AN INTEGRATED COMPUTER VISION AND NATURAL LANGUAGE PROCESSING APPROACH FOR CONVERSATION EXTRACTION FROM MESSAGING APPLICATION SCREENSHOTS

Nguyen Van Anh[1], Pham Van Ha[1,*]

DOI: https://doi.org/10.57001/huih5804.2026.074

ABSTRACT

Messaging platforms such as Zalo are widely used for communication and online services, generating large volumes of conversational data. However, in many real-world scenarios, such data cannot be accessed directly through platform APIs and instead exist only as user interface (UI) screenshots, posing significant challenges for content extraction and processing. This paper proposes an automated conversational processing system for UI screenshots, based on an integrated pipeline combining computer vision and natural language processing. The system consists of several main stages, including text region detection, optical character recognition (OCR), Vietnamese intent classification, and automated response generation. Experimental results show that the proposed system achieves a Character Error Rate (CER) of 0.065, an average intent classification confidence of 0.78, and an average latency of 0.93 seconds per interaction, meeting the requirements of near real-time applications. The findings demonstrate the feasibility of building automated conversational processing systems in scenarios where API access is unavailable and input data are affected by UI-related noise.

***Keywords*:** *Text detection; OCR; PhoBERT; conversational AI; computer vision.*

[1]School of Information and Communications Technology, Hanoi University of Industry, Vietnam

[*]Email: hapv@haui.edu.vn

Received: 14/01/2026

Revised: 18/3/2026

Accepted: 30/3/2026

1. INTRODUCTION

In the context of digital transformation, messaging applications such as Zalo, Facebook Messenger, and Telegram play a crucial role in online communication and generate large volumes of conversational data [1]. However, in many real-world scenarios, such data cannot be accessed via APIs and only exists in the form of user interface (UI) screenshots, posing significant challenges for content extraction and processing.

The problem of text detection and recognition (TDR) in images has been extensively studied, ranging from traditional approaches to deep learning-based methods [2-4], where Convolutional Neural Networks (CNNs) play a central role in feature extraction [5]. Object detection models such as YOLO enable fast and efficient detection across various practical scenarios [6-9], while modern OCR systems such as Tesseract and PP-OCRv3 achieve high performance in text recognition [10-12].

Meanwhile, Transformer-based language models such as BERT and PhoBERT enable effective semantic representation and analysis, particularly for the Vietnamese language [13-15]. Despite these advances, most existing studies focus on individual subproblems or operate on clean text or natural images. The task of processing conversational data directly from Vietnamese messaging application screenshots, particularly under conditions where API access is unavailable, remains underexplored.

Unlike prior studies that primarily address clean text or natural images, the problem addressed in this work involves UI data characterized by small text, high density, and interface noise, which necessitates an end-to-end processing capability. Therefore, this paper proposes an end-to-end pipeline that integrates computer vision and natural language processing techniques to process conversational data from UI screenshots. The system employs YOLOv8n for text region detection, OCR for content extraction, PhoBERT for intent classification, and a rule-based mechanism for response generation. The novelty of this work lies not in proposing a new OCR or NLP model, but in developing and evaluating an end-to-

end conversational processing framework for Vietnamese messaging application screenshots, a scenario that remains underexplored in the literature. The proposed system integrates text detection, OCR, intent classification, dialogue management, and response generation within a unified pipeline and evaluates their combined performance under realistic operating conditions where API access is unavailable.

The main contributions of this study are as follows:

(1) Proposing an end-to-end pipeline for processing conversational data from UI screenshots without relying on APIs;

(2) Constructing a dataset comprising 226 images and 917 text regions that reflect real-world conditions;

(3) Evaluating the system at both module-level and end-to-end levels, including latency and near real-time processing capability.

2. PROPOSED SYSTEM ARCHITECTURE

2.1. Overall system architecture

The system is designed as an end-to-end architecture that integrates computer vision and natural language processing to automatically extract and process conversational content from messaging application UI screenshots. The input to the system is a conversational interface image, while the output is an automated response generated based on the analyzed content.

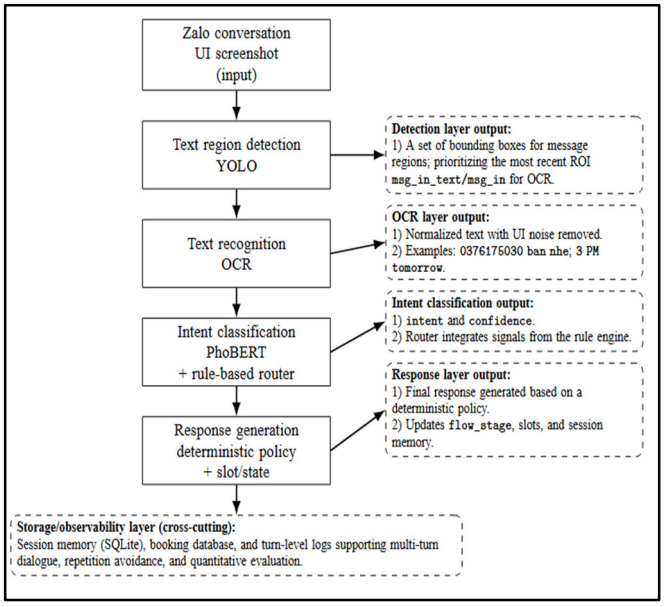

Figure 1. Overall architecture of the conversational processing system from UI screenshots

The processing pipeline consists of four main stages: (1) text region detection, (2) text recognition via OCR, (3) intent classification, and (4) automatic response generation. The overall architecture is illustrated in Figure 1.

The system is modularly designed, enabling independent evaluation of each component and facilitating the analysis of error propagation across the entire pipeline.

2.2. Text region detection

This study employs the YOLOv8n model to detect text regions containing message content within UI screenshots. The problem is formulated as an object detection task, where message regions are treated as target objects. Low-confidence bounding boxes are filtered out using a probability threshold, and the remaining regions are extracted as Regions of Interest (ROIs) for the subsequent text recognition stage (Figure 2). The use of YOLOv8n achieves a balance between accuracy and inference speed, making it suitable for near real-time applications.

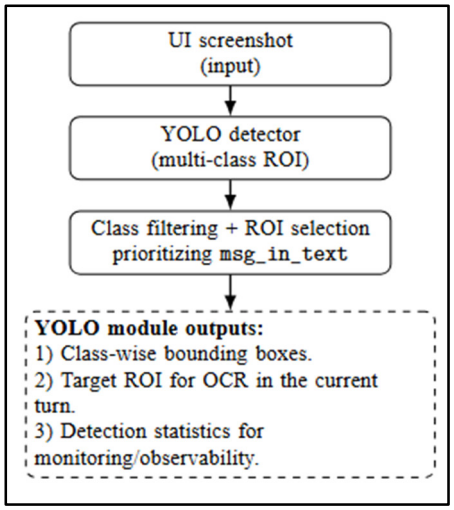

Figure 2. Text region detection process in UI screenshots

2.3. Text recognition and processing

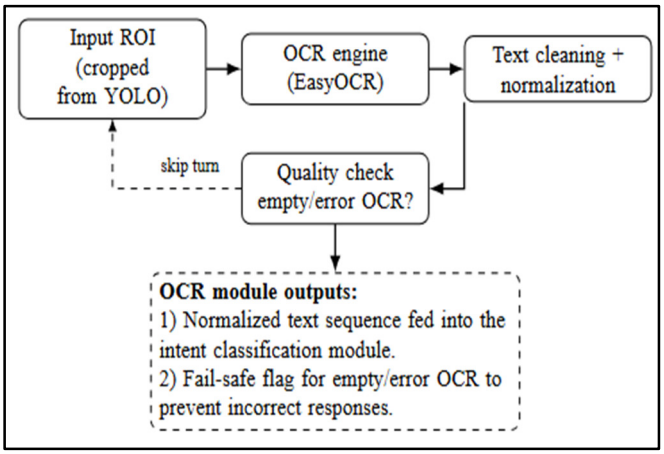

Figure 3. Text recognition pipeline from ROI image regions

The extracted ROIs are fed into an OCR module for text recognition. In this study, Tesseract is utilized for Vietnamese text processing (Figure 3).

Due to the characteristics of UI images, including small text size and high density, OCR outputs may contain errors. Therefore, a post-processing step is applied to improve the quality of the recognized text.

2.4. Intent classification and dialogue management

The model is fine-tuned on common conversational intent classes, producing outputs that include intent labels and prediction confidence scores. The classification results are integrated with the dialogue state through a rule-based coordination mechanism, enabling effective management of multi-turn conversations based on a finite-state machine paradigm (Figure 4).

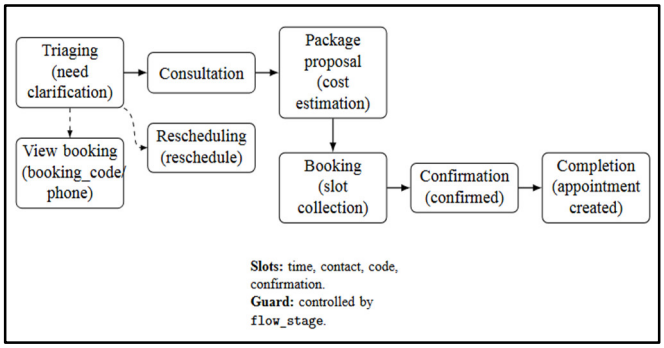

Figure 4. Finite-state machine for multi-turn transactional dialogue management using slot-filling mechanisms

The dialogue state is updated after each interaction, enabling the system to manage multi-turn conversations and slot-filling processes.

2.5. Response generation

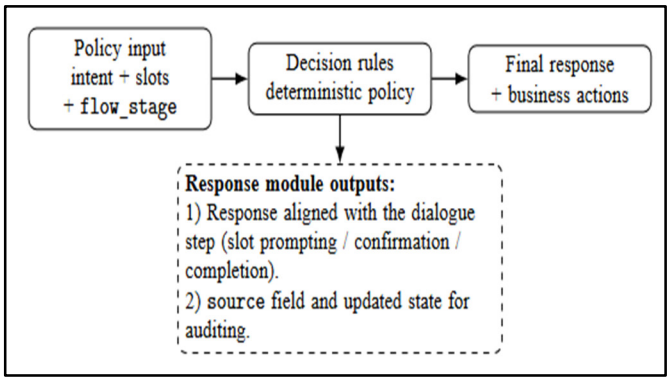

Figure 5. Response generation mechanism based on a deterministic policy

The system generates automated responses based on a deterministic policy to ensure controllability and consistency in conversations. Responses are constructed using the predicted intent, collected information, and the current dialogue state, enabling conversational actions such as requesting additional information or confirming user-provided data (Figure 5).

3. DATASET AND EXPERIMENTAL SETUP

3.1. Data collection

The dataset consists of conversational UI screenshots collected from the Zalo Desktop application. The data contain small text, dense layouts, and interface noise, representing realistic conversational scenarios.

3.2. Text region detection dataset

To train the text region detection model, a dataset consisting of 226 conversational UI screenshots with 917 annotated text regions was constructed. The annotations were provided in the form of bounding boxes. The labeling process was performed manually and cross-validated to ensure annotation accuracy.

Table 1. Specification of the conversational text detection dataset

Attribute	Description
Number of images	226 UI screenshots
Language	Vietnamese
Number of text regions	917
Text size	Small size, high character density
Annotation method	Manual labeling with cross-validation

The dataset was collected from real conversational screenshots captured from the Zalo Desktop application in a Windows desktop environment. The screenshots contain different conversational contents and interface layouts, reflecting practical usage conditions for conversational UI processing.

3.3. Data split

The dataset is divided into two subsets: 180 images for training and 46 images for validation to evaluate model performance.

Table 2. Training and validation data split

Data	Size	Purpose
Train	180	Model training
Validation	46	Model evaluation

3.4. OCR dataset

The text regions extracted from the detection stage are used to construct the OCR dataset, comprising 917 samples with corresponding ground truth annotations. This dataset is used to compute evaluation metrics such as Character Error Rate (CER).

3.5. Intent classification dataset

The intent classification dataset consists of 1,407 Vietnamese message samples for training and 301 samples for validation. The data are labeled into six intent classes:

- BASIC_INFO
- POST_TREATMENT
- PRICING
- SERVICE_DETAIL
- SOCIAL_OR_OTHER
- SYMPTOM_DESCRIPTION.

Table 3. Dataset split for PhoBERT training

Dataset	Number of samples
Train	1407
Validation	301
Number of intent classes	6

To assess model robustness, five test sets were created to represent common noise patterns in Vietnamese conversational data, including abbreviations, missing diacritics, context-dependent inputs, and non-accented text. Each test set contains 302 samples. Performance is evaluated using Accuracy and Macro-F1.

3.6. End-to-end evaluation dataset

The end-to-end evaluation was conducted on 55 real conversational sessions covering appointment booking, schedule inquiry, rescheduling, and cancellation scenarios.

4. EXPERIMENTAL RESULTS AND DISCUSSION

4.1. Text detection model evaluation

Table 4. Text detection performance

Model	Precision	Recall	mAP@0.5	mAP@0.5:0.95
YOLOv5s	0.95	0.78	0.85	0.55
YOLOv7-tiny	0.83	0.84	0.87	0.54
YOLOv8n	0.93	0.86	0.91	0.64

Three models, YOLOv5s, YOLOv7-tiny, and YOLOv8n, were trained and evaluated on the validation set using precision, recall, mAP@0.5, and mAP@0.5:0.95 metrics. The models were trained for 120 epochs with an input image size of 640×640 pixels. All experiments were conducted on a computer equipped with an AMD Ryzen 5 7535HS CPU, NVIDIA RTX 2050 GPU (4 GB VRAM), and 16GB RAM. The results indicate that YOLOv8n achieves the highest overall performance with mAP@0.5 = 0.91 and mAP@0.5:0.95 = 0.64. Although YOLOv5s achieve higher precision (0.95), its recall is lower (0.78). YOLOv7-tiny provides a more balanced trade-off between precision and recall but yields lower overall mAP. Therefore, YOLOv8n was selected for the subsequent stages of the proposed system.

4.2. Inference time of detection models

In addition to accuracy, inference speed is a critical factor for near real-time systems. As shown in Table 5, the inference time of the YOLO models is relatively similar (~28ms).

Table 5. Inference latency of YOLO models

Model	Pre-processing (ms)	Inference (ms)	Post-processing (ms)
YOLOv5s	0.8	28.3	8.3
YOLOv7-tiny	–	28.3	8.8
YOLOv8n	2.7	28.7	8.4

Although YOLOv5s exhibits slightly lower inference time, YOLOv8n benefits from reduced post-processing time, resulting in lower overall latency. Due to hardware constraints, only these three representative models were evaluated. Considering both accuracy and speed, YOLOv8n is selected as the most suitable model for the proposed system.

4.3. OCR module evaluation

Due to the characteristics of UI screenshots with small and dense text, preprocessing plays a crucial role in improving OCR performance. The results show that preprocessing reduces CER by 31.29% for Tesseract and 39.17% for PaddleOCR, while the improvement for EasyOCR is only 2.75%.

Table 6. Impact of preprocessing on CER

Model	CER Raw	CER Clean	Improvement (%)
Tesseract	0.272	0.187	31.29
EasyOCR	0.373	0.363	2.75
PaddleOCR	0.484	0.294	39.17

These findings indicate that input normalization significantly reduces OCR errors, although the level of improvement depends on the OCR method. The results indicate that OCR performance is highly dependent on image quality and text visibility. The improvement achieved after preprocessing demonstrates the importance of input normalization when processing conversational UI screenshots containing small and densely distributed text regions.

4.3.1. Comparison of OCR methods

Table 7. CER comparison across OCR methods

Method	Mean	Median	Std
Tesseract	0.187	0.118	0.412
VLM	0.220	0.000	0.537
PaddleOCR + VLM	0.417	0.233	0.902

OCR methods are compared using CER-based metrics, including Mean, Median, Standard Deviation.

Tesseract achieves the lowest mean CER and standard deviation, indicating the best overall accuracy and stability. This result is likely related to the characteristics of the UI screenshot dataset, which contains horizontally aligned text and relatively simple backgrounds. VLM achieves a median CER of zero, suggesting perfect recognition for many samples; however, semantic paraphrasing, text normalization, and omission of short text segments increase the mean CER and variance. PaddleOCR + VLM exhibits the highest CER and variability, likely due to error propagation between text detection and recognition stages. Despite OCR errors, most outputs preserve sufficient semantic information, allowing PhoBERT to maintain stable intent classification performance.

4.3.2. OCR Latency

In addition to accuracy, OCR latency is also evaluated.

Table 8. OCR latency comparison

Model	Mean (ms)	Median	P95	Samples/sec
Tesseract	305.36	275.07	520.27	3.27
PaddleOCR + VLM	809.10	275.08	1769.48	3.24
VLM	12798.25	10864.92	28269.27	0.078

Tesseract achieves the lowest OCR latency (305ms), followed by PaddleOCR+VLM (809ms), while VLM is substantially slower (12.8s). Therefore, Tesseract is selected for the proposed system.

4.4. Intent classification evaluation

The performance of the intent classification module is evaluated across multiple datasets with different types of linguistic noise (Figures 6 ÷ 8).

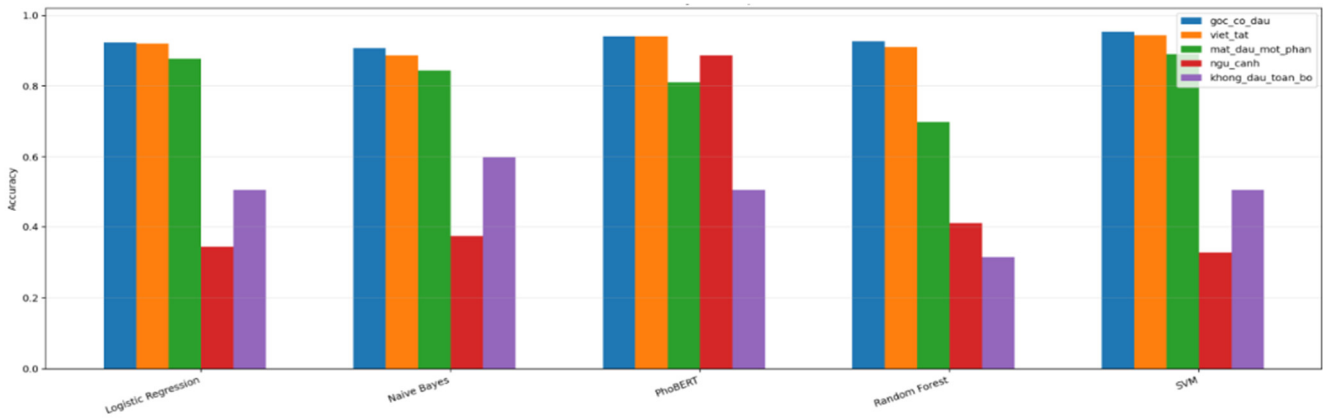

Figure 6. Accuracy comparison across five test datasets

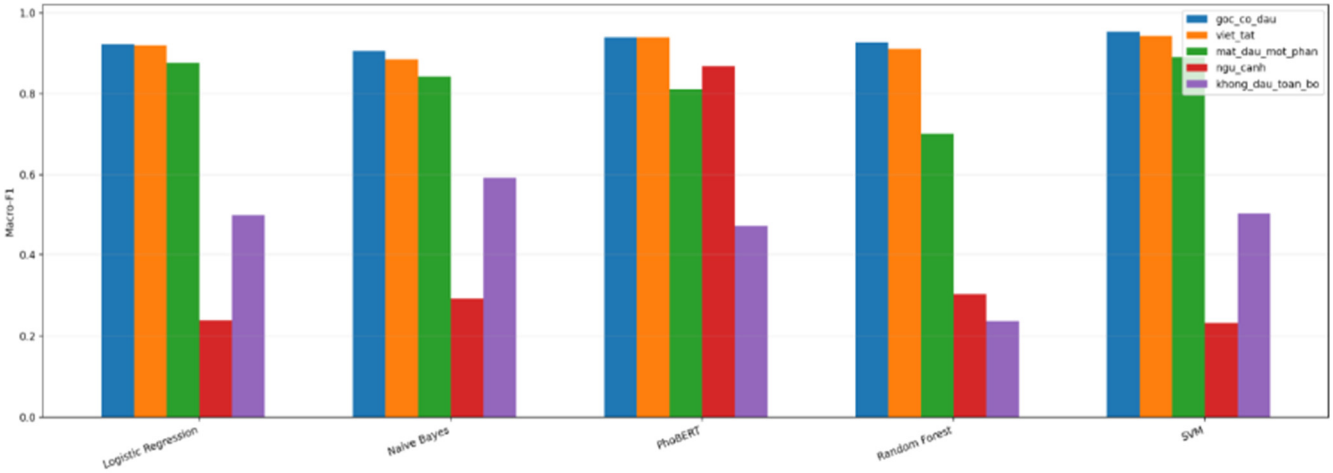

Figure 7. Macro-F1 comparison across models

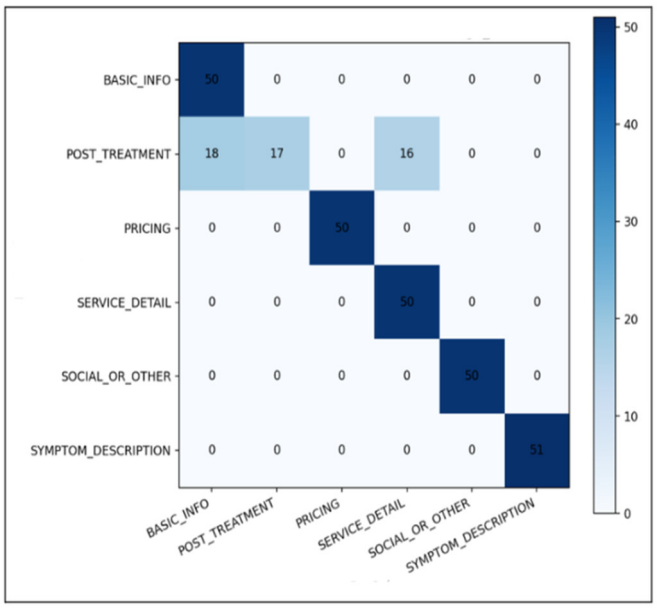

Figure 8. Confusion matrix of the PhoBERT model

The confusion matrix indicates that most intent classes are correctly classified, while misclassifications mainly occur between semantically similar intents. This observation is consistent with the overall evaluation results across noisy conversational datasets.

The results show that PhoBERT achieves a Macro-F1 score of approximately 0.78 across noisy datasets, including abbreviations, missing diacritics, and spelling errors. This demonstrates robust semantic representation and resilience to noise in Vietnamese conversational data. Therefore, the intent classification module plays a critical role in maintaining end-to-end system performance.

4.5. End-to-end system evaluation

The full pipeline is evaluated on real-world conversational scenarios.

Table 9. End-to-end evaluation results

Metric	Value
OCR availability	1.00
CER	0.065
Intent confidence	0.7803
Empty response rate	0
Average response length	108.1 characters
Average latency	0.93s
p50 latency	0.397s
p90 latency	0.480s

The system achieves an end-to-end CER of 0.065, average intent confidence of 0.78, and no empty responses. The average latency is 0.93s per interaction (p50 = 0.397s, p90 = 0.480s), satisfying near real-time requirements. OCR is the most time-consuming component (~300ms), while NLP and dialogue management account for the remaining processing time. The system achieves an 85.5% task completion rate across 55 conversational sessions. Although errors from text detection and OCR may propagate through the pipeline, the semantic robustness of PhoBERT helps maintain stable end-to-end performance under realistic conditions.

4.6. Error analysis

Despite achieving a CER of 0.065 and Macro-F1 of 0.78, several errors still occur during processing. In the detection stage, the model struggles with small, occluded, or UI-noise-affected text regions. In the OCR stage, errors frequently occur with low-resolution text or missing Vietnamese diacritics. Due to the sequential pipeline structure, these errors may propagate and negatively impact intent classification. Analysis shows that approximately 60 - 70% of errors originate from the OCR module. However, the strong semantic representation capability of PhoBERT helps mitigate the impact of noise and maintain overall system performance. Future work may focus on integrating more advanced language models to improve contextual understanding and reduce accumulated errors within the pipeline.

5. CONCLUSION AND FUTURE WORK

This paper presents an end-to-end conversational processing system for messaging application UI screenshots, integrating YOLOv8n for text detection, Tesseract for OCR, and PhoBERT for intent classification. Experimental results achieve mAP@0.5 = 0.91, CER = 0.065, and an average latency of 0.93s, demonstrating the feasibility of near real-time conversational processing without API access. Current limitations include the relatively small dataset, OCR errors in challenging UI conditions, the absence of baseline comparisons for intent classification, and the limited flexibility of rule-based response generation. Future work will focus on larger datasets, comparisons with baseline models such as BiLSTM, Multilingual BERT, and XLM-R, stronger language models, and more advanced conversational generation methods.

REFERENCES

[1]. Ministry of Information and Communications of Vietnam, *Annual Report on the Implementation of Tasks in 2024 and Directions for 2025*. Hanoi, 2024. https://mic.mediacdn.vn/63935241018719846 4/2024/12/28/3-bao-cao-tom-tat-173539798790015505468 03.pdf (accessed January 2026).

[2]. Ye Q., Doermann D.S., "Text Detection and Recognition in Imagery: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(7), 1480-1500, 2015. https://doi.org/10.1109/TPAMI.2014.2366765

[3]. Liu X., Meng G., Pan C., "Scene Text Detection and Recognition with Advances in Deep Learning: A Survey," *International Journal on Document Analysis and Recognition*, 22(2), 143-162, 2019. https://doi.org/10.1007/s10032-019-00320-5

[4]. Penarrubia C., Valero-Mas J.J., Calvo-Zaragoza J., "Self-Supervised Learning for Text Recognition: A Critical Survey," *International Journal of Computer Vision*, 133(9), 6221-6250, 2025. https://doi.org/10.1007/s11263-025-02487-3

[5]. LeCun Y., Bengio Y., Hinton G., "Deep Learning," *Nature,* 521(7553), 436-444, 2015. https://doi.org/10.1038/nature14539

[6]. Redmon J., Divvala S., Girshick R., Farhadi A., "You Only Look Once: Unified, Real-Time Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779-788, 2016. https://doi.org/10.1109/CVPR.2016.91

[7]. Hussain M., "YOLOv5, YOLOv8 and YOLOv10: The Go-To Detectors for Real-Time Vision," *arXiv preprint arXiv:2407.02988*, 2024.

[8]. Wang A., et al., "YOLOv10: Real-Time End-to-End Object Detection," *arXiv preprint arXiv:2405.14458*, 2024. https://doi.org/10.48550/arXiv.2405.14458

[9]. Ramos L.T., Sappa A.D., "A Decade of You Only Look Once (YOLO) for Object Detection," *arXiv preprint arXiv:2504.18586,* 2025. https://doi.org/10.48550/arXiv.2504.18586

[10]. Smith R., "An Overview of the Tesseract OCR Engine," in *Proceedings of the International Conference on Document Analysis and Recognition,* 629-633, 2007. https://doi.org/10.1109/ICDAR.2007.4376991

[11]. Du Y., Li C., Li M., Yang B., "PP-OCRv3: More Attempts for the Improvement of Ultra Lightweight OCR System," *arXiv preprint arXiv:2206.03001,* 2022. https://doi.org/10.48550/arXiv.2206.03001

[12]. Do T., et al., "Reference-Based Post-OCR Processing with LLM for Diacritic Languages," *arXiv preprint arXiv:2410.13305*, 2024. https://doi.org/10.48550/arXiv.2410.13305

[13]. Vaswani A., et al., "Attention Is All You Need," *arXiv preprint arXiv:1706.03762*, 2017. https://doi.org/10.48550/arXiv.1706.03762

[14]. Devlin J., Chang M.W., Lee K., Toutanova K., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2018. https://doi.org/10.48550/arXiv.1810.04805

[15]. Nguyen D.Q., Nguyen A.T., "PhoBERT: Pre-trained Language Models for Vietnamese," *Findings of the Association for Computational Linguistics (EMNLP),* 1037-1048, 2020. https://doi.org/10.18653/v1/2020.findings-emnlp.92