOv10 and YOLOv11 implementations applied to document layouts (Terven & Cordova-Esparza, 2024; Tian et al., 2025), demonstrating the maturity of single-stage architectures for structured document understanding.

Beyond numeric metrics, the model generalized well across three Indonesian bank templates (BCA, Mandiri, and BRI) that differ in header style, typography, and grid alignment. The YOLOv12-n consistently localized table regions with high precision despite structural heterogeneity, confirming strong cross-template adaptability. This behavior echoes prior observations that statement formats differ widely in layout density and column spacing (Trivedi et al., 2024; Vo-Nguyen et al., 2021). Including multiple institutions increased representational diversity, aligning with Mursari & Wibowo (2021), who showed that localized preprocessing improves OCR and detection accuracy in Indonesian financial documents.

Stage 2 delivered high-quality segmentation with column precision of 0.996 / mAP@50–95 = 0.973 and row precision of 0.988 / mAP@50–95 = 0.824, demonstrating that the model can reliably recover structural grids necessary for downstream OCR alignment. The slight asymmetry between row and column accuracy reflects intrinsic layout characteristics. Columns span the vertical axis, forming stable geometric boundaries that facilitate learning (Hou & Wang, 2025; Xu et al., 2022), while rows are susceptible to text wrapping, irregular spacing, and varied transaction density, which are challenges similarly identified in TabSniper (Trivedi et al., 2024) and GFTE (Y. Li et al., 2020). These findings reinforce that table parsing is structurally non-uniform: row detection inherits more semantic noise from textual content, whereas column boundaries are geometrically consistent.

In context, this research extends the progression of document-adapted YOLO architectures. Earlier works such as DocLayout-YOLO (Zhao et al., 2024), YOLOv10-CBRC (Wu et al., 2025), and Beyene & Dancy (2025) established the effectiveness of YOLOv10 for document layout analysis, achieving high layout precision (mAP@50–95 ≈ 0.86-0.90) on heritage or archival datasets. However, those studies targeted heterogeneous page segmentation rather than dense, financial table layouts. YOLOv12 introduces architectural refinements with improved attention-based feature fusion and gradient-path optimization that enhance localization under high-density, grid-structured conditions. While this paper does not provide a direct quantitative benchmark against other YOLO family, the achieved 0.989 mAP@50–95 (Stage 1) and 0.899 mAP@50–95 (Stage 2) suggest that YOLOv12 attains comparable or superior structural detection performance with smaller computational overhead. This supports the trend that newer YOLO generations achieve better efficiency–accuracy balance for document-layout tasks.

When compared to broader literature, the results strengthen the consensus that lightweight convolutional detectors can match or outperform complex transformer or graph-based architectures while retaining real-time throughput (Smock et al., 2021; Terven & Cordova-Esparza, 2024). Moreover, integrating PaddleOCR for text recognition bridges structural segmentation with content extraction, producing structured, machine-readable Excel outputs (Du et al., 2020; Vo-Nguyen et al., 2021). This combination demonstrates the pipeline's end-to-end practicality for applications such as automated reconciliation and financial auditing (Agrawal et al., 2021; H. Li et al., 2021).

Nevertheless, several limitations remain. The dataset, while incorporating multiple banks, remains modest in scale and limited to Bahasa Indonesia-formatted statements, which may restrict generalizability to multilingual or highly stylized templates (Mursari & Wibowo, 2021). Confidence scores in Stage 1 averaged ~0.72, reflecting conservative thresholding that could reduce recall under degraded scans. Furthermore, Stage 2 focused on row-column detection only, without modeling merged or nested subtables which is a challenge noted by Hou & Wang (2025) as central to next-generation table extraction. Expanding dataset diversity and benchmarking YOLOv12 against prior YOLO variants in cross-institution and multilingual contexts would be a logical next step toward full scalability and domain generalization.

CONCLUSION

This study demonstrates that a two-stage pipeline combining YOLOv12 for structural detection and PaddleOCR for text recognition can effectively automate data extraction from Indonesian bank statements. The system achieved high accuracy in table localization (precision = 0.998, recall = 1.0, mAP@50–95 = 0.989) and reliable structure segmentation (precision = 0.992, mAP@50–95 = 0.899), confirming that lightweight object detectors can deliver precise and consistent results on dense, grid-based financial layouts.

By incorporating templates from BCA, Mandiri, and BRI, the model proved robust across different statement designs and typographies, showing strong adaptability to layout variations that are typical in real-world financial documents. This reinforces that structural consistency, rather than dataset size alone, is key to achieving dependable performance across institutions.

Compared with prior research using earlier YOLO variants and transformer-based models such as CascadeTabNet, TabSniper, and TABLET, the proposed pipeline provides a more efficient, end-to-end approach that balances accuracy and inference speed. Its integration of PaddleOCR further ensures that extracted results are readily exportable into structured, machine-readable formats such as Excel, bridging the gap between academic methods and operational financial workflows.

Overall, the findings highlight that YOLOv12-based document detectors can form a scalable foundation for bank-statement automation, offering accuracy, efficiency, and adaptability for future applications in digital banking, auditing, and financial data management.

ACKNOWLEDGMENT

The acknowledgments are given at the end of the research paper and should at a minimum name the sources of funding that contributed to the article. You may also recognize other people who contributed to the article or data contained in the article but at a level of effort that does not justify their inclusion as authors. You may also state the research grant contract number if any.

REFERENCES

Agarwal, M., Mondal, A., & Jawahar, C. V. (2020). *CDeC-Net: Composite Deformable Cascade Network for Table Detection in Document Images*. https://doi.org/10.48550/arXiv.2008.10831

Agrawal, P., Chaudhary, D., Madaan, V., Zabrovskiy, A., Prodan, R., Kimovski, D., & Timmerer, C. (2021). Automated bank cheque verification using image processing and deep learning methods. *Multimedia Tools and Applications*, *80*(4), 5319–5350. https://doi.org/10.1007/s11042-020-09818-1

Bai, Y., Chen, M., Zhou, P., Zhao, T., Lee, J. D., Kakade, S., … Xiong, C. (2021). *How Important is the Train-Validation Split in Meta-Learning?* https://doi.org/10.48550/arXiv.2010.05843

Beyene, F. S., & Dancy, C. L. (2025). *Layout-Aware OCR for Black Digital Archives with Unsupervised Evaluation*. https://doi.org/10.48550/arXiv.2509.13236

Du, Y., Li, C., Guo, R., Yin, X., Liu, W., Zhou, J., … Wang, H. (2020). *PP-OCR: A Practical Ultra Lightweight OCR System*. https://doi.org/10.48550/arXiv.2009.09941

Hou, Q., & Wang, J. (2025). *TABLET: Table Structure Recognition using Encoder-only Transformers*. https://doi.org/10.48550/arXiv.2506.07015

Kannaopat U. (2025, July 28). Paddle OCR vs Tesseract (OCR Features Comparison). Retrieved September 19, 2025, from https://ironsoftware.com/csharp/ocr/blog/compare-to-other-components/paddle-ocr-vs-tesseract/

Li, H., Huang, C., & Gu, L. (2021). Image pattern recognition in identification of financial bills risk management. *Neural Computing and Applications*, *33*(3), 867–876. https://doi.org/10.1007/s00521-020-05261-3

Li, M., Cui, L., Huang, S., Wei, F., Zhou, M., & Li, Z. (2020). *TableBank: A Benchmark Dataset for Table Detection and Recognition*. https://doi.org/0.48550/arXiv.1903.01949

Li, Y., Huang, Z., Yan, J., Zhou, Y., Ye, F., & Liu, X. (2020). *GFTE: Graph-based Financial Table Extraction*. https://doi.org/10.48550/arXiv.2003.07560

Mursari, L. R., & Wibowo, A. (2021). The Effectiveness of Image Preprocessing on Digital Handwritten Scripts Recognition with The Implementation of OCR Tesseract. *Computer Engineering and Applications Journal*, *10*(3), 177–186. https://doi.org/10.18495/comengapp.v10i3.386

Prasad, D., Gadpal, A., Kapadni, K., Visave, M., & Sultanpure, K. (2020). *CascadeTabNet: An approach for end to end table detection and structure recognition from image-based documents*. https://doi.org/10.48550/arXiv.2004.12629

Smock, B., Pesala, R., & Abraham, R. (2021). *PubTables-1M: Towards comprehensive table extraction from unstructured documents*. https://doi.org/10.48550/arXiv.2110.00061

Terven, J., & Cordova-Esparza, D. (2024). *A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS*. https://doi.org/10.3390/make5040083

Tian, Y., Ye, Q., & Doermann, D. (2025). *YOLOv12: Attention-Centric Real-Time Object Detectors*. https://doi.org/10.48550/arXiv.2502.12524

Timothy, & Tanuwijaya, E. (2024). Dangerous Objects Detection and Segmentation in X-Ray Images of Passenger Goods Using YOLOV8. *2024 2nd International Conference on Technology Innovation and Its Applications (ICTIIA)*, 1–6. IEEE. https://doi.org/10.1109/ICTIIA61827.2024.10761162

Trivedi, A., Mukherjee, S., Singh, R. K., Agarwal, V., Ramakrishnan, S., & Bhatt, H. S. (2024). *TabSniper: Towards Accurate Table Detection & Structure Recognition for Bank Statements*. https://doi.org/10.48550/arXiv.2412.12827

Vo-Nguyen, T.-A., Nguyen, P., & Le, H.-S. (2021). An Efficient Method to Extract Data from Bank Statements Based on Image-Based Table Detection. *2021 15th International Conference on Advanced Computing and Applications (ACOMP)*, 186–190. IEEE. https://doi.org/10.1109/ACOMP53746.2021.00033

Wu, Z., Wang, W., & Li, H. (2025). YOLOv10-CBRC: A high-precision document image layout analysis model. *Journal of King Saud University Computer and Information Sciences*, *37*(6), 145. https://doi.org/10.1007/s44443-025-00168-2

Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., … Zhou, L. (2022). *LayoutLMv2: Multi-modal Pre-training for Visually-Rich Document Understanding*. https://doi.org/10.48550/arXiv.2012.14740

Zhao, Z., Kang, H., Wang, B., & He, C. (2024). *DocLayout-YOLO: Enhancing Document Layout Analysis through Diverse Synthetic Data and Global-to-Local Adaptive Perception*. https://doi.org/10.48550/arXiv.2410.12628

Zhong, X., Tang, J., & Yepes, A. J. (2019). *PubLayNet: largest dataset ever for document layout analysis*. https://doi.org/10.48550/arXiv.1908.07836