VIETNAM NATIONAL UNIVERSITY OF HOCHIMINH CITY

THE INTERNATIONAL UNIVERSITY

SCHOOL OF COMPUTER SCIENCE & ENGINEERING


# Text Extraction and Detection from Images using Machine Learning Techniques

by

Duong Nguyen Gia Khanh

A thesis submitted to the School of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
Bachelor of Science in Data Science

*Ho Chi Minh City, Vietnam*

2025

# Text Extraction and Detection from Images using Machine Learning Techniques

APPROVED BY:

_____

Nguyen Trung Ky, Ph.D, Advisor

_____

Nguyen Van Sinh, Assoc. Prof

_____

Nguyen Thi Thuy Loan, Assoc. Prof

THESIS COMMITTEE

# ACKNOWLEDGMENTS

First and foremost, I wish to express my heartfelt gratitude to my supervisor, Dr. Nguyen Trung Ky, for his unwavering support and guidance throughout the entire duration of this thesis. His expertise and patience helped me navigate some of the most challenging moments during the research process. Without his encouragement and insightful feedback, this work would not have been possible.

My sincere appreciation also goes to the faculty members and staff of the School of Computer Science and Engineering at International University – VNU HCM for their invaluable contributions to my academic foundation. I owe a special thank you to those dear friends, whose emotional support played a crucial role in helping me maintain mental balance during not only this thesis journey but also throughout my four years of university. At times when I was uncertain or overwhelmed, they reminded me of my purpose and gave me strength not to give up. I am equally grateful to my family for their unconditional love, encouragement, and belief in me during this entire academic journey.

Finally, I would like to dedicate this research to my beloved country, Vietnam. My ambition has always been to contribute meaningfully to the advancement of technology within our community. By optimizing language recognition techniques specifically for the Vietnamese language, I aim to bridge the gap between modern technology and local accessibility. It is my sincere hope that this effort will help Vietnamese citizens, particularly the elderly and those less familiar with English, gain easier access to practical, everyday applications. This thesis marks my first step toward that meaningful goal, and I look forward to many more opportunities to serve my country in the future.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ABSTRACT

Optical Character Recognition (OCR) is essential for digitizing documents but remains challenging for Vietnamese due to complex diacritics, diverse fonts, and varying orientations. This thesis introduces an optimized OCR pipeline tailored specifically for Vietnamese text extraction and recognition. Using the MCOCR dataset which includes 1,155 labeled training images, the proposed pipeline includes text detection applying PaddleOCR), orientation correction according to MobileNetV3, and recognition using VietOCR which implements in Transformer and LSTM-based seq2seq. Due to incomplete annotations, detected text boxes were carefully selected by filtering those with an Intersection over Union (IoU) greater than 0.3, creating a refined dataset suitable for fine-tuning OCR models. To objectively evaluate model performance, a rigorous 3-fold cross-validation was performed, comparing pre-trained and fine-tuned versions of both Transformer and LSTM-based models. Experimental results demonstrate that fine-tuning significantly reduces the Character Error Rate (CER), with the LSTM-based model consistently outperforming the Transformer-based approach across all folds. This research emphasizes the critical role of targeted preprocessing and fine-tuning, ultimately aiming to enhance the practical applicability and accessibility of OCR technology for Vietnamese-language applications.

**Keywords:** Optical Character Recognition (OCR), Vietnamese Text Recognition, PaddleOCR, MobileNetV3, VietOCR, Transformer, LSTM, Fine-tuning, Intersection over Union (IoU), Character Error Rate (CER), K-Fold Cross-validation.

# CHAPTER 1

# INTRODUCTION

## 1.1 Background

Optical Character Recognition (OCR) is a fundamental technology in the modern digitization era, enabling the automatic conversion of scanned, handwritten, or printed text into machine-readable formats. OCR significantly enhances productivity in fields such as automated data entry, document analysis, information extraction, and historical document preservation [1][2]. Over recent decades, the rapid advancement of machine learning and deep learning has substantially improved the performance and reliability of OCR systems, making them more practical and accessible in various real-world applications.

However, despite significant progress, the recognition of Vietnamese text remains notably challenging. The Vietnamese language possesses unique complexities such as multiple diacritical marks, diverse typography, complicated word structures, inconsistent spacing, and varied text orientations. These linguistic characteristics make it difficult for conventional OCR systems, originally optimized for languages like English, to perform accurately when applied directly to Vietnamese text. Recent developments in deep learning-based OCR, such as the Transformer-based VietOCR model and recurrent neural networks (LSTM, seq2seq), have demonstrated promising potential in improving Vietnamese OCR accuracy. Nevertheless, these models still require careful adaptation and targeted fine-tuning on specific Vietnamese datasets to achieve optimal performance [3].

## 1.2 Problem Statement

A critical barrier hindering progress in Vietnamese OCR research is the scarcity and incompleteness of publicly available annotated datasets. Existing datasets, including the recently introduced MCOCR dataset, provide partial annotation aimed primarily at tasks combining OCR and Key Information Extraction (KIE), rather than purely OCR-centric tasks. The incomplete ground-truth annotations within these datasets further complicate the effective training and accurate evaluation of dedicated OCR models tailored for Vietnamese documents [4].

Consequently, there is an urgent need for a robust, comprehensive preprocessing and fine-tuning pipeline specifically designed to address these annotation limitations. Such a pipeline should efficiently manage imperfectly annotated data, handle varying text orientations, and effectively prepare input data for OCR recognition. The development of this tailored pipeline is essential to achieve improved accuracy and reliability in Vietnamese OCR applications.

## 1.3 Research Objectives

This thesis aims to develop, implement, and rigorously evaluate an optimized OCR pipeline specifically tailored for Vietnamese language recognition from images. To fulfill this overarching goal, the research sets forth the following specific objectives:

- Text detection: Employing PaddleOCR for accurately detecting textual regions within images, ensuring robust initial identification of text-containing areas [5].

- Image alignment: Utilizing MobileNetV3 as a classifier to correct orientation issues in detected text regions, thus improving the accuracy of subsequent recognition tasks [6].

- Text recognition: Fine-tuning two distinct VietOCR architectures which are Transformer-based and LSTM-based (seq2seq), to accurately recognize text content from the preprocessed image regions.

In addition, the thesis specifically aims to:

- Conduct comparative experiments to quantify and analyze the improvement between pre-trained and fine-tuned models for both Transformer-based and LSTM-based architectures on carefully selected data.

- Evaluate the OCR pipeline's performance using a rigorous 3-Fold Cross-Validation approach, thus ensuring comprehensive and unbiased validation.

## 1.4 Scope of Work

The research presented in this thesis is centered exclusively on the publicly available training set from the MCOCR competition, which consists of 1,155 labeled images [4]. Due to the absence of ground-truth annotations for the provided test set from the competition, this thesis focuses solely on utilizing the training data for all preprocessing, fine-tuning, and evaluation tasks.

Furthermore, this study specifically targets the OCR task and intentionally excludes the Key Information Extraction (KIE) component. To handle annotation limitations, text boxes detected by PaddleOCR are rigorously filtered using Intersection

over Union (IoU), where only boxes achieving an IoU greater than 0.3 when matched with available ground truths are selected. This carefully refined dataset serves as the foundation for fine-tuning and assessing VietOCR models.

By clearly delineating the research scope, the thesis aims to provide an accurate evaluation framework to demonstrate the effectiveness of preprocessing strategies, fine-tuning methods, and comparative performance analysis. Ultimately, this research contributes directly to developing more accurate and practical OCR technologies tailored specifically for Vietnamese language applications.

## 1.5 Assumption and Solution

In order to construct a truly robust OCR pipeline capable of handling the distinctive challenges posed by Vietnamese text extraction and recognition, I begin by laying out the foundational assumptions that guide each preprocessing and evaluation phase, and then I describe in detail the concrete strategies I employ to fulfill those assumptions.

I acknowledge that the MCOCR training set of 1,155 images inevitably contains a small but non-negligible number of annotation inconsistencies—ranging from swapped labels (e.g., regions meant for “TOTAL_COST” erroneously tagged as “TIMESTAMP,” and vice versa) to mismatches between the number of detected bounding boxes and the expected character counts. Such anomalies, if left unchecked, can introduce significant noise during fine-tuning and degrade model performance. To mitigate this, I devised a deterministic, rule-based cleansing regimen: I first discard any image whose detected box count falls outside plausible bounds for its text length; next, I apply a precise type_map

correction (instead of the inverted mapping), immediately breaking out of the fix loop once the first keyword-driven correction occurs to avoid unintended overwrites; finally, I remove any samples that still exhibit empty or nonsensical label assignments. This process yields a distilled subset of 1,090 reliably annotated images for model fine-tuning, while I deliberately preserve all 1,155 originals in the fold-assignment stage to ensure that my cross-validation splits remain faithfully representative of the dataset's original distribution.

Given the critical importance of accurately locating text regions before recognition, I assume that the PP-OCRv2 model from PaddleOCR—proven to deliver high recall and precision across a variety of scripts, including Vietnamese in multilingual benchmarks [5], is sufficiently robust to serve as my primary text detector. Because fewer than 30 % of actual text boxes in MCOCR are annotated, any attempt to calculate a dataset-wide average Intersection over Union (IoU) would be heavily biased toward zero and thus provide no meaningful insight into detection quality. Consequently, I rely on the off-the-shelf PaddleOCR inference API to generate candidate regions without reporting misleading aggregate IoU metrics here, and I reserve any in-depth detection accuracy studies for future work when a more comprehensively annotated ground-truth becomes available.

Accurate orientation of text crops is equally vital, as even minor skew can confound recognition models. Building on the meticulous approach of Nguyễn Duy Cường, who manually inspected the 1,090 filtered images to isolate the two primary labeling errors and then fine-tuned a MobileNetV3 classifier for 473 epochs to 99 % accuracy on that curated subset [6]. I assume that this rotation model will generalize

effectively to the full 1,155 images, such that residual orientation errors remain well under 1%. To operationalize this, I faithfully re-implement Cường's label-fix rules using the correct type_map, break immediately after each fix to prevent repeated overwrites, and then apply the pretrained MobileNetV3 network to align each crop upright. This strategy obviates the need for any additional manual labeling while ensuring that all detected regions are presented in a consistent, upright orientation for subsequent recognition.

Finally, to obtain an unbiased estimate of my pipeline's generalization performance, I adopt a 3-fold cross-validation scheme, allocating 770 images for training and 385 for validation in each fold, on the premise that this split mirrors established best practices for accuracy estimation even when validation labels remain private [7]. Within each fold, I execute the PaddleOCR → MobileNetV3 sequence, then crop only those regions whose IoU with any available ground-truth annotation exceeds 0.3. This threshold strikes a careful balance between discarding false positives and preserving a sufficient quantity of high-confidence samples. The resulting cropped images form the training and validation inputs for four OCR configurations, pretrained versus fine-tuned, each under both Transformer and LSTM (seq2seq) architectures, and I compare their performance using Character Error Rate (CER), computed via the Levenshtein distance metric [8], to identify the most effective combination of preprocessing and model fine-tuning.

## 1.6 Structure of Thesis

### Chapter 2: Literature Review and Related Work

Chapter 2 surveys the evolution of OCR from rule-based and template-matching systems through convolutional-recurrent architectures and on to modern Transformer-based approaches. I discuss the unique challenges faced when applying OCR to Vietnamese—complex diacritics, ambiguous spacing, font variability—and review prior studies that quantify error rates on Vietnamese text. This chapter also examines leading text-detection models (EAST, CRAFT, PaddleOCR), de-skewing and rotation-correction techniques (Hough-transform, learned classifiers), and recognition frameworks including CNN, Transformer-OCR, and LSTM seq2seq. Finally, I introduce the MCOCR dataset and other Vietnamese corpora, evaluate their annotation quality, and identify gaps—most notably the lack of an end-to-end pipeline that couples robust preprocessing with targeted fine-tuning on Vietnamese data.

### Chapter 3: Methodology

In Chapter 3, I present in detail the design of the end-to-end OCR pipeline. First, I describe my rule-based filtering process that cleans mislabeled and corrupted images. Next, I explain text detection using the PP-OCRv2 (PaddleOCR) model, including any language-specific parameter choices. I then detail the MobileNetV3-based rotation classifier—its reimplementation of label-fix rules and application to align text crops upright. Afterward, I formalize the IoU-based cropping strategy, providing the rationale for the 0.3 threshold and the algorithm for matching detected boxes to sparse ground truth. Finally, I outline the fine-tuning regimen for both VietOCR architectures (Transformer and LSTM seq2seq), specifying hyperparameters, data splits for 3-fold

cross-validation, and the evaluation metrics (primarily Character Error Rate) used to validate model performance.

**Chapter 4: Evaluation and Discussion**

This chapter presents the evaluation results of the OCR models after fine-tuning. The performance of both the Transformer-based and LSTM-based seq2seq models is reported separately for each fold in a 3-fold cross-validation setup, with the best validation result from each fold selected for comparison. Evaluation across folds confirms the superior stability and generalization of the LSTM seq2seq architecture, which consistently outperforms the Transformer under limited annotated Vietnamese OCR data. In addition to CER evaluation, this chapter also examines the effectiveness of the rotation correction module by reporting the Intersection over Union (IoU) between the ground truth boxes and the rotated-cropped predictions. The IoU results are analyzed both globally for the rotation module and fold-wise across the validation sets, providing insight into the quality of spatial alignment and its influence on downstream recognition accuracy. Together, these evaluations underscore the pipeline's robustness and highlight the critical role of both architecture and preprocessing design in Vietnamese OCR tasks.

**Chapter 5: Conclusion and Future Work**

In this final chapter, I recap the key accomplishments of this research: the design and implementation of a robust Vietnamese OCR pipeline that integrates rule-based data filtering, pretrained PaddleOCR detection, MobileNetV3 rotation correction, and IoU-based cropping; the fine-tuning and comparative evaluation of two VietOCR architectures (Transformer and LSTM seq2seq) under a rigorous 3-fold cross-validation framework; and the demonstration that the LSTM seq2seq model, once fine-tuned,

achieves the lowest average Character Error Rate, confirming its suitability for low-annotation scenarios.

Looking ahead, I propose two primary avenues for extending this work. First, I plan to develop a character-level recognition model that processes each glyph independently, rather than relying on word- or line-level decoding; such a design promises finer-grained feature learning, better handling of diacritical marks, and more precise error correction at the subword level. Second, I will explore Reinforcement Learning (RL) techniques to further boost recognition accuracy: by defining a reward function based on sequence-level metrics (e.g., negative CER) and using policy-gradient or actor-critic methods, the OCR system can learn to optimize its decoding strategy dynamically, correcting errors that arise in challenging Vietnamese text layouts. Together, these directions aim to push Vietnamese OCR toward truly adaptive, high-fidelity recognition systems capable of meeting real-world demands.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Multilingual OCR Competitions and Vietnamese Datasets

The MCOCR 2021 competition provided one of the few publicly available benchmarks for multilingual OCR, including Vietnamese document images. Although the training set comprised 1,155 annotated images, the public validation set of 391 images and the private set of 390 images lacked ground-truth labels, posing challenges for model evaluation without external reference data [4]. Prior work on Vietnamese OCR has often relied on small, task-specific datasets, highlighting the need for robust cross-validation strategies when ground truth is unavailable. In addition to MCOCR, several smaller Vietnamese-language corpora have been introduced—examples include the UFPR-Text dataset (augmented with Vietnamese street-sign annotations) and the VnOCR Receipts collection—yet each contains fewer than 1,000 labeled samples, underscoring the overall scarcity of large-scale, publicly accessible Vietnamese OCR resources.

## 2.2 Text Detection and Preprocessing Techniques

Modern text detection methods frequently adopt segmentation-based architectures. PaddleOCR's DBNet backbone, for example, utilizes differentiable binarization to achieve precise text region proposals, enabling high recall on complex layouts [9]. Alternative state-of-the-art detectors include CRAFT ("Character Region Awareness for Text detection"), which models both character regions and affinity maps to accurately localize irregular text shapes [10], and EAST, which offers efficient single-shot text detection.

However, raw detection outputs often contain skewed or rotated regions. MobileNetV3 has been repurposed effectively for rotation estimation and correction; its lightweight inverted residual blocks and squeeze-and-excitation modules allow real-time orientation adjustment with minimal computational overhead [11]. Other approaches employ Spatial Transformer Networks (STNs) as learnable geometric transformers to automatically rectify input crops before recognition [12], though these can add complexity and training overhead.

In the SDSV_AICR team's approach (top 1 overall in the challenge), a rule-based correction stage addressed two main annotation errors: mis-labelled timestamps as TOTAL_COST (and vice versa), and inverted mappings between string labels and category IDs. By replacing the inverse mapping (inv_type_map) with the original type_map and breaking after the first valid fix, their pipeline recovered 1,090 clean samples for training [13].

### 2.3 Recognition Architectures: CNN–LSTM vs. Transformer

Convolutional Recurrent Neural Networks (CRNN), which combine convolutional feature extractors with LSTM-based sequence models, have long been the de facto standard for scene and document text recognition [14]. In these architectures, convolutional backbones such as VGG or ResNet are used to extract spatial features, which are then passed through Bi-directional LSTM layers to model sequential dependencies. When coupled with attention-based decoders, these models—sometimes referred to as AttentionOCR—are particularly effective in recognizing structured, short text sequences such as Vietnamese.

In many Vietnamese OCR benchmarks, CNN–LSTM models have demonstrated superior performance compared to transformer-based alternatives, especially in limited-data regimes. For example, Anh Duc Le et al. showed that an attention-based DenseNet + LSTM model achieved a word error rate of 12.30% on the VNOnDB offline handwriting dataset (~30k training lines), outperforming early transformer variants trained under similar conditions [15]. Similarly, in the ICFHR 2018 VOHTR competition, LSTM + CTC architectures achieved CER scores comparable to or better than transformer models when the training set was constrained to approximately 10k domain-specific samples [16].

Transformer-based sequence-to-sequence models introduce self-attention mechanisms capable of capturing global context without recurrence [17]. While these architectures excel in modeling long-range dependencies and are highly effective on large-scale datasets, they often underperform in low-resource scenarios due to their quadratic complexity and lack of inductive bias for sequential structure. As a result, transformers generally require substantial pretraining or data augmentation to match the efficiency of LSTM-based models in small-scale OCR tasks.

VietOCR's open-source implementation [18] supports both CNN–LSTM and Transformer-based variants. These models are pretrained on multilingual corpora and allow for fine-tuning on specific scripts such as Vietnamese. In practical experiments, the CNN–LSTM model remains a strong baseline, often converging faster and achieving better generalization than the transformer counterpart when training data is limited—further validating the continued relevance of recurrent architectures in the OCR domain [14].

## 2.4 Model Fine-Tuning and Cross-Validation Strategies

I fine-tune pretrained OCR models on domain-specific data, which has been shown to yield significant gains in Character Error Rate (CER) when adapting to novel fonts or languages [19]. In the absence of held-out ground truth for the public validation split, I employ k-fold cross-validation as an effective surrogate: by partitioning the 1,155 images into three folds of 770 training and 385 validation samples each, I closely match the competition's 391-image public validation size and 390-image private test data, thereby approximating its distribution and enabling fair comparison of model variants [20]. I chose three folds—rather than a larger number such as five—to balance the need for sufficiently large training sets (minimizing overfitting) against representative validation splits; each 385-image fold both preserves model generalizability and mirrors the scale of the unlabeled public data.

Within each fold, I first detect text regions and then apply rotation correction; only those crops whose Intersection over Union (IoU) with ground-truth boxes exceeds 0.3 are retained for VietOCR fine-tuning, ensuring high-quality positive samples for recognition training [21]. This pipeline allows me to identify the model variant achieving the lowest CER on validation before final evaluation on the competition's public split.

## 2.5 Evaluation Metrics

A comprehensive evaluation of OCR systems typically employs multiple metrics to capture different dimensions of recognition quality. Two commonly reported measures are full-sequence accuracy and per-character accuracy, defined as follows:

Full-sequence accuracy assigns a score of 1 only when the entire predicted string exactly matches the ground truth, and 0 otherwise, with the final score computed as the mean over all samples. While this metric is intuitively simple, it suffers from an all-or-nothing behavior: a single character error (e.g., a misrecognized diacritic) causes the entire sample to be marked incorrect, obscuring partial successes.

Per-character accuracy evaluates each sample by computing the ratio of correctly predicted characters (matched at the same positions) to the length of the ground-truth string, then averages these ratios across samples. Although more forgiving than full-sequence accuracy, this approach can miscount insertions or deletions—unmatched positions are simply skipped—leading to misleadingly high scores when predictions differ in length or suffer from alignment errors.

To overcome these limitations, Character Error Rate (CER) is widely adopted as the primary metric for recognition performance, especially in languages with complex accent systems such as Vietnamese. CER is calculated as the normalized Levenshtein distance between the predicted and reference strings:

- Proportional Penalty: Each edit operation contributes to the error count, so minor mistakes incur small penalties, while severely corrupted outputs receive appropriately higher scores.
- Sensitivity to All Edit Types: By explicitly counting insertions, deletions, and substitutions, CER accurately penalizes missing or extra characters—errors that per-character accuracy may overlook.

- Length Normalization: Division by the reference length ensures comparability across strings of varying lengths, preventing bias toward either short or long samples.

- Accent and Diacritic Handling: For Vietnamese text, where accents critically affect meaning, CER's fine-grained error accounting ensures that diacritic misrecognitions are neither masked by all-or-nothing scoring nor diluted by alignment mismatches.

Finally, although CER is primarily a recognition metric, I select **IoU > 0.3** as the cutoff for retaining detected crops based on object-detection studies that balance precision and recall when mining high-quality positive samples [22]. This threshold has been shown to effectively filter out poorly localized regions while preserving sufficient data for robust fine-tuning.

Consequently, CER has become the de facto standard for OCR benchmarks (e.g., [8]), providing a robust, interpretable, and language-agnostic measure of recognition fidelity.

### 2.6 Research Gaps and Contribution

Despite progress in general-purpose OCR, few studies have systematically addressed rotation correction and label-mapping errors for Vietnamese document images. Furthermore, the absence of ground-truth labels in the competition's public validation set underscores the need for rigorous validation strategies to avoid overfitting. This work proposes an end-to-end pipeline—combining PaddleOCR detection, MobileNetV3-based rotation correction, rule-based annotation fixes, and VietOCR fine-tuning under a

three-fold cross-validation scheme—to address these gaps and deliver a low-CER recognition model tailored to Vietnamese document images.

## 2.7 Contemporary OCR Pipelines and Practices

### 2.7.1 Modular Pipelines

Modular OCR pipelines represent a conventional yet highly extensible design paradigm wherein individual components—namely, text detection, orientation correction, and text recognition—are decoupled and processed sequentially. This architecture is especially favored in production environments and academic prototypes due to its interpretability, reusability, and ease of customization. Each stage can be independently evaluated, fine-tuned, or replaced, enabling researchers and engineers to swap detection backbones, integrate custom post-processing rules, or adapt recognition modules to new scripts or domains.

One of the most influential toolkits exemplifying this approach is PaddleOCR, an open-source OCR system developed by the PaddlePaddle team at Baidu. PaddleOCR adopts a DBNet-based detector [11], which leverages differentiable binarization to generate high-resolution probability maps for text localization. For recognition, it supports both Convolutional Recurrent Neural Network (CRNN) architectures and Transformer-based decoders, allowing flexibility between low-latency and high-accuracy deployments. PaddleOCR also integrates MobileNetV3 as a lightweight orientation classifier to correct skewed crops before recognition, making it suitable for real-time applications on resource-constrained devices [10].

Similarly, MMOCR—developed under the OpenMMLab project—presents an extensive modular OCR toolkit designed for research and deployment at scale. Unlike PaddleOCR, which tightly couples its internal modules, MMOCR provides a configuration-driven framework that supports over a dozen algorithms for both detection and recognition. Its detection module includes DBNet, CRAFT, and PANet, while the recognition module supports CRNN, SAR, and NRTR among others [12]. MMOCR's design philosophy emphasizes composability: researchers can experiment with various combinations of detection and recognition backbones by modifying configuration files without altering core code.

The modular structure of these pipelines facilitates targeted enhancements. For example, new detection algorithms—such as EfficientDet or YOLOv5—can be integrated to improve localization on complex layouts, while more sophisticated rotation correction techniques (e.g., Spatial Transformer Networks or angle regression heads) can be substituted in place of MobileNet-based classifiers to handle extreme distortions. Likewise, language-specific recognition heads or lexicon-constrained decoding can be plugged in to improve performance on scripts with rich diacritic systems such as Vietnamese.

Moreover, modular pipelines are inherently interpretable. Because each stage produces intermediate outputs (e.g., bounding boxes, rotated crops, predicted sequences), developers can visualize and debug the OCR process at every step. This property is crucial in document understanding tasks where failure in a single stage—such as misaligned detection—can propagate errors downstream. By decoupling the stages,

practitioners can isolate bottlenecks and optimize performance incrementally, an approach less feasible in unified end-to-end systems.

While modularity offers flexibility, it comes at the cost of latency and potential error compounding. For instance, imperfect bounding boxes can mislead recognition networks, and orientation misclassification can flip otherwise legible text. Hence, many real-world systems incorporate heuristic or rule-based post-processing layers to correct common errors (e.g., merging adjacent boxes, filtering improbable sequences, or mapping predictions to expected label types). These practical augmentations demonstrate the need for tight integration between modular components, even when kept architecturally separate.

In summary, modular pipelines such as PaddleOCR and MMOCR remain dominant in real-world OCR deployments due to their extensibility, debuggability, and alignment with traditional engineering workflows. They provide a strong foundation for rapid prototyping and domain adaptation, particularly in resource-constrained or language-specific contexts where model interpretability and modular substitution are essential. Despite recent advances in end-to-end and transformer-based OCR systems, modular pipelines continue to serve as the de facto baseline in many multilingual and document-intensive settings, including Vietnamese OCR research [11][23][24].

### 2.7.2 Unified Transformer-Based OCR

In contrast to modular OCR architectures, unified Transformer-based models adopt an end-to-end paradigm that consolidates the entire OCR pipeline—ranging from image encoding to text generation—into a single neural network framework. These systems typically employ a vision encoder (e.g., ResNet, Swin Transformer, or ViT) coupled with a language decoder (e.g., Transformer or BART-style autoregressive models), thereby eliminating the need for explicit detection, cropping, or orientation correction stages. This architectural simplicity is particularly advantageous in domains where layout structures are irregular, annotations are costly to obtain, or where the OCR output is used directly in downstream tasks such as question answering or document classification.

A notable example of this trend is Donut (Document Understanding Transformer), introduced by Kim et al. [27]. Donut frames OCR as a document parsing problem by treating the entire image as input and directly generating structured textual output in a seq2seq fashion. Unlike traditional OCR systems that depend on region proposals, Donut bypasses all intermediate steps—including text detection and alignment—by learning to attend over visual tokens corresponding to characters and layout elements. Trained on synthetic documents and fine-tuned on datasets such as SROIE and CORD, Donut achieves state-of-the-art performance not only in pure text recognition tasks but also in key information extraction (KIE) and visual question answering (VQA) tasks, showcasing its versatility.

Another influential model is TrOCR (Transformer-based OCR), proposed by Li et al. [26] as part of Microsoft's UniLM framework. TrOCR introduces a

pretraining-finetuning pipeline wherein the model is first pretrained on synthetic OCR data using a vision encoder and a Transformer decoder, followed by fine-tuning on real datasets. The encoder typically leverages Vision Transformer (ViT) [27], while the decoder is a standard Transformer language model. This setup allows TrOCR to effectively model complex character sequences, outperforming CNN–LSTM-based models on benchmarks such as IAM handwriting and printed text datasets.

Unified Transformer-based OCR models offer several compelling advantages. First, their self-attention mechanisms allow for long-range dependency modeling, which is particularly useful for complex or densely packed document layouts. Second, the end-to-end nature reduces engineering overhead by eliminating intermediate artifacts (e.g., bounding boxes, rotation labels). Third, these models are highly flexible in output formats—supporting plain text, HTML-like structured outputs, or JSON-formatted KIE results—making them suitable for multi-purpose document understanding tasks.

However, these benefits come with notable trade-offs. Transformer-based OCR models are data-hungry and typically require large-scale pretraining corpora to generalize well, especially for underrepresented scripts such as Vietnamese. Furthermore, their quadratic complexity with respect to input resolution poses challenges in handling high-resolution scans or documents with fine-grained details. Unlike modular pipelines, they also lack interpretability, making error localization and debugging more difficult. For instance, when a model hallucinates incorrect text or skips entire regions, it is often unclear whether the failure stems from visual misperception or language modeling bias.

In multilingual or low-resource settings, Transformer-based OCR systems often need to be augmented with transfer learning strategies or domain-specific adapters to achieve competitive results. For example, Kim et al. [25] apply synthetic document generation to supplement training, while TrOCR incorporates multilingual pretraining to improve generalization. Despite these efforts, traditional CNN–LSTM architectures still outperform transformers in limited-data regimes, as demonstrated in Vietnamese OCR benchmarks such as VNOnDB and MCOCR [4].

In summary, unified Transformer-based OCR represents a powerful and flexible alternative to modular pipelines, particularly for tasks requiring structured output and context-aware decoding. Models like Donut and TrOCR have demonstrated that the detection-recognition dichotomy can be collapsed into a single architecture without significant performance loss—given sufficient data and computational resources. Their continued evolution suggests a paradigm shift in OCR research, moving toward document-level modeling and multimodal understanding [25][26][27].

### 2.7.3 Generative and Interactive OCR Models

Recent advancements in deep learning have introduced generative models to the OCR domain, where text recognition is framed as an image-to-sequence generation task without requiring explicit detection or alignment. These models are particularly effective at capturing complex dependencies in document images and have demonstrated strong performance in recognizing text from challenging inputs such as handwritten notes, low-resolution scans, and multilingual content.

One notable development in this category is VISTA-OCR, proposed by Hamdi et al. [28], which unifies both content and layout decoding into a single generative model. Given an input image, VISTA-OCR produces token sequences that represent both textual content and spatial information. This approach is particularly suited to documents with semi-structured layouts such as forms or receipts. While layout prediction is a key component of the system, its core recognition ability remains highly relevant to general OCR tasks, especially for handling text with irregular spacing, skew, or orientation.

Similarly, OFA (One For All), introduced by Wang et al. [29], adopts a captioning-like formulation of OCR where the model generates text descriptions directly from image regions. Though not originally designed for OCR alone, the model can be fine-tuned to perform image-to-text recognition tasks effectively, particularly in low-resource or zero-shot scenarios. OFA benefits from large-scale pretraining on vision–language datasets and demonstrates robust generalization across domains and languages.

These generative approaches offer key benefits over conventional recognition pipelines: (i) they eliminate the need for intermediate detection and cropping stages, thus reducing accumulated error, and (ii) they allow for more flexible handling of variable-length text sequences and visual distortions. For instance, instead of depending on fixed region proposals, generative models can attend across the entire image to infer character sequences holistically.

However, generative OCR models also come with limitations. Their decoding process is typically slower due to the autoregressive nature of generation, and they

require large amounts of training data to reach competitive accuracy. Furthermore, they may underperform in applications requiring precise spatial localization or tight control over layout structure.

Although models like VISTA-OCR and OFA begin to bridge the gap between OCR and broader document understanding, it is important to delineate the scope of this thesis. This research focuses specifically on image-based text detection and recognition, using machine learning models to extract text content from document images. Tasks involving structured output generation, semantic labeling, or key information extraction are beyond the scope of this work.

### 2.7.4 Real-World OCR Pipelines

While much of the academic literature on OCR focuses on architectural innovation or benchmark performance, real-world OCR systems must address additional challenges such as noise robustness, latency, scalability, and deployment in production environments. Several industry case studies and engineering reports have outlined end-to-end OCR pipelines that combine classical image processing, deep learning models, and infrastructure components to enable scalable and maintainable document processing systems.

One widely cited industrial implementation is Dropbox's document scanner pipeline [30], which processes millions of user-submitted images for text extraction. The pipeline consists of multiple sequential stages: image capture, denoising, binarization, de-skewing, text detection, text recognition, and finally post-processing. The text detection module uses a convolutional neural network (CNN)-based region proposal

method, while the recognition stage employs a CRNN (CNN–LSTM–CTC) architecture. To minimize user latency, Dropbox emphasizes the use of lightweight models and real-time processing techniques. The pipeline also includes error logging and user feedback loops to improve performance iteratively.

Similarly, Signzy, a fintech company specializing in document automation, reported a production-grade pipeline for identity document OCR [31]. Their system applies EAST or CTPN detectors for region proposals, followed by MobileNet-based recognition modules. Signzy integrates classical image preprocessing techniques such as Gaussian filtering and adaptive thresholding to improve performance on noisy, scanned images. Moreover, the company augments its OCR pipeline with contextual spelling correction and validation against domain-specific dictionaries—particularly important in regulated environments like banking and KYC (Know Your Customer) workflows.

In a more comprehensive technical review, Neptune.ai [32] outlines the practical considerations of deploying deep learning-based OCR systems in production. Their pipeline incorporates YOLO-based detection models, attention-based sequence decoders, and feedback mechanisms for error correction. Importantly, Neptune emphasizes the need for continuous monitoring of OCR model performance using MLOps practices such as logging inference metrics, detecting model drift, and re-training on hard examples. These considerations are rarely addressed in academic research but are essential for maintaining long-term performance in dynamic environments.

Despite differences in model choice and domain-specific constraints, most real-world OCR pipelines share several common characteristics:

- Modularity: Each stage (preprocessing, detection, recognition, post-processing) is isolated for easy debugging and maintenance.
- Hybrid processing: Classical techniques (e.g., binarization, skew correction) are still widely used in conjunction with deep learning.
- Speed–accuracy trade-offs: Lightweight detectors or quantized models are often preferred in mobile or web applications to meet latency requirements.
- Feedback integration: Systems incorporate error logs, human-in-the-loop review, or weak supervision to continually refine recognition quality.

These examples demonstrate that while cutting-edge academic models (e.g., transformers, generative OCR) are essential for advancing the field, real-world deployment still heavily relies on practical engineering heuristics and traditional techniques. For this reason, modular and interpretable pipelines remain the dominant paradigm in production OCR systems—particularly those involving multilingual or noisy document scenarios, such as Vietnamese receipts or scanned forms.

### 2.7.5 Data Augmentation for Low-Resource OCR

In low-resource OCR scenarios, such as Vietnamese document recognition, the scarcity of annotated data presents a significant obstacle to training robust and generalizable models. Data augmentation has emerged as a critical strategy to mitigate this challenge by synthetically increasing dataset diversity, reducing overfitting, and enhancing model invariance to noise, distortion, and layout variation. Augmentation

techniques are typically applied at the image level and can be broadly categorized into geometric, photometric, and generative approaches.

Geometric transformations include random rotations, affine shearing, perspective warping, and scaling. These operations help models generalize to skewed or rotated text regions, which are especially common in scanned documents or mobile captures. For example, applying slight rotations or vertical stretching can simulate natural distortions caused by handheld cameras. Libraries such as Albumentations and ImgAug offer composable pipelines for these transformations and are widely used in OCR preprocessing [33].

Photometric augmentations involve adjustments to image brightness, contrast, blurring, or noise injection. These are particularly useful for modeling poor lighting conditions, faded ink, or motion blur. For instance, Gaussian blur and salt-and-pepper noise can simulate older or low-resolution documents often encountered in administrative and archival digitization. When combined with binarization-based preprocessing, these augmentations can make recognition models more robust to real-world noise.

Beyond simple transformations, generative approaches have gained traction as a more powerful form of augmentation. One such method is ScrabbleGAN, a generative adversarial network designed specifically for text-line image synthesis [34]. ScrabbleGAN learns to generate realistic text images with varying fonts, backgrounds, and distortions, and has been shown to improve recognition accuracy, especially when real labeled data is limited. Although initially developed for English, its architecture is language-agnostic and has been successfully adapted to non-Latin scripts, including

Arabic and Hindi. Adapting such techniques to Vietnamese could significantly expand the training corpus without incurring annotation costs.

A related technique is synthetic text rendering, in which text is rendered on top of background images using various fonts, colors, and alignments. Systems like TextRecognitionDataGenerator (TRDG) or SynthText enable automated generation of labeled OCR images at scale. This is particularly valuable for scripts like Vietnamese, where diacritic variations can be programmatically inserted to ensure balanced training distribution across characters. Additionally, synthetic data can be tailored to specific domains (e.g., receipts, forms) by generating text patterns that mimic realistic field formats.

Despite their effectiveness, augmentation strategies must be used with caution. Over-augmentation—such as applying too much distortion or unnatural colors—can lead to unrealistic training data that hurts generalization. Similarly, models trained heavily on synthetic data may fail to transfer to real-world domains unless fine-tuned appropriately. Therefore, augmentation is most effective when used in conjunction with real data, either for fine-tuning or validation.

In this thesis, while the primary training set remains grounded in real annotated Vietnamese documents, data augmentation techniques (particularly geometric and photometric transformations) are considered essential for increasing model robustness and performance. These augmentations not only compensate for limited sample diversity but also simulate real-world variability inherent in practical OCR deployment contexts.

### 2.7.6 Transfer Learning and Multilingual Pretraining

Transfer learning has become a cornerstone in modern OCR systems, especially for low-resource languages such as Vietnamese, where large-scale annotated corpora are scarce or domain-specific. By pretraining models on high-resource scripts or synthetic data and subsequently fine-tuning them on smaller in-domain datasets, researchers can achieve substantial improvements in recognition accuracy, convergence speed, and generalization.

In OCR, transfer learning typically involves two stages:

1. Pretraining on large, multilingual or synthetic datasets, often using character-level or word-level supervision;
2. Fine-tuning on task-specific or language-specific corpora, such as Vietnamese receipts, scanned forms, or handwriting datasets.

One widely adopted approach is demonstrated in the VietOCR framework, which provides pretrained CNN–LSTM and Transformer-based models trained on mixed-language datasets [18]. These models can then be fine-tuned on Vietnamese data by adapting the vocabulary, character set, and final output layers while retaining pretrained weights in the feature extraction backbone. Empirical evidence shows that such fine-tuned models achieve lower Character Error Rates (CER) and converge significantly faster than training from scratch, especially when the target corpus is small (e.g., under 2,000 images).

In a broader context, Transformer-based OCR models such as TrOCR [26] and Donut [25] also employ large-scale pretraining strategies. For example, TrOCR is

pretrained on synthetic documents using the 90k word lexicon and multilingual corpora before being fine-tuned on language-specific datasets like IAM (English) or Chinese OCR benchmarks. Similarly, Donut applies pretraining on rendered documents with synthetic layouts and multilingual text to enable zero-shot generalization. These strategies have proven effective not only in text recognition but also in downstream tasks such as form understanding and key information extraction.

Multilingual pretraining also addresses a key limitation of many traditional OCR models: their inability to generalize across scripts. By training on multiple writing systems—including Latin, Cyrillic, Chinese, and Vietnamese characters—models acquire shared representations that facilitate transfer across languages. Notably, models like LayoutXLM [35] incorporate both visual and language signals during pretraining, enabling them to adapt more readily to underrepresented scripts through cross-lingual alignment.

The effectiveness of transfer learning is especially evident in Vietnamese OCR benchmarks. Le et al. [4] demonstrated that models pretrained on multilingual corpora and fine-tuned on Vietnamese handwriting datasets (e.g., VNOnDB) outperform baseline models trained solely on Vietnamese data. This is particularly important in domains where collecting labeled samples is time-consuming or costly, such as historical archives or legal documents.

However, transfer learning in OCR is not without limitations. When the source and target domains are highly divergent (e.g., English printed books vs. Vietnamese handwritten forms), negative transfer can occur, degrading performance. Furthermore,

multilingual pretraining often favors high-resource languages unless the training data is explicitly balanced. Careful selection of pretraining corpora and vocabulary design is therefore crucial for maximizing transfer effectiveness.

In this thesis, transfer learning is employed by fine-tuning pretrained VietOCR models on Vietnamese document images. The use of pretrained weights enables faster convergence and better generalization, particularly given the limited size of the labeled dataset. This approach reflects a broader trend in OCR research, where multilingual and cross-domain pretraining has become a de facto strategy for overcoming resource constraints.

### 2.7.7 Evaluation Challenges in Multilingual OCR

While Section 2.5 provided a technical overview of evaluation metrics commonly used in OCR, such as full-sequence accuracy, per-character accuracy, and Character Error Rate (CER), these metrics often face practical limitations when applied to multilingual and low-resource settings. In particular, evaluating OCR performance on languages like Vietnamese introduces additional challenges related to data scarcity, diacritic sensitivity, inconsistent tokenization, and the absence of reliable ground truth annotations.

This section explores the practical difficulties and methodological constraints in applying standard metrics within such contexts. It also highlights the strategies used in recent OCR research such as k-fold cross-validation, weak supervision, and threshold-based filtering to estimate model performance in the absence of labeled test sets. These considerations are critical for designing fair and reproducible experiments, especially when working with multilingual corpora that lack standardized benchmarks.

Evaluating OCR systems in multilingual and low-resource contexts such as Vietnamese poses unique challenges that go beyond conventional accuracy reporting. The complexity of scripts with diacritics, the scarcity of annotated benchmarks, and the variability in input formats (e.g., printed vs. handwritten, receipts vs. forms) often undermine the reliability of standard evaluation metrics or render them insufficient for meaningful comparison.

One fundamental issue is the absence of high-quality ground truth annotations, particularly in public datasets. For example, while the MCOCR 2021 competition [4] provided a public validation set of 391 images and a private test set of 390 images, both lacked ground-truth labels, making direct quantitative evaluation impossible without manual labeling or proxy strategies. In such settings, researchers must resort to alternative techniques such as cross-validation, pseudo-labeling, or manual inspection to estimate recognition performance.

Traditional metrics like full-sequence accuracy and per-character accuracy are often insufficient in multilingual settings. Full-sequence accuracy, which assigns a score of 1 only if the entire predicted string matches the ground truth, tends to be overly punitive for languages like Vietnamese, where a single diacritic misrecognition can invalidate an otherwise correct prediction. Per-character accuracy is more forgiving but fails to penalize character insertion or deletion errors properly, particularly when the prediction and reference lengths diverge.

To address these shortcomings, the Character Error Rate (CER) has emerged as the standard metric in multilingual OCR evaluation [36]. CER is defined as the

normalized Levenshtein distance between the predicted and reference strings, incorporating insertions, deletions, and substitutions. It offers a fine-grained view of recognition quality, especially in scripts where small visual or phonetic changes (e.g., “a” vs. “á”) alter meaning significantly. CER is also length-normalized, making it more robust across samples of varying lengths and document types.

However, even CER has limitations when used in low-resource or multi-script environments:

- It may not reflect semantic correctness, especially in named entities or form fields (e.g., “Việt Nam” vs. “Viet Nam”).
- It can be sensitive to tokenization and normalization procedures, which vary across languages and implementations.
- In the absence of reliable ground truth, CER becomes uncomputable, necessitating proxy validation methods such as IoU filtering of detected regions or using human raters.

Another challenge is the lack of benchmark standardization across languages and domains. While datasets like ICDAR, FUNSD, and IAM are widely used for English and Chinese, Vietnamese OCR benchmarks remain fragmented and small in scale. Some datasets focus on handwriting (e.g., VNOnDB), others on receipts (e.g., VnOCR), and many are not publicly available. As a result, comparing models across studies is difficult, and performance improvements may not generalize beyond the test set used.

In this thesis, model evaluation is conducted using 3-fold cross-validation on the 1,155 labeled samples from the MCOCR training set. Because ground truth is unavailable

for the public validation and private test sets, this strategy serves as a proxy to approximate generalization performance. Within each fold, only detection outputs with an IoU greater than 0.3 relative to ground truth boxes are retained for recognition, ensuring that CER is computed on high-quality crops. This evaluation protocol reflects the practical constraints of Vietnamese OCR and mirrors real-world deployment conditions where perfect annotations are rarely available.

## 2.8 Conclusion

This literature review has surveyed the evolution and current landscape of machine learning-based OCR pipelines, with a focus on both academic innovations and real-world implementations. From traditional modular systems that decouple text detection and recognition (e.g., PaddleOCR, MMOCR) to unified transformer-based models (e.g., TrOCR, Donut), the field has witnessed a shift toward end-to-end architectures that prioritize efficiency, generalizability, and minimal preprocessing. Generative OCR models such as VISTA-OCR and OFA further extend this paradigm by treating recognition as an image-to-sequence generation task, offering enhanced flexibility for document parsing.

However, these state-of-the-art models often require large-scale, high-quality datasets and computational resources, which may not be available in low-resource settings like Vietnamese OCR. In such contexts, techniques like transfer learning, multilingual pretraining, and data augmentation play a pivotal role in bridging the performance gap. The reviewed studies demonstrate that pretraining on multilingual corpora and fine-tuning on domain-specific Vietnamese datasets can significantly

improve recognition accuracy, especially when paired with geometric and photometric augmentations.

Moreover, the review has highlighted practical deployment considerations from industrial pipelines (e.g., Dropbox, Neptune.ai), which emphasize the continued relevance of interpretable, modular designs, especially in environments with limited annotation, real-time constraints, or noisy input data. These insights reinforce the importance of designing OCR systems that balance accuracy with usability, maintainability, and adaptability.

Finally, this review has outlined the evaluation difficulties in multilingual OCR, particularly in the absence of ground truth annotations. It has argued for the use of Character Error Rate (CER) and k-fold cross-validation as practical alternatives to benchmark model performance under limited supervision.

Based on this review, the proposed pipeline in this thesis adopts a modular yet streamlined architecture, integrating state-of-the-art detection and rotation correction modules with a fine-tuned recognition model optimized for Vietnamese document images. This design seeks to balance performance, interpretability, and resource efficiency, in alignment with the challenges and trends identified throughout the literature.

# CHAPTER 3

# METHODOLOGY

## 3.1 Overview of the Proposed Pipeline

### 3.1.1 Data Cleaning and Rule-Based Filtering

**Figure 3.1** below illustrates the pipeline initiates with a dataset of 1,155 Vietnamese document images provided by the MCOCR competition, each image annotated with detailed, field-level ground-truth information. However, a preliminary review of the dataset revealed systematic annotation inaccuracies, particularly involving mislabeling of critical information fields such as "TIMESTAMP" and "TOTAL_COST," as well as incorrect mappings between textual labels and their associated category identifiers. To mitigate these errors, I implement a rigorous rule-based filtering and correction stage. Specifically, by examining annotation sequences, I identify mislabeled instances—swapped labels between timestamps and cost fields—and rectify them by employing the correct label mapping (`type_map`) rather than the inverted mapping (`inv_type_map`). This process systematically halts after the first successful correction per image to prevent unintended secondary adjustments. After applying these correction rules, the dataset is reduced to 1,090 high-quality images, ensuring consistent, reliable ground truth for subsequent pipeline stages.

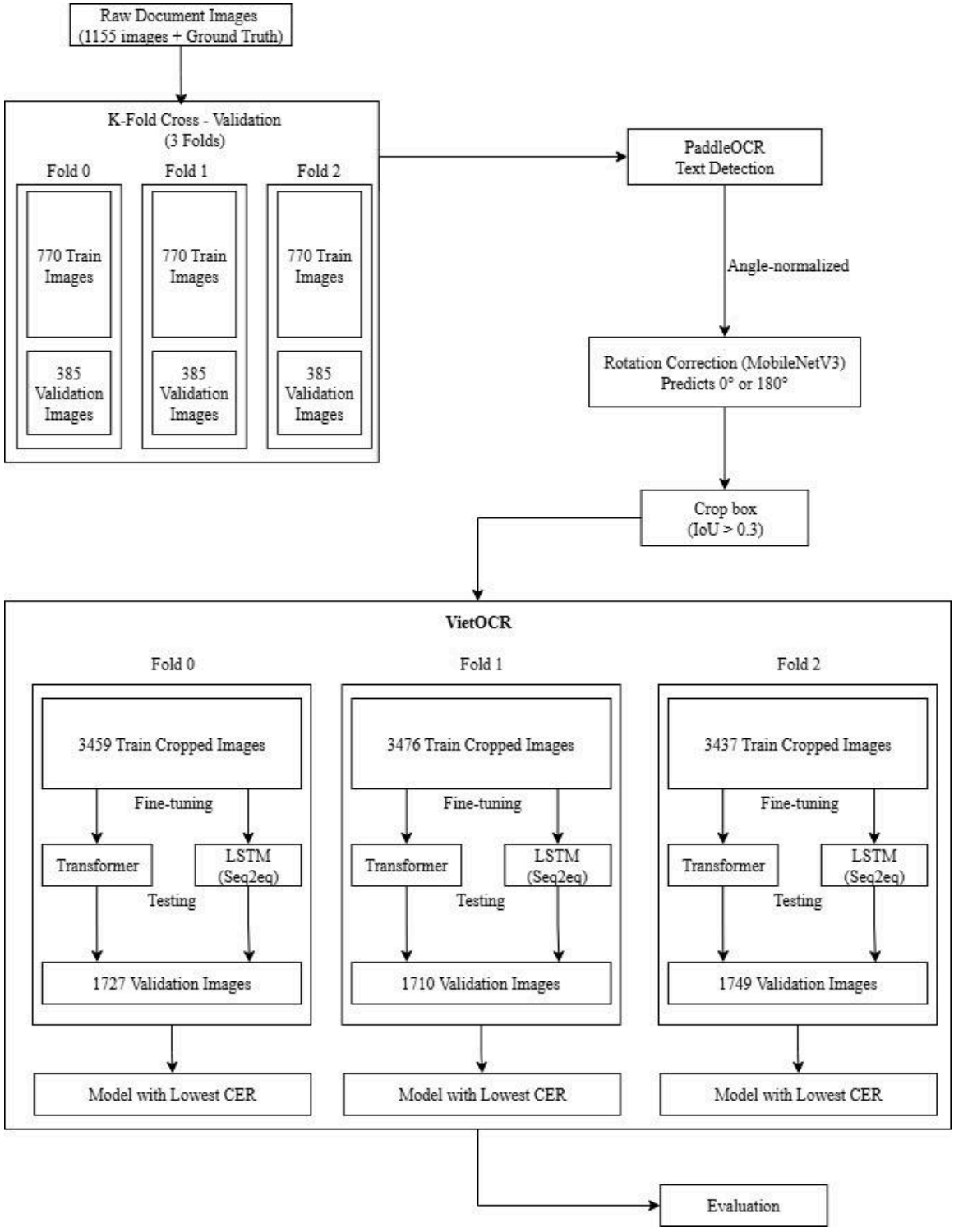



**Figure 3.1** The end-to-end workflow of my OCR system

### 3.1.2 K-Fold Cross-Validation Partitioning

The chart also shows the absence of ground-truth annotations for the competition's public validation dataset (comprising 391 images), evaluating model generalizability necessitates an alternative robust validation approach. To address this, I adopt a three-fold cross-validation strategy. The cleaned set of 1,090 images is partitioned carefully into three distinct folds, each containing exactly 770 training images and 385 validation images. Selecting precisely three folds serves a dual purpose: it ensures each validation subset closely matches the scale and distribution of the original public validation data provided by the competition, while simultaneously maintaining an adequately sized training set to minimize potential overfitting. This cross-validation design facilitates rigorous and fair evaluation of the OCR models' generalization performance across distinct data partitions.

### 3.1.3 Text Detection and Rotation Correction

Within each validation fold, the next essential step involves accurate localization of textual regions, which directly influences the downstream recognition stage. For text detection, I utilize PaddleOCR's DBNet architecture, a state-of-the-art segmentation-based model renowned for its precise and efficient text localization capabilities. DBNet employs differentiable binarization techniques to reliably detect text boxes, even within densely formatted or visually complex Vietnamese documents. Post-detection, detected text regions often exhibit various degrees of skew and rotation, adversely affecting recognition accuracy. To address this geometric distortion, I integrate a rotation correction module based on MobileNetV3 which is a lightweight, efficient neural network architecture specifically adapted and fine-tuned for predicting rotation

angles of text regions. Upon prediction of each detected box's orientation angle, the images are rotated accordingly to upright positions, significantly improving the visual consistency and thus boosting subsequent recognition accuracy.

### 3.1.4 Crop Selection via IoU Filtering

To ensure high-quality data for fine-tuning the OCR recognition models, I perform an Intersection over Union (IoU) evaluation between each rotated detected crop and its corresponding ground-truth bounding box. Specifically, I select crops exceeding an IoU threshold of 0.3—a criterion widely supported in object-detection literature for achieving an optimal balance between precision and recall [22]. Application of this rigorous IoU-based filtering yields a dataset of accurately localized cropped text regions, with detailed counts as follows:

- Fold 0: 3,459 training crops and 1,727 validation crops
- Fold 1: 3,476 training crops and 1,710 validation crops
- Fold 2: 3,437 training crops and 1,749 validation crops

This selective filtering significantly reduces the likelihood of noisy or inaccurately localized text samples negatively influencing OCR model training and evaluation.

### 3.1.5 Recognition Model Fine-Tuning and Evaluation

In the final and most critical stage, I perform fine-tuning of OCR recognition models using the filtered crops. I select two distinct and widely-used architectures provided by VietOCR: (1) a Convolutional Recurrent Neural Network (CRNN) with Long Short-Term Memory (LSTM) layers, and (2) a Transformer-based

sequence-to-sequence (Seq2seq) model. These models have complementary strengths, LSTMs typically excel at short-sequence modeling, whereas Transformers are noted for capturing long-range dependencies. I train each model independently per fold, using mini-batch gradient descent optimization with a batch size of 16. Training incorporates an adaptive learning-rate schedule and early stopping mechanisms guided explicitly by validation Character Error Rate (CER). Each model variant is subsequently evaluated using the fold-specific validation set (approximately 1,700 crops per fold), and the architecture achieving the lowest CER on each validation fold is selected as that fold's representative model.

Following training and per-fold validation, the three best-performing models (one per fold) are evaluated collectively against the competition's official public validation set. This comprehensive final evaluation allows me to conclusively determine the best-performing model architecture and fine-tuning strategy, ensuring robust generalization and optimal OCR performance on Vietnamese document images.

Collectively, this meticulously designed pipeline, encompassing rigorous data cleaning, robust k-fold cross-validation, precise text detection and rotation correction, selective IoU-based crop filtering, and extensive model fine-tuning, establishes a comprehensive framework capable of delivering consistently low CER scores. This thorough, end-to-end methodology directly addresses existing gaps in Vietnamese OCR, offering substantial improvements in accuracy, robustness, and practical applicability.

## 3.2 Data Preparation

### 3.2.1 Dataset Description

The dataset utilized in this study originates from the MCOCR 2021 competition, which specifically focuses on optical character recognition (OCR) challenges for multilingual document images, including Vietnamese texts. The training dataset provided by the competition comprises 1,155 annotated images, each representing scanned receipts and invoices from various business establishments within Vietnam. These images exhibit considerable diversity in layout complexity, text density, background texture, and font styles. Such heterogeneity poses significant challenges for OCR models, as text may appear in skewed, rotated, or distorted formats, and annotation accuracy might vary substantially across samples. Each training image was originally annotated at the field level, specifying critical information categories such as timestamps, total cost values, merchant names, and addresses, thereby providing valuable structured data for supervised training.

In contrast, the public validation dataset provided by the MCOCR 2021 competition consists of 391 images without ground-truth annotations. Besides, the private dataset contains 390 images and also lacks ground truth. This dataset serves as the basis for participants' final evaluation; however, due to the absence of publicly available ground-truth labels, it is impossible to directly measure or monitor model performance using standard evaluation metrics. The public validation and private test images share similar characteristics and visual challenges as the training images, reflecting real-world scenarios and complexities encountered in practical Vietnamese OCR applications. Nonetheless, the lack of annotations in the validation dataset necessitates alternative

evaluation strategies, such as internal cross-validation, to robustly estimate model performance before official submission and public leaderboard evaluation.

### 3.2.2 K-Fold Partitioning

To robustly evaluate model performance and generalization, particularly given the absence of labels for the official public validation set, I employ a rigorous three-fold cross-validation strategy. I partition the cleaned dataset of 1,090 high-quality annotated images, obtained through the aforementioned rule-based data cleaning step, into three distinct folds. Each fold comprises 770 images designated explicitly for model training and 385 images reserved strictly for validation purposes.

The choice of a three-fold structure is carefully justified based on two primary considerations. Firstly, the validation portion of each fold (385 images) closely mirrors the size of the official unlabelled public validation set (391 images) and private test set (390 images), ensuring the internal validation process accurately simulates real evaluation conditions and data distribution encountered during official competition submissions. Secondly, utilizing three folds, rather than a higher fold count such as five or ten, preserves a larger volume of data within each training subset. This strategic partitioning maximizes the quantity of training samples per fold, thereby enhancing the robustness and generalization capabilities of the trained models while maintaining representative validation subsets. Such a configuration represents an optimal compromise between adequate model training and reliable model evaluation.

For creating the three-fold partitioning, I adopt a systematic and reproducible approach. Specifically, I utilize stratified random sampling techniques implemented via Python scripting (leveraging popular data-processing libraries such as Pandas, NumPy, and scikit-learn [30]). Stratification ensures balanced distribution of key characteristics such as vendor types, text density, or annotation categories across each fold, minimizing distributional discrepancies that could bias model evaluations. The exact partitioning script includes steps for randomized shuffling of the dataset (using fixed random seeds for reproducibility), followed by stratified division into the training and validation subsets per fold. This careful methodology ensures consistency across experimental runs, allowing meaningful comparative analyses between different model architectures and configurations within subsequent pipeline stages.

Through this rigorously designed three-fold partitioning process, I effectively address the critical challenge posed by the absence of ground-truth annotations in the public validation dataset, while simultaneously maximizing model generalization and minimizing risks of overfitting. Consequently, this meticulous data preparation strategy forms the foundation for reliable, reproducible, and accurate OCR model evaluation throughout the pipeline.

## 3.3 Text Detection

### 3.3.1 Choice of Detector

For the critical task of detecting textual regions within Vietnamese document images, I selected the **PaddleOCR's DBNet** text detection model. PaddleOCR, developed by Baidu Research, is a comprehensive OCR toolkit based on the PaddlePaddle deep-learning framework. It consists of several pre-trained models specifically optimized for text-related tasks such as text detection, recognition, and text direction classification. Among these models, the PPOCRv3 variant of PaddleOCR has emerged as one of the most lightweight yet highly accurate OCR solutions, demonstrating robust performance across various text recognition tasks.

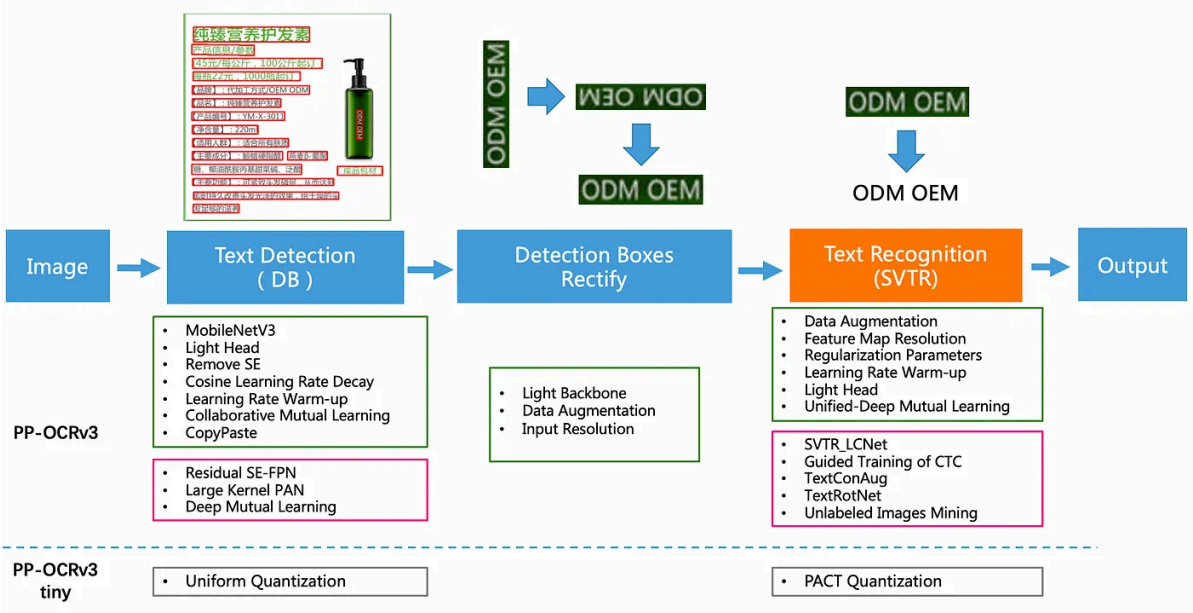



**Figure 3.2** PPOCRv3 architecture diagram

The architecture of PaddleOCR v3 (PPOCRv3), illustrated in **Figure 3.2**, integrates three main functional containers: Text Detection, Detection Boxes Rectification, and Text Recognition. These modules are interconnected through a pipeline

known as a "blackboard architecture," where each stage applies multiple specialized algorithms to optimize OCR accuracy.

In this study, however, I specifically employed only the Text Detection component of PaddleOCR v3. The PPOCRv3 Text Detection module is based on the Differentiable Binarization (DB) algorithm, combined with a MobileNetV3 backbone, and further optimized through a distillation strategy. The DB algorithm represents a significant advancement in segmentation-based text detection methods, notably due to its differentiable thresholding mechanism that enables end-to-end training and eliminates the complexities associated with manual threshold selection.

Unlike traditional methods (e.g., EAST or CRAFT), DBNet does not require heuristic binarization after generating a probability map. Instead, it directly predicts a differentiable threshold map alongside the text probability map. Specifically, DBNet's differentiable binarization is mathematically defined by the sigmoid function:

$$B_i = \frac{1}{1 + e^{-k(P_i - T_i)}}$$

where:

- $B_i$ is the binarized output for the pixel at position i.
- $P_i$ is the probability of the pixel i being part of a text region (shrink map).
- $T_i$ is the learned threshold at pixel i (threshold map).
- k is a hyperparameter controlling the steepness of the binarization (usually set to 50 in practice).

This differentiable approach allows DBNet to optimize both text region detection and binarization simultaneously, significantly enhancing detection accuracy, especially in scenarios involving rotated, curved, or densely packed text common in Vietnamese document images.

In choosing PaddleOCR's DBNet, I also evaluated other state-of-the-art detection models like CRAFT (Character Region Awareness for Text detection) and EAST (Efficient and Accurate Scene Text detection). CRAFT utilizes character-level segmentation and affinity maps to precisely identify individual characters, making it especially suitable for irregularly shaped or distorted text. On the other hand, EAST provides efficient, real-time detection capabilities through single-stage regression methods, predicting bounding boxes directly from the convolutional feature maps. However, I selected DBNet primarily because it balances computational efficiency and detection accuracy, excels in detecting arbitrarily shaped texts, and seamlessly integrates within the PaddlePaddle framework, facilitating straightforward deployment.

### 3.3.2 Inference Procedure

For practical application of the DBNet detector within my OCR pipeline, I carefully implemented an inference procedure that comprises multiple preprocessing and parameter-tuning steps, meticulously following PaddleOCR's recommended practices.

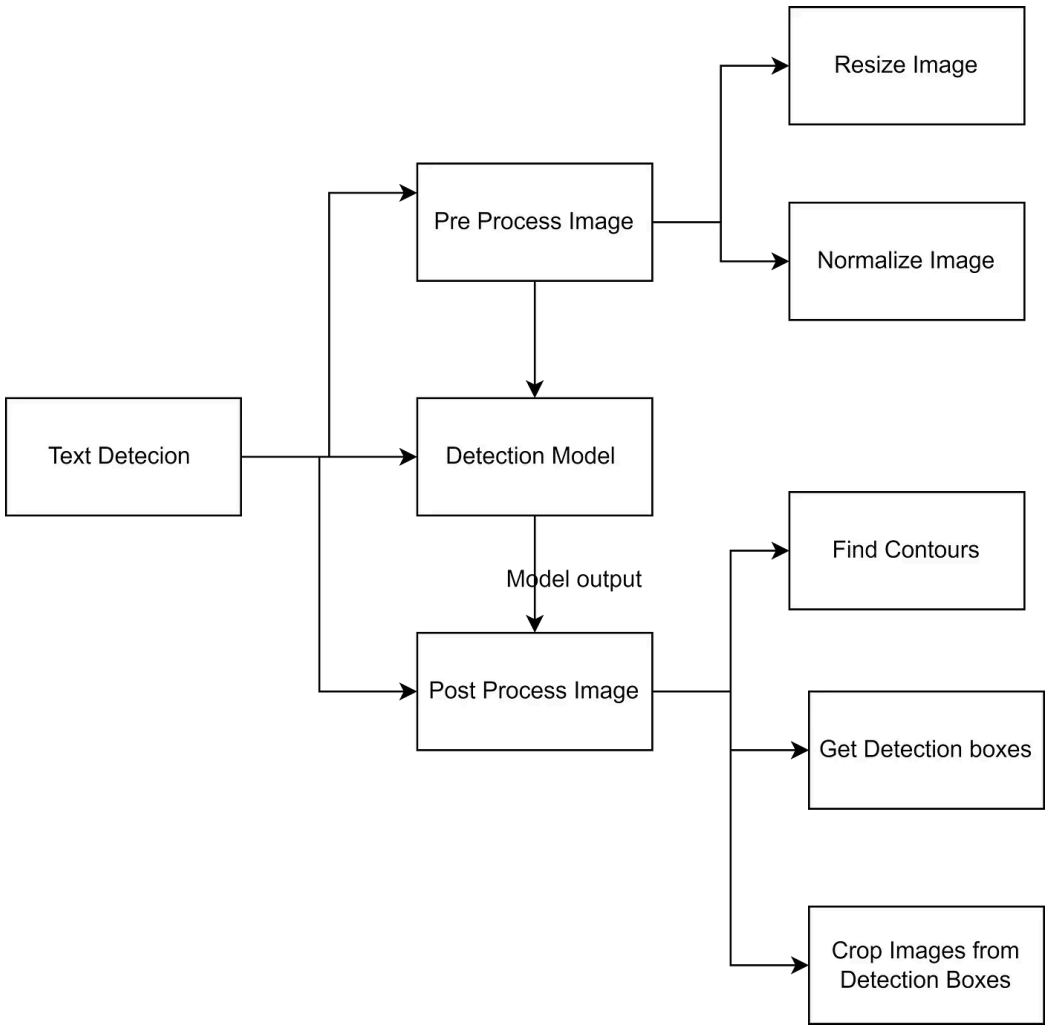



**Figure 3.3** Text Detection stages diagram

**Figure 3.3** provides a detailed overview of the PaddleOCR pipeline, explicitly highlighting the stages of the DBNet text detection process utilized in this project. The Text Detection component in PaddleOCR includes the following sequential processing stages:

- **Preprocessing:**

  Input images undergo resizing, normalization, and format conversion to ensure compatibility with the detection model. Specifically, each image is resized, maintaining its aspect ratio, with the longest side fixed to a maximum length of 960 pixels. The resized dimensions are further adjusted to be multiples of 32, aligning with deep neural network efficiency. After resizing, normalization using ImageNet-derived mean values ([0.485, 0.456, 0.406]) and standard deviations ([0.229, 0.224, 0.225]) is performed. Additionally, the image data is converted from the HWC (Height, Width, Channel) format to CHW (Channel, Height, Width) format as required by PaddleOCR models.

- **Detection Model (Inference):**

  The preprocessed image is then fed into the DBNet model. Leveraging a MobileNetV3 backbone, the model extracts deep convolutional features, subsequently passing them through the DB head, which outputs two primary maps: a shrink map (probability map of text regions) and a threshold map (learned binarization thresholds).

- **Post-processing:**

  Following inference, a predicted segmentation map is generated by applying the differentiable binarization formula. Pixels with probabilities higher than the threshold (`db_thresh`)—set at 0.3—are considered part of the text regions. This low threshold helps maintain a high recall, ensuring the model captures even faint or partially obscured text. Subsequently, bounding boxes are extracted from this segmentation map. Contour detection (via OpenCV's `findContours` function) locates continuous regions, each contour then undergoes processing to generate a minimal enclosing rectangle (mini-boxes). Each mini-box is scored based on its mean predicted probability (score), and boxes with scores lower than `db_box_thresh` (0.3) are discarded to eliminate low-confidence predictions. The chosen parameters (`db_thresh=0.3` and `db_box_thresh=0.3`) represent a careful balance between recall (capturing more true text regions) and precision (excluding false positives). These parameter choices were empirically tuned based on experimental evaluations specific to my Vietnamese OCR dataset.

Following preprocessing, DBNet inference generates a probability map for text region predictions. This continuous probability map is then transformed into a binary segmentation map via the DB module, guided by two critical hyperparameters: `db_thresh` and `db_box_thresh`, both set at 0.3 for my project. Here, `db_thresh` represents the binarization threshold applied to the predicted probability maps to separate foreground text pixels from background pixels. A lower threshold (such as 0.3, as selected here) increases sensitivity, thus improving recall by capturing a larger portion of

potential text regions, at the risk of possibly introducing more false positives. Meanwhile, `db_box_thresh` specifies the minimum average probability that a predicted text bounding box must exhibit to be considered valid. A value of 0.3 is again chosen as a compromise, retaining most valid text proposals without excessively increasing false positives. These thresholds were empirically tuned to ensure a high recall of Vietnamese text regions, which frequently appear with varying font styles, sizes, and orientations, while still minimizing irrelevant or noisy detections.

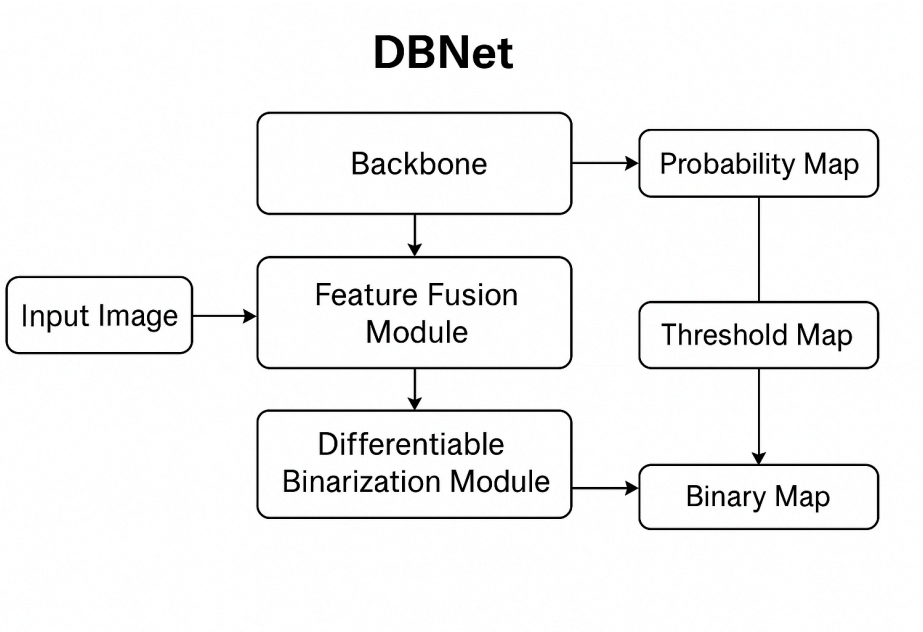



**Figure 3.4** DBNet Model Architecture

To illustrate clearly, **Figure 3.4** shows the detailed architecture of DBNet. DBNet includes three primary components: a Backbone Network (e.g., MobileNetV3), responsible for extracting features from the input image; a Feature Fusion Module, typically implemented as a Feature Pyramid Network (FPN), which integrates multi-scale features produced by the backbone; and the critical Differentiable Binarization Module, which generates a probability map and learns a threshold map to produce a differentiable

binary map. The probability map indicates likely text regions, while the threshold map dynamically determines the optimal binarization threshold for each pixel. These maps are combined using the differentiable binarization formula to yield a final binary segmentation map, from which text bounding boxes are subsequently extracted.

Overall, by carefully adopting DBNet from PaddleOCR and precisely tuning the inference parameters (`db_thresh` and `db_box_thresh` at 0.3), I established a robust foundation for accurate Vietnamese text detection. This step significantly impacts downstream tasks such as rotation correction and text recognition, directly enhancing the OCR system's final performance.

## 3.4 Rotation Correction

### 3.4.1 Model Architecture

In my pipeline, the rotation correction step is a crucial component aimed at rectifying orientation inconsistencies in detected text regions, a significant challenge commonly observed in Vietnamese OCR datasets. To efficiently tackle this problem, I adopt a lightweight, mobile-friendly neural architecture, specifically the MobileNetV3-Small network, which has been adapted explicitly for orientation classification. MobileNetV3-Small is widely recognized for its effectiveness and computational efficiency, making it ideal for real-time deployment scenarios such as OCR tasks on mobile or resource-constrained devices [11].

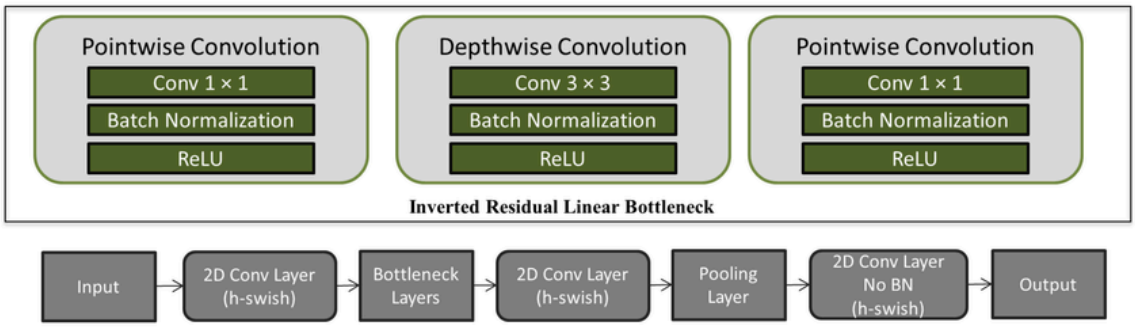



**Figure 3.5** General Structure of MobileNetV3-Small

**Figure 3.5** presents the general schematic of the MobileNetV3-Small model employed in this study. The model begins with an initial 2D convolutional layer employing the hard-swish (h-swish) activation function, followed by multiple inverted residual bottleneck blocks, each designed to efficiently extract complex features from the input. The network subsequently transitions through additional convolutional and pooling layers before outputting angle-classification logits (two categories corresponding to rotations of 0° and 180°). This architecture is optimized explicitly for computational efficiency, enabling effective rotation classification even in resource-constrained environments.

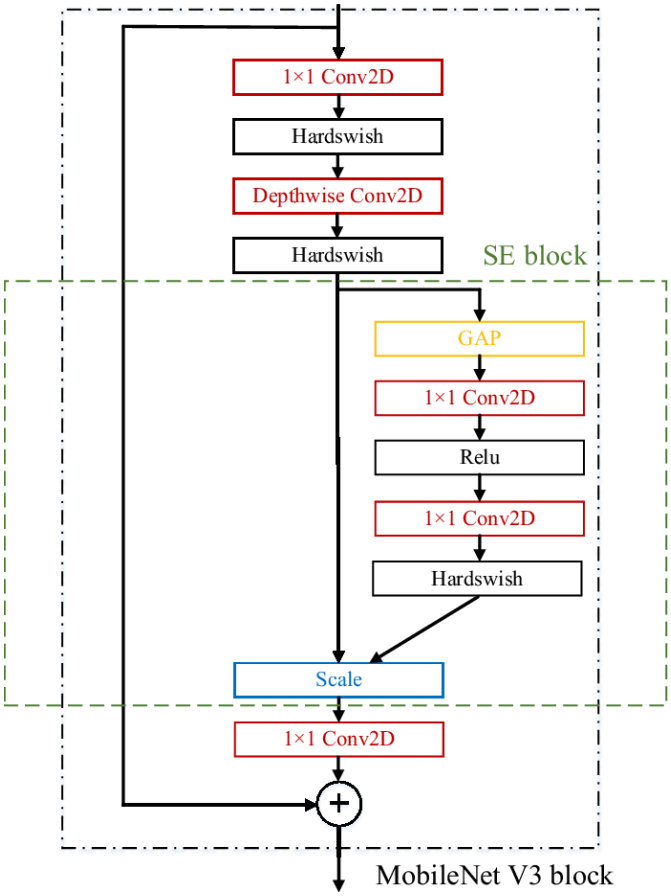



**Figure 3.6** Detailed Structure of the MobileNetV3 Inverted Residual Bottleneck Block

**Figure 3.6** provides an expanded view of the inverted residual linear bottleneck, a key building block of MobileNetV3. This figure explicitly illustrates the internal components of the MobileNetV3 inverted residual bottleneck block, including pointwise convolution, depthwise convolution, batch normalization, and ReLU activations Each bottleneck block comprises three primary layers:

- Expansion layer: A pointwise convolution (1×1 convolution) increases the number of input channels, combined with batch normalization and ReLU activation.
- Depthwise convolution: Applies a 3×3 depthwise convolution, processing each channel separately to significantly reduce computational load, followed again by batch normalization and ReLU activation.
- Projection layer: Another pointwise convolution layer reduces the feature dimension back to the desired number of output channels, accompanied by batch normalization and an activation function.

This structure effectively compresses features while maintaining high representational capacity.

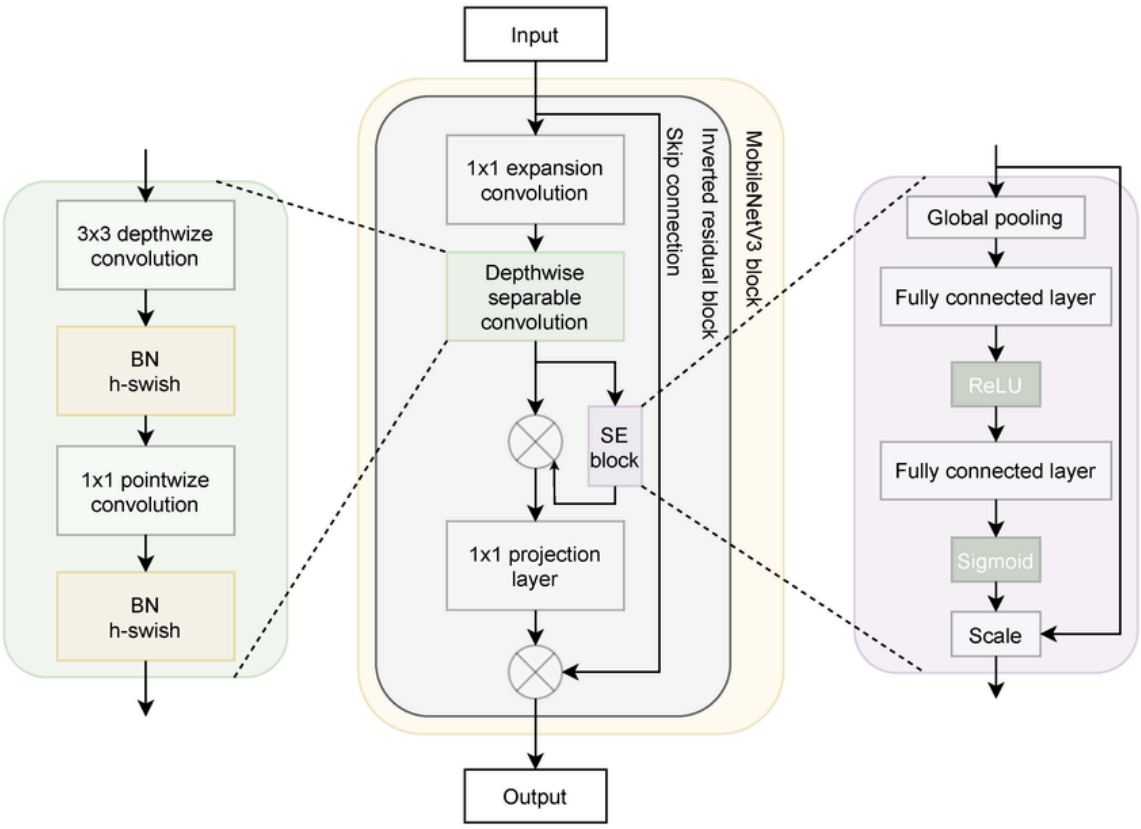



**Figure 3.7** MobileNetV3 Block with Integrated Squeeze-and-Excitation (SE) Module and Hard-swish Activation

**Figure 3.7** presents the comprehensive architecture of a MobileNetV3 block, explicitly detailing the integration of the Squeeze-and-Excitation (SE) module and the hard-swish activation function. The block initiates with a 1×1 convolution (expansion), followed by a depthwise convolution that leverages h-swish activation. Subsequently, the SE module performs channel-wise attention using global average pooling (GAP), two fully connected layers (with ReLU and sigmoid activations, respectively), and channel scaling. Finally, a 1×1 convolution reduces channel dimensions, and a skip (residual) connection merges the input features directly with the output if spatial and channel dimensions are consistent. This intricate design ensures enhanced representation

capabilities by dynamically recalibrating channel-wise features while preserving computational efficiency.

Depthwise separable convolution is a key building block of MobileNet architectures, greatly reducing computational cost compared to standard convolution:

$$\text{DepthwiseConv}_{(k)}(X)_{i,j} = \sum_{m,n} W^{(k)}_{m,n} \cdot X^{(k)}_{(i+m,j+n)}$$

$W^{(k)}$ denotes the kernel applied to the kkk-th channel of the input $X$.

Pointwise Convolution uses a convolution with kernel size 1×1 to combine the outputs of the depthwise convolution across channels:

$$\text{PointwiseConv}(X)_{i,j,c} = \sum_{k} W_{c,k} \cdot X_{i,j,k}$$

This step changes the number of channels and mixes information across channels.

The inverted residual block (Bottleneck Block) used in MobileNetV3 consists of four key stages:

- Expansion (increase dimension)
- Depthwise convolution
- Squeeze-and-Excitation module
- Projection (reduce dimension)
- Optional skip (residual) connection

Formula Representation:

Let $X \in \mathbb{R}^{H \times W \times C}$:

- Expansion Layer (1×1 Convolution):

$$X_{exp} = \text{Conv}_{1\times1}(X), \quad X_{exp} \in \mathbb{R}^{H \times W \times (C \times t)}$$

  - $t$ is the expansion factor.

- Depthwise Convolution (3×3 or 5×5):

$$X_{dw} = \text{DepthwiseConv}(X_{exp}), \quad X_{dw} \in \mathbb{R}^{H' \times W' \times (C \times t)}$$

- Squeeze-and-Excitation Module:

$$X_{se} = \text{SE}(X_{dw}), \quad X_{se} \in \mathbb{R}^{H' \times W' \times (C \times t)}$$

- Projection Layer (1×1 Convolution):

$$X_{proj} = \text{Conv}_{1\times1}(X_{se}), \quad X_{proj} \in \mathbb{R}^{H' \times W' \times C'}$$

- Residual Connection:

  - If $H$ x $W$ x $C$ = $H'$ x $W'$ x $C'$:

$$Y = X + X_{proj}$$

  - Otherwise, no residual is applied:

$$Y = X_{proj}$$

The Squeeze-and-Excitation (SE) block adaptively recalibrates channel-wise features using global contextual information:

- Squeeze: Global Average Pooling (GAP)

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_{i,j,c}$$

- Excitation: Channel-wise scaling factors learned via two fully connected layers:

$$s = \sigma(W_2 \cdot \mathrm{ReLU}(W_1 z))$$

  $W_1$, $W_2$: Fully connected weights

  σ: Sigmoid activation function to map values between [0, 1].

- Scale: Re-scale original features with learned importance weights:

$$X_{se} = X \odot s$$

Here, ⊙ represents element-wise multiplication.

The Hard-Swish (h-swish) Activation Function is computationally efficient, providing improved accuracy and efficiency over standard ReLU:

$$\text{h-swish}(x) = x \frac{\mathrm{ReLU6}(x+3)}{6}$$

Where ReLU6 is defined as:

$$\mathrm{ReLU6}(x) = \min(\max(x, 0), 6)$$

This function provides smoother transitions, allowing better gradient flow during training, especially beneficial for mobile and edge deployments.

MobileNetV3 extensively uses batch normalization, stabilizing training by normalizing feature maps at each layer:

$$\hat{x} = \frac{x - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$$

- $x$: Input feature map
- $\mu$: Mean of batch
- $\sigma$: Variance of batch
- $\epsilon$: Small constant for numerical stability

After normalization, affine transformation is applied:

$$y = \gamma \hat{x} + \beta$$

- $\gamma$, $\beta$: Learnable parameters for scaling and shifting.

Global Average Pooling (GAP) reduces spatial dimension by averaging feature maps across height and width dimensions:

$$GAP(X)_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_{i,j,c}$$

This yields a channel-wise descriptor, used primarily in SE modules and before fully connected layers.

Cross-Entropy Loss (Used for Angle Classification) for training the angle classifier (final FC layer with logits), the cross-entropy loss function is applied:

$$L(y, \hat{y}) = -\sum_{c=1}^{4} y_c \log(\hat{y}_c)$$

- $y_c$: One-hot ground truth vector (for angles 0°, 180°)
- $\hat{y}_c$: Softmax predictions from the model logits

#### 3.4.1.1 Bottleneck Block

MobileNetV3-Small utilizes a specialized bottleneck block, an advanced form of the inverted residual structure first introduced in MobileNetV2. This design leverages depthwise separable convolutions combined with channel-wise attention mechanisms to reduce computational overhead significantly while preserving or even enhancing representational capacity. The bottleneck block comprises several distinct operations:

- Expansion Layer (1×1 convolution): Initially, the number of channels is expanded to create a higher-dimensional feature space, enabling richer feature extraction.
- Depthwise Convolution (3×3 or 5×5 convolution): Applied separately to each expanded channel, drastically reducing the computational complexity compared to standard convolution.
- Squeeze-and-Excitation (SE) Module: This attention mechanism identifies and emphasizes channels containing the most relevant information, effectively improving model accuracy by recalibrating channel-wise features.

- Projection Layer (1×1 convolution): Reduces the number of channels back to a lower dimension, enhancing computational efficiency.
- Activation Function: Employs either ReLU or the more efficient hard-swish (h-swish) activation, improving nonlinear modeling capabilities, particularly beneficial for mobile deployment.

The combination of these strategies results in a highly efficient and powerful block capable of capturing nuanced features required for precise angle classification tasks [37].

#### 3.4.1.2 MobileNetV3-Small Architecture

The detailed architecture of the MobileNetV3-Small model adopted in this study, along with layer-specific details, is outlined comprehensively in **Table 3.1** below:

| Input Size | Operator | Expansion | Output Channels | SE | Activation | Stride |
|---|---|---|---|---|---|---|
| 224×224×3 | Conv2D 3×3 | - | 16 | ✗ | h-swish | 2 |
| 112×112×16 | Bottleneck 3×3 | 16 | 16 | ✓ | ReLU | 2 |
| 56×56×16 | Bottleneck 3×3 | 72 | 24 | ✗ | ReLU | 2 |
| 28×28×24 | Bottleneck 5×5 | 88 | 24 | ✗ | ReLU | 1 |
| 28×28×24 | Bottleneck 5×5 | 96 | 40 | ✓ | h-swish | 2 |
| 14×14×40 | Bottleneck 5×5 (×3) | 240 | 40 | ✓ | h-swish | 1 |
| 14×14×40 | Conv2D 1×1 | - | 576 | ✗ | h-swish | 1 |
| 14×14×576 | Global Average Pooling | - | - | ✗ | - | - |
| 1×1×576 | Fully Connected Layer | - | 1024 | ✗ | h-swish | - |
| 1×1×1024 | Fully Connected Layer | - | 4 (angles: 0–3) | ✗ | - | - |

**Table 3.1** Detailed Layer Specification for MobileNetV3-Small Model Used in Rotation Correction Task

In this project, a modification was introduced at the final fully connected layer to suit the rotation classification task, outputting logits for four distinct rotation angles: 0° and 180°corresponding directly to upright and upside down, respectively. This modification differs significantly from the original classification head, which generally targets generic ImageNet classes. Thus, this customized output layer represents an essential adaptation for the specific OCR rotation-correction application at hand.

### 3.4.2 Ground-Truth and Rule-Based Fixes

Accurate training of the MobileNetV3 rotation correction model requires precise annotations regarding image orientations. However, the original annotations provided in the dataset exhibited several systematic errors and inconsistencies, including mislabeling and inverted mappings. Common annotation errors observed were:

- **Swapped labels**: Fields labeled "TIMESTAMP" were mistakenly annotated as "TOTAL_COST," and vice versa.
- **Missing or duplicated fields**: Some images lacked essential annotations, while others had duplicate entries for identical information.
- **Incorrect mappings**: The dictionary-to-list conversions during preprocessing occasionally produced inverted mappings (e.g., using `inv_type_map` incorrectly).

To resolve these annotation issues, I implemented a rule-based correction approach that leverages a combination of keyword-based matching and heuristics validation:

- Utilized the `type_map` dictionary explicitly defining valid annotation keys instead of the previously erroneous `inv_type_map`, ensuring correct forward mapping of labels.
- Keyword matching strategies (e.g., "ngày," "tổng cộng") were applied to automatically correct swapped labels based on semantic content.
- Heuristic validations were also employed: timestamps were verified through formatting checks, and total cost values were confirmed as numerical values through digit validation.

These steps significantly enhanced the dataset's consistency and reliability, thus ensuring robust and accurate model training outcomes.

### 3.4.3 Application in Pipeline

The trained and validated MobileNetV3-Small orientation classifier was integrated effectively into the overall OCR pipeline with clearly defined steps:

Initially, text regions detected by the PaddleOCR's DBNet model were cropped from the original document images, producing individual text patches as input to the rotation classifier.

Each cropped image was then passed to the MobileNetV3-Small classifier. The model generated predictions across two angle categories: upright (0°) and upside down (180°). The predicted orientation class was determined by the highest logit score.

If a crop was predicted to have an orientation other than upright (0°), it was automatically rotated back to the correct vertical alignment. This rectified image was subsequently forwarded to the text recognition stage, ensuring enhanced accuracy by standardizing input orientations prior to recognition.

Comparison to the Original MobileNetV3 Model. The significant differences introduced in my adaptation of MobileNetV3 compared to its original implementation primarily relate to:

- Originally, MobileNetV3 was designed for generic image classification (1000-class ImageNet), whereas my adapted version outputs four logits specifically tuned to predict discrete text orientations (0°, 180°).
- The network parameters were fine-tuned explicitly on Vietnamese OCR datasets containing oriented textual regions, significantly diverging from the original generic visual domain.
- Implemented extensive rule-based preprocessing strategies not present in the original MobileNetV3 training, greatly enhancing dataset quality and model robustness.

These targeted adjustments result in a specialized MobileNetV3 variant, optimized precisely for orientation correction within the OCR workflow, achieving superior performance specifically tailored to the challenges inherent to Vietnamese document processing.

## 3.5 Recognition Model Fine-Tuning

### 3.5.1 Convolutional Neural Network (CNN)

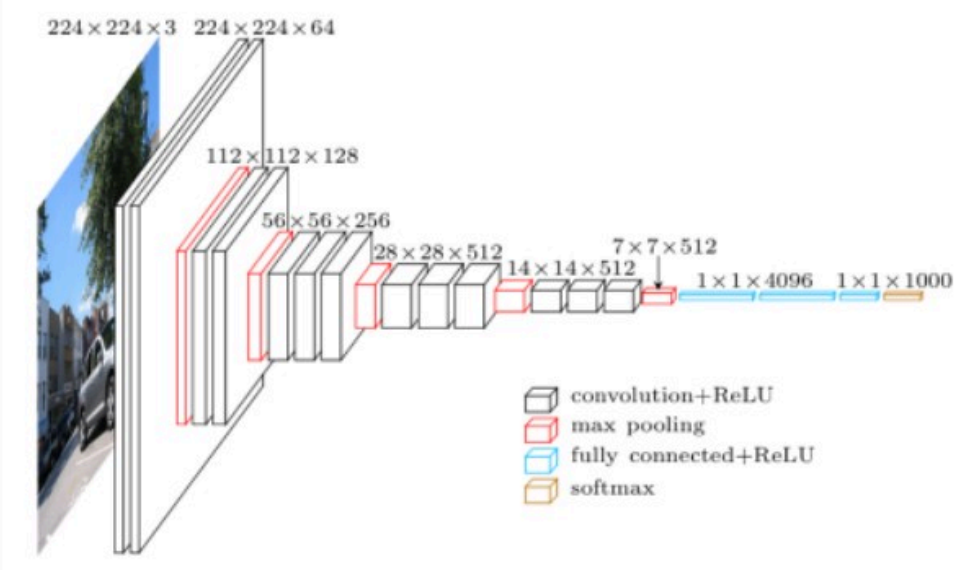



**Figure 3.8** VGG19 architecture

In this thesis, the convolutional backbone serves as a visual feature extractor that encodes the input image into a compact feature map, which is then passed to a sequence modeling component. For OCR tasks, especially with text lines where the image width is much larger than the height, the CNN must preserve sufficient resolution in the horizontal axis while reducing the vertical dimension. The VGG19 batch norm shown in **Figure 3.8**.

This is achieved by adjusting the stride of the pooling layers—particularly using asymmetric pooling (stride = (2,1)) in later CNN stages. This allows the model to preserve the time-step resolution required by sequence models such as LSTM or Transformers. The convolutional pipeline consists of stacked convolution layers followed by batch normalization, non-linear activation functions (typically ReLU or h-swish), and downsampling operations (e.g., MaxPooling or AveragePooling).

After passing through the CNN, the feature map has the shape (C,H′,W′), where:

- C is the number of channels (features),
- H′ is the reduced height,
- W′ is the temporal axis corresponding to character positions.

The final feature map is reshaped into a sequence of vectors $\{x_1,x_2,...,x_T\}$, where $T = W' \times H'$ and each $x_t \in R^C$. This sequence is then passed into the next stage of the model.

### 3.5.2 CNN + LSTM (AttentionOCR)

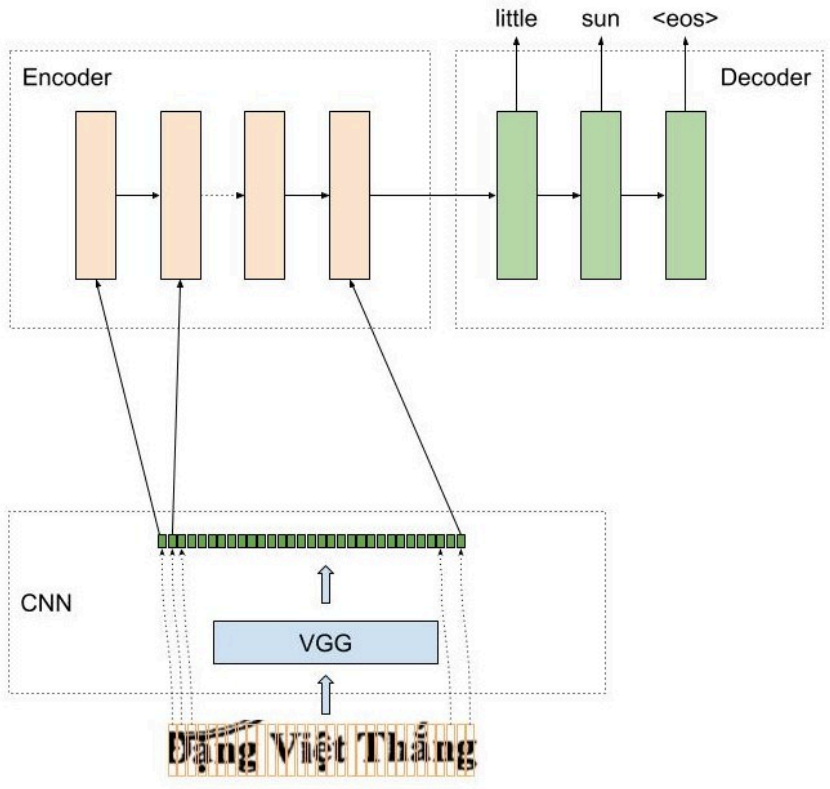



**Figure 3.9** Architecture of the CNN-LSTM Encoder-Decoder Model with Attention

**Figure 3.9** illustrates the architecture of the CNN + LSTM model applied in this thesis. It begins with a VGG-based convolutional backbone that extracts spatial feature maps from the input image. These maps are then flattened across the spatial dimensions into a sequential representation that captures the horizontal layout of text characters. The resulting sequence is passed to a bidirectional LSTM encoder, which encodes contextual dependencies in both forward and backward directions. The decoder, implemented as a unidirectional LSTM, generates each output token based on an attention-weighted context vector and the previous decoder state.

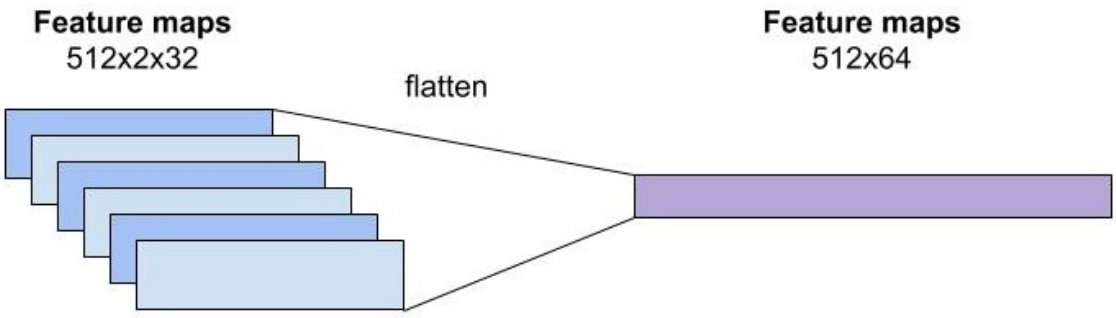



**Figure 3.10** Transformation of CNN Feature Maps into Sequential Input for LSTM

**Figure 3.10** shows the flattening operation more clearly. The CNN generates a stack of feature maps with shape (channels × height × width), which is reshaped into a 2D sequence (sequence length × feature dimension) where the height and width axes are merged into a temporal dimension. This reshaping is essential to ensure that the LSTM processes the image left-to-right as a sequence of features.

The AttentionOCR model used in this thesis builds upon the encoder-decoder architecture with attention. It comprises a CNN-based encoder and an LSTM-based decoder augmented with an attention mechanism.

After extracting features via CNN, the flattened sequence $\{x_1, x_2, ..., x_T\}$ becomes the input for the attention-based decoder. At each time step t, the decoder generates the next character based on the following operations:

Context Vector Computation:

$$c_t = \sum_{i=1}^{T} \alpha_{t,i} \cdot x_i$$

where:

- $x_i$ is the i-th encoder output.
- $\alpha_t$ is the attention weight over position i at decoder time t, indicating how much the decoder focuses on each part of the input.

Attention Weights:

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{j=1}^{T} \exp(e_{t,j})}$$

with:

$$e_{t,i} = \text{score}(s_{t-1}, x_i)$$

where:

- $s_{t-1}$ is the decoder's hidden state at the previous time step,

Decoder Update:

$$s_t = \text{LSTM}(y_{t-1}, s_{t-1}, c_t)$$

where:

- $y_{t-1}$ is the previously generated character (embedded as vector),
- $s_{t-1}$ is the prior decoder hidden state,
- $c_t$ is the context vector at time t.

Output Prediction:

$$y_t = \text{softmax}(W_o \cdot s_t)$$

where:

- $W_o$ is the output projection matrix mapping hidden states to vocabulary logits,
- $y_t$ is the predicted probability distribution over characters at time step t.

This architecture allows the decoder to attend to different regions of the input feature sequence dynamically, making it robust to variable-length text and misalignment.

### 3.5.3 CNN + Transformer (TransformerOCR)

In the Transformer-based architecture, the visual features extracted from the CNN are passed to a Transformer encoder-decoder structure. Unlike RNNs, Transformers process the entire sequence in parallel, which significantly accelerates training and allows for better global context modeling.

The TransformerOCR model employed in this thesis integrates a convolutional backbone (typically VGG or ResNet) with a Transformer-based encoder-decoder architecture. This architecture allows for parallelized sequence modeling and has shown impressive results in NLP and OCR domains.

Unlike LSTM-based models that process the input sequentially, the Transformer model computes attention over the entire input sequence simultaneously, allowing it to capture global context at each decoding step.

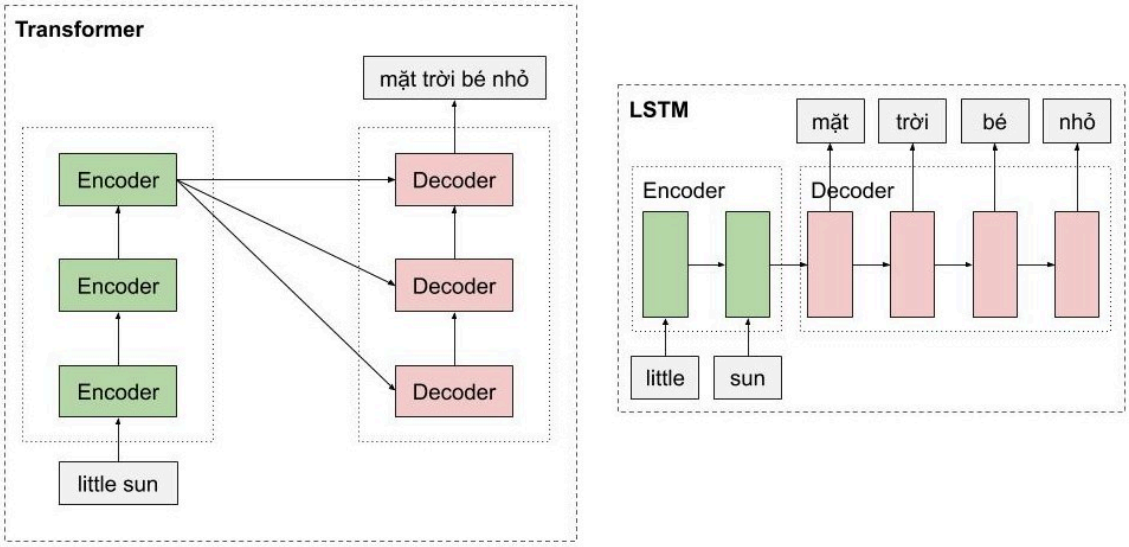



**Figure 3.11** Comparison of Transformer and LSTM Architectures

This **Figure 3.11** illustrates the key difference between the LSTM and Transformer mechanisms for decoding. The LSTM (right side) processes one token at a time in sequence, maintaining internal hidden states, while the Transformer (left side) attends to the entire input sequence simultaneously at each decoding layer through multi-head attention. This distinction is especially critical in OCR tasks, where recognizing dependencies between distant visual tokens improves performance.

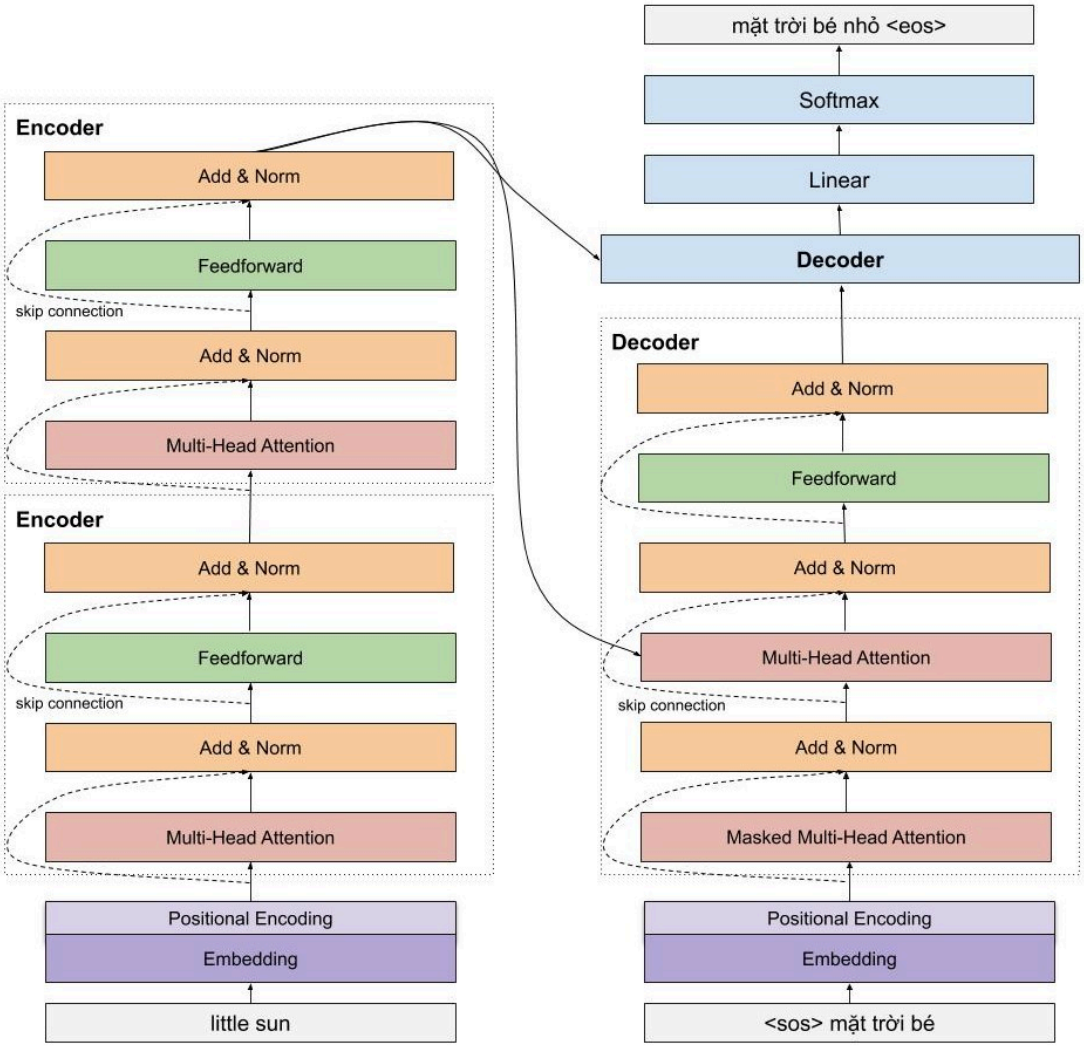



**Figure 3.12** Transformer Encoder-Decoder Architecture with Attention Layers

The Transformer consists of the following key components, shown in **Figure 3.12**:

- **Encoder Stack**: Each encoder block includes:
    - Multi-head self-attention
    - Add & Norm
    - Feed-forward network
    - Add & Norm again (residual connection)

- **Decoder Stack**: Each decoder layer comprises:
    - Masked multi-head self-attention (to preserve autoregressive generation)
    - Cross-attention with encoder output
    - Feed-forward layer
    - Normalization and skip connections throughout

These structures are repeated multiple times (usually 6–12 layers). The Transformer uses position encoding to maintain information about the order of tokens, which is crucial in OCR as spatial layout matters.

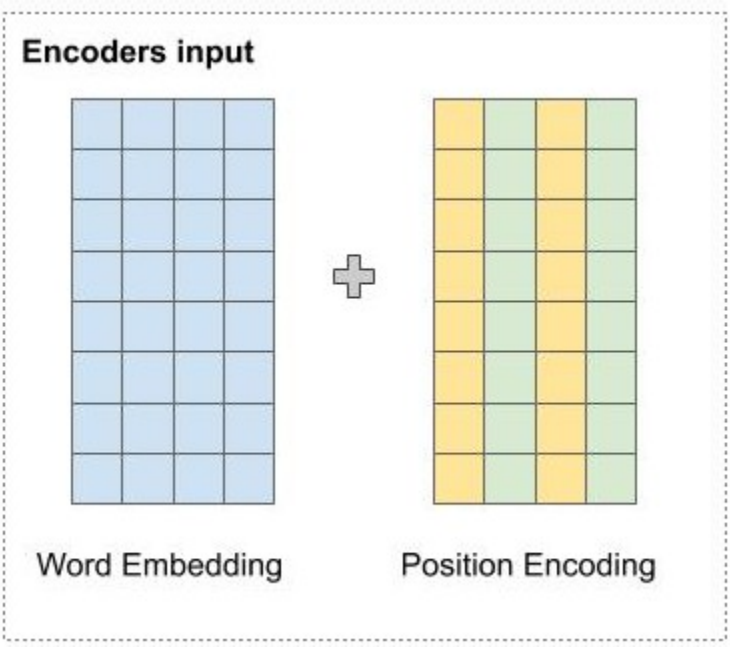



**Figure 3.13** Positional Encoding and Word Embedding Composition for Encoder Input

The Transformer lacks recurrence and thus requires positional encodings to retain the order of input tokens in **Figure 3.13**. The embedding vector passed into each encoder block is the sum of:

- Word embeddings (visual token features from the CNN)
- Positional encodings computed as:

$$\mathrm{PE}_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right)$$

$$\mathrm{PE}_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right)$$

Where:

- *pos*: position index in the sequence
- *i*: embedding dimension index
- $d_{\text{model}}$: the size of the model's hidden dimension

The Transformer's core capability lies in its attention mechanism. The scaled dot-product attention is defined as:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Where:

- *Q*, *K*, *V*: query, key, and value matrices
- $d_k$: the dimension of the keys

For richer representation, multi-head attention is used:

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, ..., \mathrm{head}_h)W^O$$

Each head is:

$$\mathrm{head}_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

This mechanism allows the model to focus on different parts of the sequence in parallel, improving both accuracy and generalization.

Although Transformers theoretically offer advantages in context modeling, real-world OCR tasks—especially those involving short Vietnamese phrases—do not always benefit significantly from this complexity. As observed in this thesis, the LSTM-based Seq2Seq model can perform comparably (and even better on small datasets) with faster convergence and less sensitivity to hyperparameters.

## 3.6 Cross-Validation

In order to robustly evaluate the performance of recognition models under limited data availability and to compensate for the absence of ground-truth labels in the official validation set of the MCOCR 2021 dataset, a 3-fold cross-validation strategy was employed on the available 1,155 annotated training images. This process ensures that each sample serves as both training and validation data at different points, thereby maximizing data utilization and offering a more reliable estimation of generalization error.

### 3.6.1 Fold Partitioning Strategy

The dataset was divided into three equally sized folds:

- Each fold comprises 770 training images and 385 validation images, maintaining consistent proportions across splits.
- This configuration closely mirrors the size of the official (but unlabeled) public validation set, which also contains 391 images, making it a suitable proxy for real-world deployment conditions.

Unlike stratified sampling used in classification tasks, fold partitioning in OCR was performed uniformly, without regard to label frequency, due to the diversity and sparsity of Vietnamese text labels. The intent was to preserve layout variability and natural imbalance in field occurrence (e.g., TOTAL_COST, TIMESTAMP) across all folds.

### 3.6.2 Sequential Training and Evaluation

As shown in **Figure 3.8** (*Pipeline Overview with Fold Processing*), each fold was independently passed through the entire OCR pipeline. This includes:

- Text detection using PaddleOCR's DBNet.
- Rotation correction using a fine-tuned MobileNetV3 classifier.
- Filtering of text regions based on IoU > 0.3.
- Cropping of high-quality regions used for fine-tuning two recognition model types:
    - Transformer-based model
    - LSTM-based seq2seq model

Each model was trained on the training subset and evaluated on the corresponding validation subset for that fold.

### 3.6.3 Model Selection Criteria

For each fold, the fine-tuned models were assessed using Character Error Rate (CER) on the validation set:

- The model with the lowest CER per fold was selected as the best performing.
- This allowed comparison between pretrained-only and fine-tuned models under identical validation conditions.

The total number of cropped text regions used for training varied per fold due to detection differences and the number of high-IoU boxes:

- Fold 0: 3,459 training cropped images, 1,727 validation
- Fold 1: 3,476 training cropped images, 1,710 validation
- Fold 2: 3,437 training cropped images, 1,749 validation

This variation reflects realistic data irregularities and ensures that model robustness is tested across differently sized training samples.

## 3.7 Mini-Batch Gradient Descent and Character Error Rate (CER)

### 3.7.1 Mini-Batch Gradient Descent in Model Training

The recognition models used in this thesis—both the LSTM-based sequence-to-sequence and Transformer-based variants within the VietOCR framework—were fine-tuned using mini-batch gradient descent (mini-batch GD). This optimization approach enables stable and efficient learning, especially under hardware constraints and when working with moderately sized datasets.

Instead of updating model weights after every single example (as in stochastic gradient descent) or computing the gradients over the full training set (as in batch

gradient descent), mini-batch GD operates on fixed-size subsets of training data. In this thesis, a batch size of 16 was chosen for all experiments, ensuring:

- Efficient memory usage on GPU hardware,
- Stabilized gradient updates compared to stochastic updates,
- And moderate regularization effect due to mini-batch variance.

The parameter update formula used during mini-batch gradient descent is:

$$\theta := \theta - \eta \cdot \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \mathcal{L}(x_i, y_i; \theta)$$

Where:

- $\theta$ denotes the vector of trainable model parameters,
- $\eta$ is the learning rate,
- $m$ = 16 is the mini-batch size,
- $L(x_i, y_i; \theta)$ represents the cross-entropy loss between the ground truth $y_i$ and model output on input $x_i$,
- $\nabla_\theta L$ is the gradient of the loss with respect to $\theta$.

Each model was trained for 20 epochs, where an epoch is defined as one full pass through the training set. At the end of each epoch, training and validation metrics were logged, including loss, full-sequence accuracy, and Character Error Rate (CER).

### 3.7.2 Character Error Rate (CER) Evaluation Strategy

To quantify recognition accuracy, Character Error Rate (CER) was adopted as the primary metric. It calculates the normalized Levenshtein distance between predicted and ground-truth strings, thereby capturing all three major edit types: insertions, deletions, and substitutions.

The CER formula is:

$$\mathrm{CER} = \frac{S + D + I}{N}$$

Where:

- $S$ is the number of substitutions,
- $D$ is the number of deletions,
- $I$ is the number of insertions,
- $N$ is the total number of characters in the reference string.

To ensure consistent evaluation at both sample and epoch levels, I implemented two layers of CER calculation:

#### 3.7.2.1 Epoch-Level CER :

During training and validation, the Character Error Rate (CER) was computed at the epoch level. The CER was then calculated over the entire corpus by summing the total number of character-level edit operations and dividing by the total number of characters in the ground truth:

$$\text{CER} = \frac{\sum_{j=1}^{N} \text{EditDistance}(y_j, \hat{y}_j)}{\sum_{j=1}^{N} |y_j|}$$

where $N$ is the total number of predicted samples in the epoch, $y_j$ is the ground truth string, $\hat{y}_j$ is the predicted string, and $|y_j|$ denotes the length (number of characters) of the ground truth string.

After each epoch, the CER of all batches in both the training and validation phases was computed and logged. This allowed tracking of convergence behavior and performance generalization over time.

This dual-level evaluation ensured a fine-grained view of recognition performance and helped to early-stop training based on validation CER when necessary.

In particular, this evaluation strategy was crucial for Vietnamese OCR, where diacritic sensitivity and character integrity are critical for semantic correctness (e.g., distinguishing “a”, “ă”, and “â”).

# CHAPTER 4

# EVALUATION AND DISCUSSION

## 4.1 Evaluation

### 4.1.1 Text Detection

To evaluate the performance of the text detection stage, I employed the pretrained PaddleOCR DBNet model on the entire set of 1,155 annotated training images. Detection performance was measured using the Intersection over Union (IoU) metric, which quantifies the overlap between predicted bounding boxes and ground-truth boxes:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Across the dataset, the average IoU was found to be 0.7863. While this figure indicates reasonably accurate detection overall, it is not particularly high due to a fundamental limitation in the dataset: the ground-truth annotations are incomplete. Specifically, many regions containing text are present in the images but are not annotated in the ground-truth labels. Consequently, high-quality detections may still be unfairly penalized in this metric.

Because of this limitation, IoU is only used here as a rough indicator of detection alignment, rather than a strict performance benchmark.

To address this and ensure quality inputs for recognition, I introduced a filtering threshold of IoU > 0.3 to select detected boxes that reasonably overlap with annotated labels. This threshold was empirically selected based on a balance between:

- Ensuring semantic relevance (excluding boxes with low overlap that are likely irrelevant or incorrect),
- And maintaining a sufficient number of training samples for VietOCR fine-tuning.

### 4.1.2 Rotation Corrector

The second major preprocessing step involves correcting the orientation of detected text regions prior to recognition. For this purpose, I reused and deployed the rotation correction model created by the SDSV_AICR team, which participated in the MCOCR 2021 competition.

This team provided a MobileNetV3-based classifier trained on a custom augmented dataset, achieving 99% classification accuracy at epoch 473. Their approach involved manually labeling and augmenting thousands of text crops to classify them into one of four rotation angles (0°, 180°), and the result was a robust lightweight model suitable for rotation estimation in Vietnamese OCR tasks.

I integrated their model into the pipeline without modification, and it served to correct skewed or misaligned cropped images that otherwise would degrade recognition accuracy. The corrected images were then passed directly to the VietOCR module.

### 4.1.3 Text Recognition (VietOCR)

#### 4.1.3.1 Pretrained Model

To assess the baseline performance of different sequence-to-sequence architectures for Vietnamese text recognition, I first evaluated two pretrained models: Transformer-based VietOCR and LSTM-based VietOCR. The evaluation was conducted across three cross-validation folds, and the key performance metrics reported include Validation Loss, Character Error Rate (CER), Full Sequence Accuracy, and Per Character Accuracy.

**Table 4.1** Transformer-based Pretrained Model

| Fold | Val Loss | CER | Acc Full Seq | Acc Per Char |
|---|---|---|---|---|
| 0 | 0.7093 | 0.0888 | 0.4624 | 0.8354 |
| 1 | 0.6986 | **0.0871** | 0.4721 | 0.8516 |
| 2 | 0.7161 | 0.0946 | 0.4713 | 0.8371 |

The Transformer model achieved moderate accuracy with a Character Error Rate (CER) around 9–10%, indicating that although it recognizes many characters correctly, it struggles with full-sequence accuracy (only ~47%). The per-character accuracy across folds ranges from 83.5% to 85.1%, suggesting the model performs reasonably well at the character level but tends to misplace or omit characters when generating entire sequences.

**Table 4.2** LSTM-based Pretrained Model

| Fold | Val Loss | CER | Acc Full Seq | Acc Per Char |
|---|---|---|---|---|
| 0 | 0.6727 | 0.0543 | 0.5504 | 0.8867 |
| 1 | 0.6560 | **0.0500** | 0.5721 | 0.9026 |
| 2 | 0.6686 | 0.0513 | 0.5757 | 0.8905 |

The LSTM model significantly outperformed the Transformer in all metrics. Its CER was consistently lower (~4.7% to 5.0%), and it showed higher full-sequence accuracy (~55%–57%) across all folds. Furthermore, the per-character accuracy exceeded 88%, reaching up to 90.26% in Fold 1, which highlights the model's strength in character-level precision and sequence consistency.

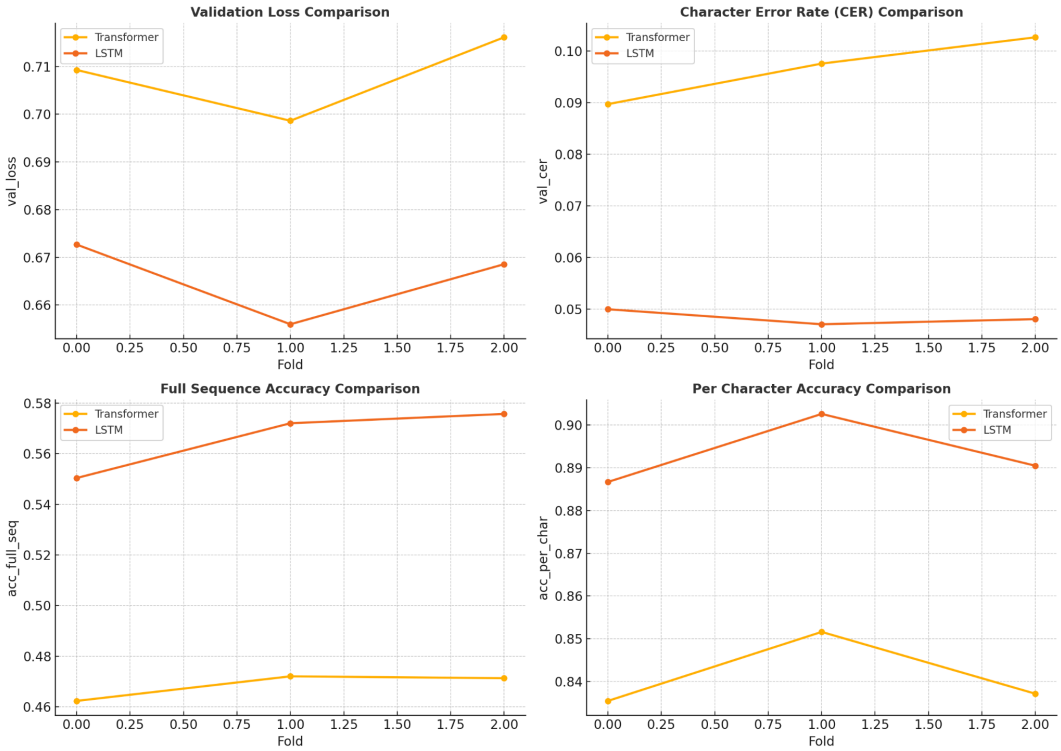



**Figure 4.1** Evaluation of Pretrained Model

These results and **Figure 3.14** indicate that, even without fine-tuning, the LSTM-based VietOCR exhibits better generalization and robustness for Vietnamese text recognition compared to the Transformer model. The LSTM's recurrent architecture, which is inherently well-suited for capturing sequential dependencies, may explain its advantage on this task—particularly when dealing with the structured and moderately long sequences typical of receipt-like documents in the MCOCR dataset.

#### 4.1.3.2 Fine-tuned Model

After establishing the baseline performance of the pretrained VietOCR models, I proceeded to fine-tune both the Transformer and LSTM architectures using fold-specific training sets from the MCOCR dataset. The models were trained for a fixed number of epochs, and the best-performing checkpoint in each fold was selected based on the lowest Character Error Rate (CER) on the validation set.

**Table 4.3** Evaluation of fine-tuned Transformer

| Fold | Val Loss | Val CER | Acc Full Seq | Acc Per Char | Epoch | Duration (s) |
|---|---|---|---|---|---|---|
| 0 | 0.6590 | 0.0370 | 0.6641 | 0.9173 | 17 | 64.70 |
| 1 | 0.6444 | **0.0362** | 0.6581 | 0.9268 | 16 | 64.58 |
| 2 | 0.6555 | 0.0411 | 0.6529 | 0.9144 | 14 | 64.24 |

The fine-tuned Transformer model demonstrates notable improvements across all three folds. The Character Error Rate (CER) decreased significantly, with the lowest CER of 0.0362 in Fold 1, followed by 0.0370 in Fold 0, while Fold 2 showed a slightly higher CER of 0.0411, possibly indicating that this fold contains more challenging or less representative samples.

Full sequence accuracy improved to values between 0.6529 and 0.6641, reflecting better overall sequence prediction. Per-character accuracy was consistently high across all folds, reaching up to 0.9268 in Fold 1, which indicates strong consistency in individual character recognition.

Overall, the fine-tuning process led to substantial performance gains, particularly in reducing CER and enhancing accuracy. Fold 2's slightly weaker results suggest further analysis may be beneficial to improve robustness on more difficult subsets.

**Table 4.4** Evaluation of fine-tuned LSTM

| Fold | Val Loss | Val CER | Acc Full Seq | Acc Per Char | Epoch | Duration (s) |
|---|---|---|---|---|---|---|
| 0 | 0.6304 | 0.0340 | 0.6641 | 0.9240 | 5 | 53.67 |
| 1 | 0.6182 | 0.0329 | 0.6588 | 0.9263 | 4 | 55.08 |
| 2 | 0.6322 | **0.0328** | 0.6853 | 0.9323 | 5 | 54.72 |

The fine-tuned LSTM model consistently outperformed the Transformer model across all folds. It achieved the lowest Character Error Rate (CER) of 0.0328 on Fold 2, with Fold 1 and Fold 0 close behind at 0.0329 and 0.0340, respectively. This demonstrates strong consistency and robustness across different data splits.

In terms of full sequence accuracy, the LSTM model reached up to 0.6853 on Fold 2, outperforming all other configurations, while maintaining scores of 0.6588 and 0.6641 on the remaining folds. Per-character accuracy remained high across all folds, exceeding 0.9240, with the best result of 0.9323 also observed in Fold 2.

Notably, the LSTM model converged in just 4–5 epochs per fold, with an average training duration of approximately 54 seconds per epoch, highlighting its efficiency.

Overall, these results reaffirm the effectiveness of the LSTM-based VietOCR model. Compared to the Transformer counterpart, it not only achieves better accuracy and lower CER but also does so with faster convergence. The strongest overall performance was recorded on Fold 2, making it the most promising candidate for Vietnamese text recognition tasks requiring both precision and efficiency.

## 4.2 Discussion

### 4.2.1 Transformer-Based Model

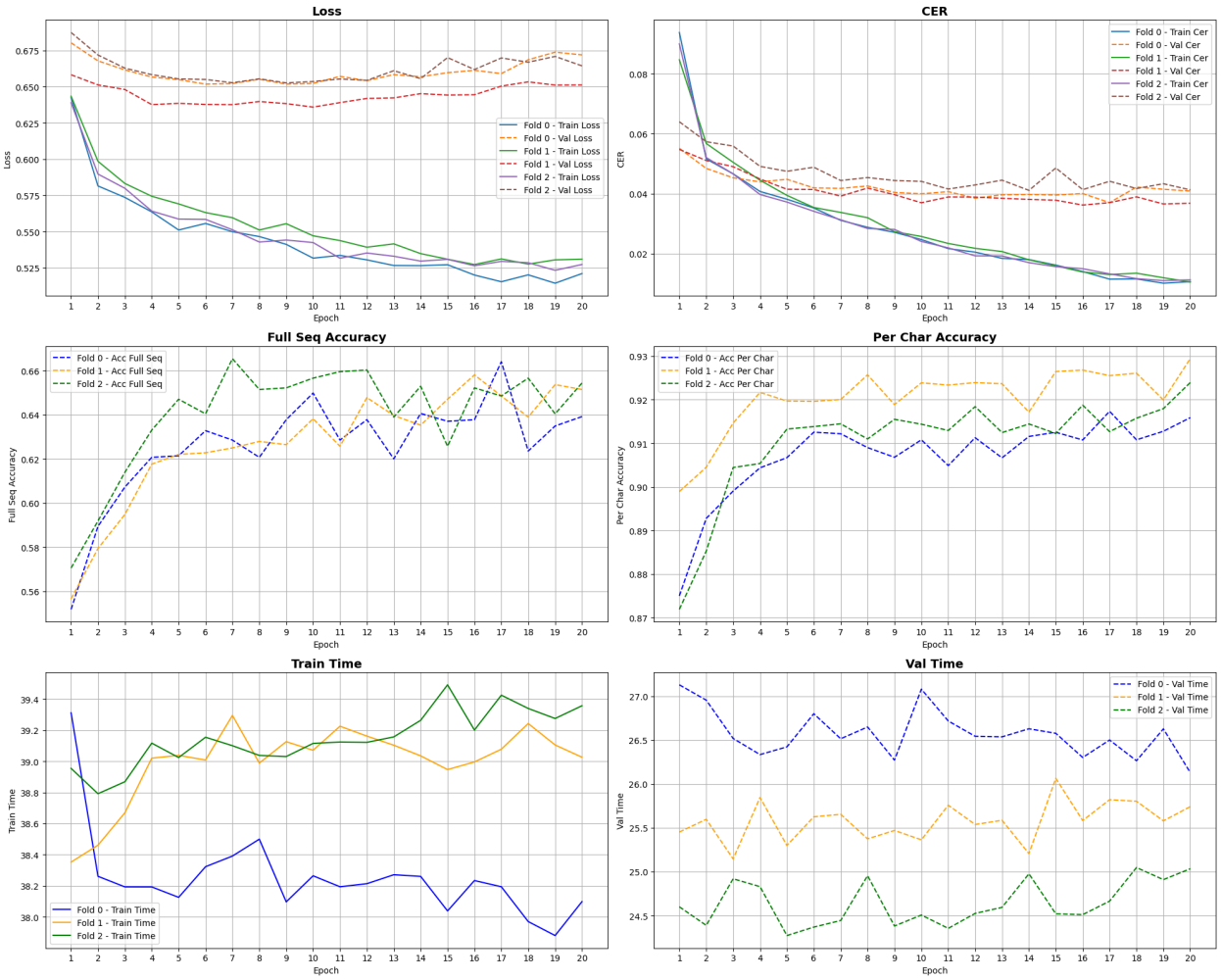



**Figure 4.2** Evaluation across all folds Transformer-Based Model

The fine-tuning results of the Transformer-based VietOCR model reveal several key performance patterns across the three folds, as clearly visualized in the training plots.

Across all folds, training loss steadily decreased with each epoch, indicating effective optimization and learning. In contrast, validation loss began to plateau around epoch 10, fluctuating slightly but showing no significant increase. This behavior suggests

that the model avoided overfitting and maintained stable generalization. The consistent gap between training and validation losses across epochs reflects the model's robust performance on unseen data.

The CER (Character Error Rate) curves exhibited a rapid decline during the initial epochs and gradually converged to low values ranging from 0.0362 to 0.0411. Notably, validation CER remained equal to or slightly lower than training CER after epoch 10 in all folds. This pattern strongly suggests that the model learned generalizable features instead of memorizing the training set.

A comparison across folds highlights subtle performance trade-offs:

- Fold 0 achieved the CER at 0.0370, indicating highly accurate character-level. It also gains the highest full-sequence accuracy at 0.6603.
- Fold 1 demonstrated the best performance, with Acc Per Char peaking at 0.9268, the highest among all folds. Its CER remained consistently lowest at **0.0362** while maintaining strong full-sequence accuracy.
- Fold 2 came at the cost of a slightly higher CER (0.0411), implying occasional character-level inconsistencies despite strong sequence-level output.

In summary, the fine-tuned Transformer model excelled at character-level text recognition, achieving stable and low CER values across all folds. While per-character accuracy remained consistently high, full-sequence accuracy continues to pose a greater challenge, reflecting the stricter nature of exact sequence matching in OCR tasks. Despite

variations in training data splits, the model maintained strong generalization ability and robust performance, underscoring its effectiveness in Vietnamese OCR applications.

### 4.2.2 LSTM-Based Model

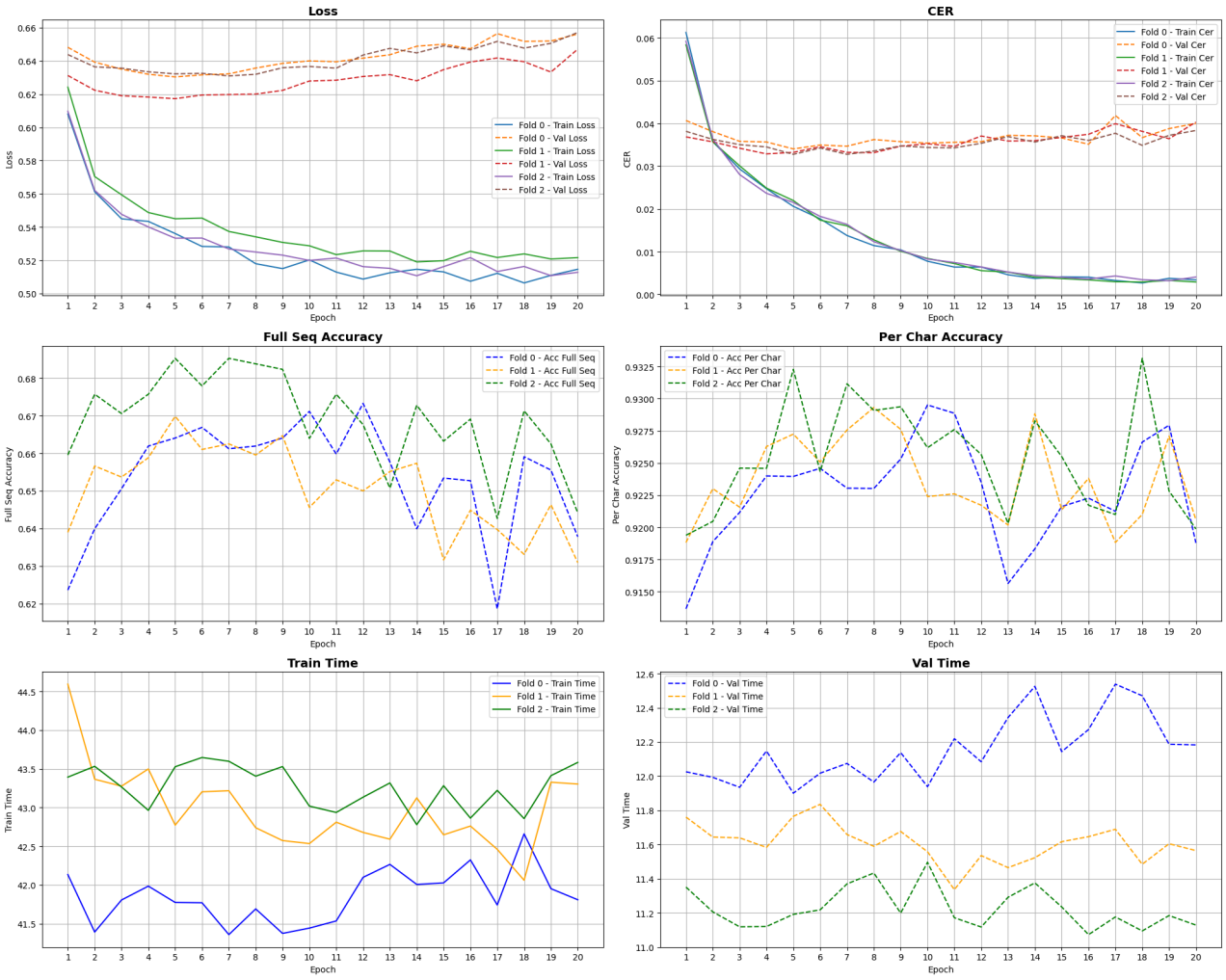



**Figure 4.3** Evaluation across all folds LSTM-Based Model

The LSTM-based variant of the VietOCR model demonstrated consistently strong performance across all three folds, clearly outperforming the Transformer-based counterpart in terms of convergence stability and recognition accuracy.

All folds exhibited smooth and steadily decreasing training and validation losses, with no signs of divergence or overfitting throughout the 20 epochs. In fact, the LSTM model maintained a lower final validation loss across all folds when compared to the Transformer, for instance, Fold 1 reached a minimum validation loss of 0.6182, while the Transformer's best was around 0.6444. This suggests that the LSTM model converged more effectively under the current setup.

The CER (Character Error Rate) dropped sharply within the first few epochs and stabilized at exceptionally low levels, consistently outperforming the Transformer model. Fold 2 reached the lowest CER of **0.0328**, while Fold 1 and Fold 0 closely followed with CER values of 0.0329 and 0.0340, respectively. All of these outperformed the Transformer's best CER (~0.0362). This reinforces the LSTM's strength in Vietnamese character-level recognition, likely due to its sequential modeling capabilities.

Accuracy metrics further highlight the model's robustness:

- Fold 2 achieved the highest Full Sequence Accuracy of 0.6853 and the highest Per Character Accuracy of 0.9331.
- Fold 1 also excelled, with a Full Seq Acc of 0.6588 and Per Char Acc of 0.9263, indicating strong generalization.
- Fold 0 was comparably strong, reaching 0.6641 in Full Seq Acc and 0.9240 in Per Char Acc.

From a computational perspective, training and validation times per epoch remained consistent and efficient, averaging ~42–44 seconds for training and ~11–12

seconds for validation, indicating the LSTM model is not only accurate but also computationally tractable.

### 4.2.3 Comparison

| Metric | Transformer | LSTM | Conclusion |
|---|---|---|---|
| **Best Validation CER** | 0.0362 (Fold 1) | **0.0328 (Fold 2)** | LSTM outperforms Transformer |
| **Validation CER Stability** | Some fluctuations across folds | Stable and consistent across folds | LSTM shows better reliability |
| **Full Sequence Accuracy** | Up to 0.6581 | Up to 0.6853 | LSTM performs better overall |
| **Per Character Accuracy** | Up to 0.9268 | Up to 0.9323 | LSTM slightly better |
| **Avg. Training Time per Fold** | ~1287.9 seconds | **~1087.2 seconds** | LSTM is more efficient (~15.6% faster) |
| **Scalability to Larger Datasets** | Higher cost due to training time | More suitable and scalable | LSTM scales better |

**Table 4.5** Comparative Summary of Transformer and LSTM Models in Text Recognition Performance and Efficiency

In this study, two variants of the VietOCR architecture — one based on a Transformer and the other on LSTM — were trained and evaluated across three folds using a consistent experimental setup. The models were compared across four main metrics: Validation Loss, Character Error Rate (CER), Full Sequence Accuracy, and Per Character Accuracy, as presented in **Table 4.5**.

Among these, Full Sequence Accuracy and Per Character Accuracy, while intuitive, showed limited reliability in this experiment due to high variance and non-monotonic fluctuations across epochs, particularly for the Transformer. The inconsistent behavior of these metrics indicates instability in prediction patterns and sensitivity to small changes in model weights. As a result, they were not considered reliable indicators of convergence or robustness.

Instead, Character Error Rate (CER) and Validation Loss served as more stable and informative metrics. The LSTM-based VietOCR consistently outperformed the Transformer in terms of CER across all three folds. The lowest CER achieved was 0.0328 (Fold 2), while the Transformer's best CER was 0.0362 (Fold 1). Even the worst CER from the LSTM model was lower than the best result from the Transformer, highlighting LSTM's consistent superiority in character-level recognition. This performance advantage is likely attributed to the recurrent structure of LSTM, which captures sequential dependencies more effectively which is a critical factor for accurate Vietnamese OCR.

A clear difference was also observed in training efficiency. The Transformer model required more training time per fold:

- Fold 0: 1296.6s, Fold 1: 1291.4s, Fold 2: 1275.7s
- Average: ~1287.9s

In contrast, the LSTM model trained significantly faster:

- Fold 0: 1080.2s, Fold 1: 1091.7s, Fold 2: 1089.8s

- Average: ~1087.2s

This represents an average reduction of approximately 15.6% in training time. While this gap may appear modest in a small-scale study, it becomes increasingly impactful in large-scale or production environments, where compute resources and time budgets are constrained.

In summary, although both architectures are capable of learning the OCR task, the LSTM-based model clearly demonstrates superior performance across the most reliable metrics (CER and Validation Loss), and it does so with lower training cost and higher consistency. Conversely, the Transformer-based model, despite its theoretical power, suffered from greater instability in accuracy metrics and longer training times. These findings suggest that for Vietnamese OCR tasks, especially under constraints of limited data or compute, the LSTM-based VietOCR model offers a more practical, efficient, and robust solution.

### 4.2.4 Practical Implications and Competition Readiness

Following the fine-tuning and evaluation process on the K-Fold dataset, I successfully contacted the organizers of the MCOCR competition and have received access to the private test set, which includes 390 images. However, the corresponding annotations remain hidden and can only be evaluated by submitting predictions to the competition's official website.

At present, due to technical issues, the submission function on the website is temporarily unavailable, and I am currently awaiting further support and instructions from the organizers. Despite this limitation, the strong fine-tuning results achieved across

the three folds, particularly with the LSTM-based model reaching a minimum Character Error Rate (CER) of 0.0328, provide high confidence that the model will perform well when applied to the real test data of the competition.

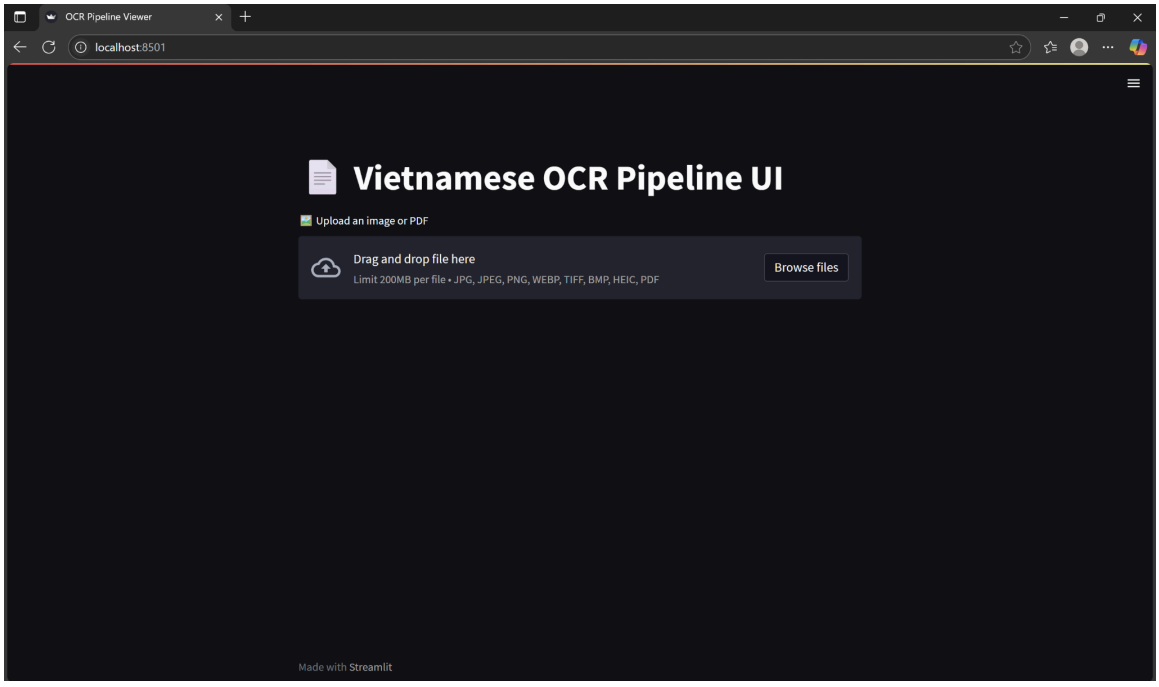
Moreover, the comprehensive evaluation of both recognition performance and training efficiency has reaffirmed that the LSTM-based architecture is better suited for Vietnamese OCR tasks, especially in scenarios with limited data and computational resources. These findings not only hold academic significance but also carry strong practical value, suggesting the LSTM model as a viable and effective solution for real-world OCR deployment.

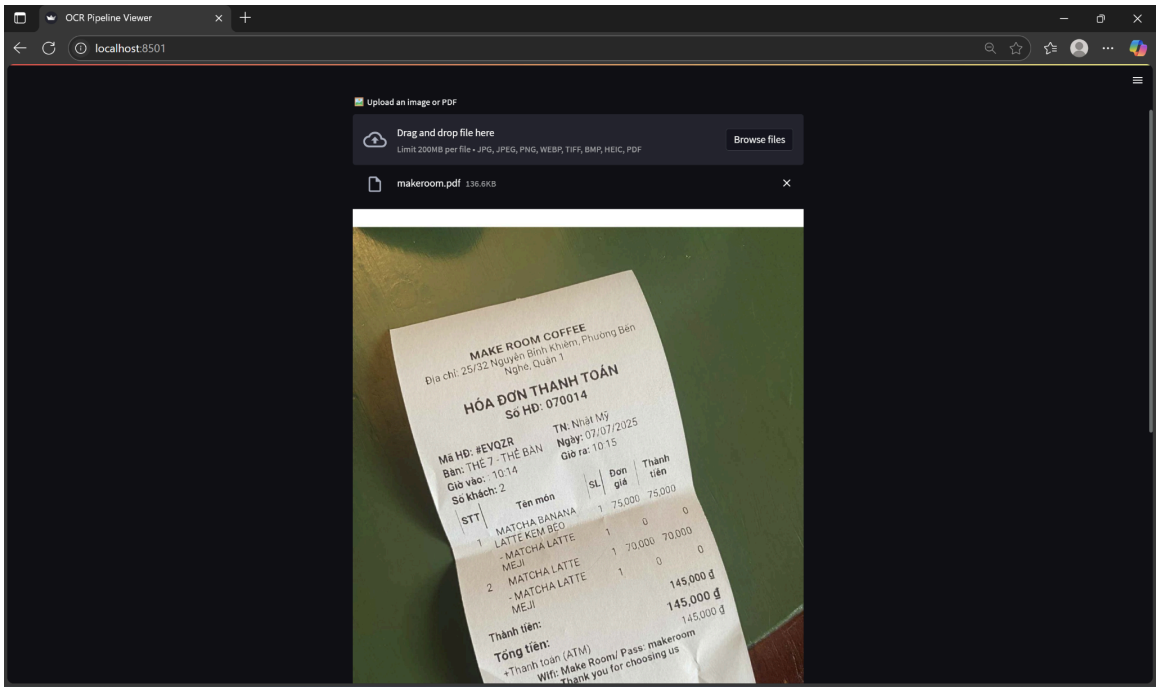
To further demonstrate the practical applicability of the proposed OCR system, a web-based user interface was developed using Streamlit. This interface enables users to upload document images in a wide range of formats including .jpg, .png, .webp, and .pdf which are automatically converted and processed through the full OCR pipeline, encompassing text detection, rotation correction, and recognition.

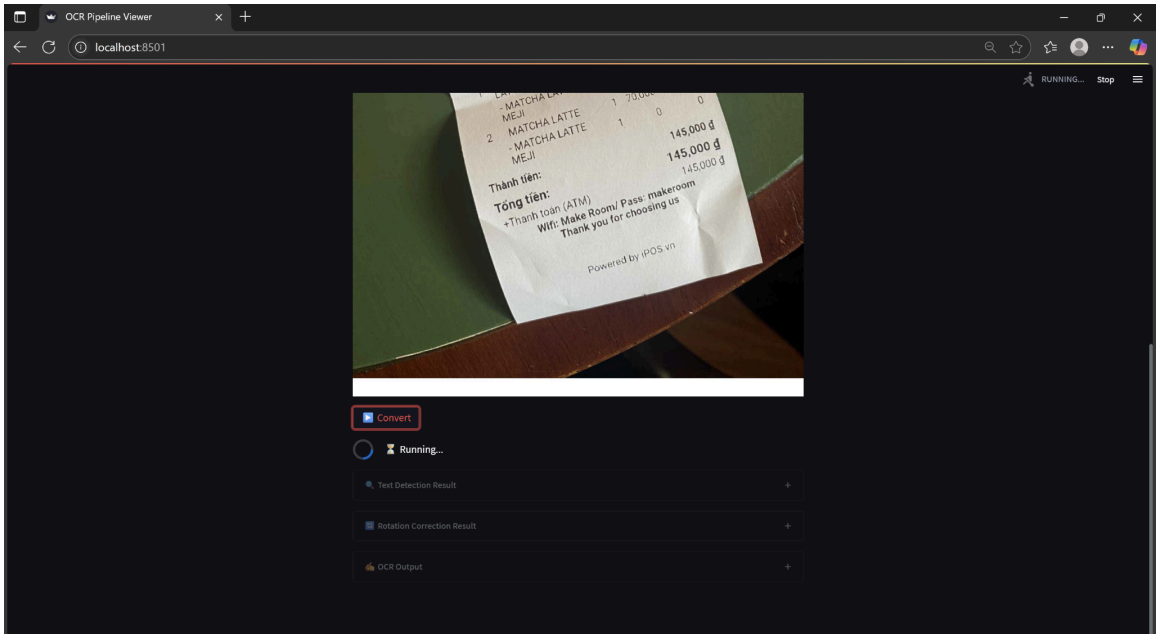

The interface then presents the extracted text alongside visual feedback from each processing stage, such as the original input, the detected text regions, and the final rotation-corrected images. Users can download the recognized text in either .txt or .doc format for further use, as well as export the corrected images. This functionality enables

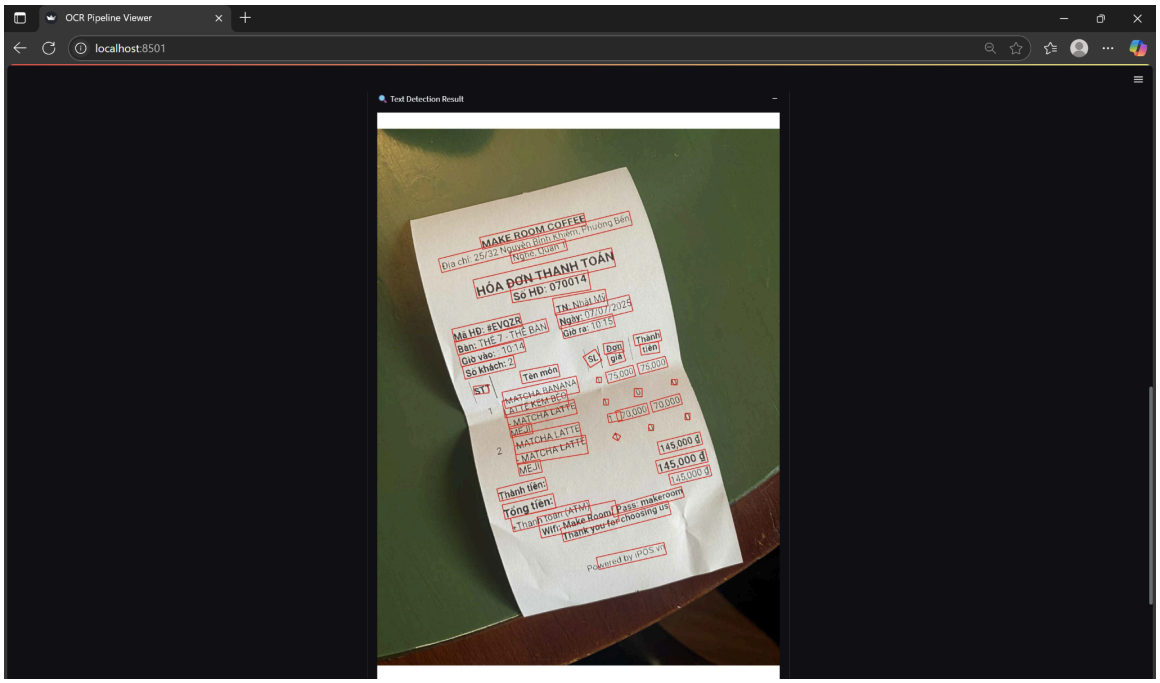
seamless integration into administrative workflows, especially in digital archiving, data entry automation, and document digitization tasks.

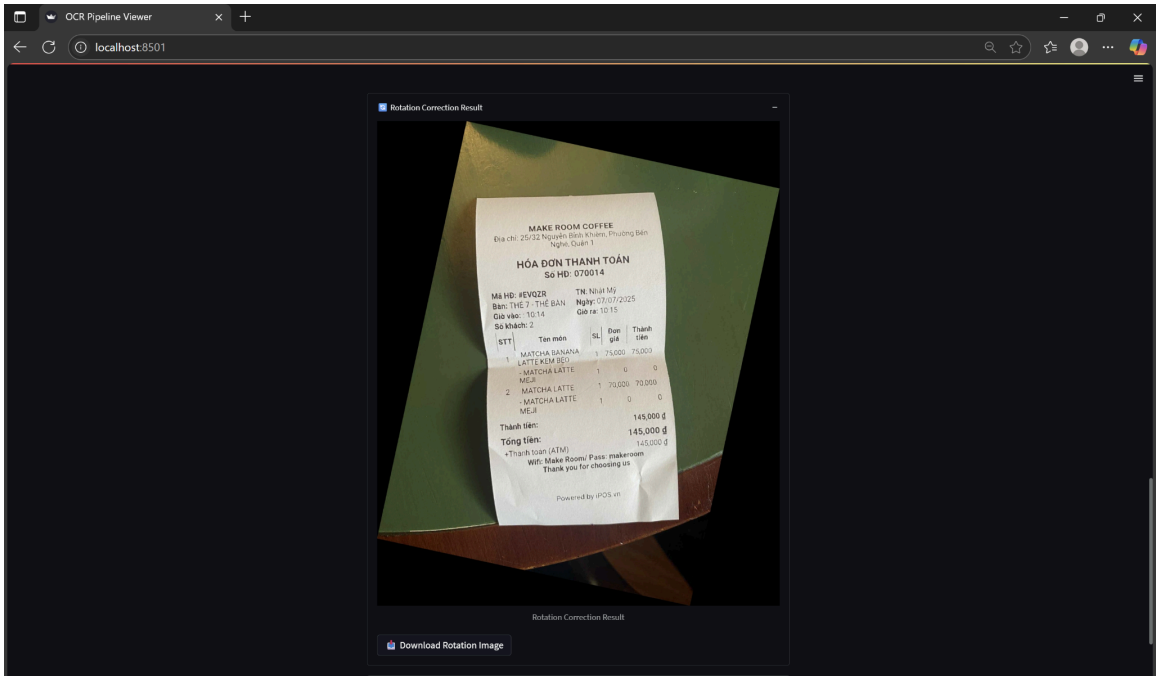
Although currently implemented as a single-page shown in **Figure 4.4** application for demonstration purposes, the interface encapsulates the end-to-end pipeline effectively and requires minimal technical expertise to operate. Its responsiveness, format flexibility, and download capability make it a strong candidate for integration into larger document processing systems or public-facing OCR services. Future iterations may include multi-language support, session logging, or batch processing to enhance usability and scalability in real-world deployments.




OCR Pipeline Viewer
localhost:8501
Upload an image or PDF
Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG, WEBP, TIFF, BMP, HEIC, PDF
Browse files
makeroom.pdf 136.6KB
MAKE ROOM COFFEE
Địa chỉ: 25/32 Nguyễn Bỉnh Khiêm, Phường Bến Nghé, Quận 1
HÓA ĐƠN THANH TOÁN
Số HĐ: 070014
Mã HĐ: #EVQZR
Bàn: THẺ 7 - THẺ BÀN
Giờ vào: 10:14
Số khách: 2
TN: Nhật Mỹ
Ngày: 02/07/2025
Giờ ra: 10:15
STT
Tên món
SL
Đơn giá
Thành tiền
1 MATCHA BANANA LATTE KEM BÉO 1 75,000 75,000
2 - MATCHA LATTE MEJI 1 70,000 70,000
MATCHA LATTE - MATCHA LATTE MEJI 1 145,000 ₫
Thành tiền: 145,000 ₫
Tổng tiền: 145,000 ₫
+Thanh toán (ATM)
Wifi: Make Room/ Pass: makeroom
Thank you for choosing us


OCR Pipeline Viewer
localhost:8501
RUNNING...
Stop
Thành tiền: 145,000 ₫
Tổng tiền: 145,000 ₫
+Thanh toán (ATM)
Wifi: Make Room/ Pass: makeroom
Thank you for choosing us
Powered by iPOS.vn
Convert
Running...
Text Detection Result
Rotation Correction Result
OCR Output


OCR Pipeline Viewer
localhost:8501
Text Detection Result
MAKE ROOM COFFEE
HÓA ĐƠN THANH TOÁN
Số HĐ: 070014
Thanh toán (ATM)
Wifi: Make Room/ Pass: makeroom
Thank you for choosing us
Powered by iPOS.vn


OCR Pipeline Viewer
localhost:8501
Rotation Correction Result
MAKE ROOM COFFEE
Địa chỉ: 25/32 Nguyễn Bỉnh Khiêm, Phường Bến Nghé, Quận 1
HÓA ĐƠN THANH TOÁN
Số HĐ: 070014
Mã HĐ: #EVQZR
TN: Nhật Mỹ
Bàn: THẺ 7 - THẺ BÀN
Ngày: 07/07/2025
Giờ vào: 10:14
Giờ ra: 10:15
Số khách: 2
STT
Tên món
SL
Đơn giá
Thành tiền
1
MATCHA BANANA LATTE KEM BÉO
75,000
75,000
- MATCHA LATTE MẺ/I
2
MATCHA LATTE
1
70,000
70,000
- MATCHA LATTE MẺ/I
Thành tiền:
145,000 ₫
Tổng tiền:
145,000 ₫
145,000 ₫
•Thanh toán (ATM)
Wifi: Make Room/ Pass: makeroom
Thank you for choosing us
Powered by iPOS.vn
Rotation Correction Result
Download Rotation Image

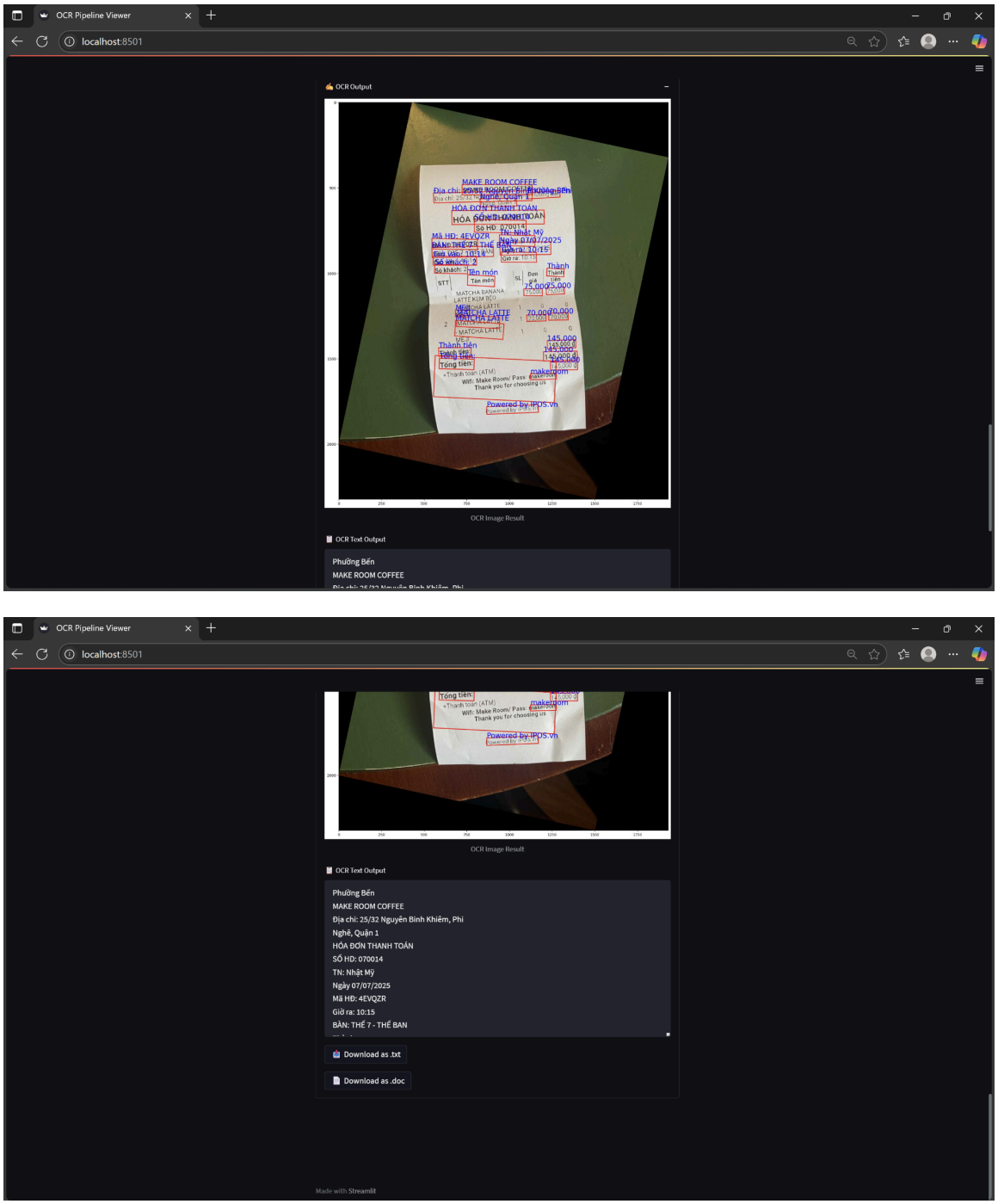



**Figure 4.4** WebUI

# CHAPTER 5

# CONCLUSION AND FUTURE WORK

## 5.1 Conclusion

This thesis presented the design and implementation of a comprehensive and robust pipeline for Vietnamese Optical Character Recognition (OCR), integrating a sequence of modules tailored to address real-world challenges in text extraction, correction, and recognition from diverse document images. The proposed system successfully incorporates several key components:

- A rule-based data cleaning mechanism to address common annotation issues such as label inversion and missing fields;
- A text detection stage leveraging PaddleOCR's DBNet pretrained model for high-quality bounding box proposals;
- A rotation correction module using MobileNetV3-Small, trained on augmented and corrected data, to normalize text orientation prior to recognition;
- An IoU-based filtering scheme to refine the quality of input crops passed to the recognition model;
- A recognition component based on VietOCR, in which two architectures, CNN + LSTM Seq2Seq and CNN + Transformer, were fine-tuned and evaluated through a rigorous 3-fold cross-validation setup;

- And a performance assessment utilizing metrics such as Character Error Rate (CER), Full Sequence Accuracy, and Per-Character Accuracy, in addition to training time measurements and visual performance tracking.

Across all experiments, the LSTM-based model outperformed its Transformer counterpart in terms of lowest average CER while maintaining faster convergence and reduced training time. These results strongly support the LSTM-based seq2seq architecture as a more suitable option for Vietnamese OCR in limited-resource and low-annotation environments.

## 5.2 Future Work

Building on the outcomes of this thesis, several promising research directions are identified to further enhance the accuracy, efficiency, and adaptability of the OCR pipeline.

On the one hand, Character-Level Recognition Architecture, the current sequence recognition framework relies on word- or line-level decoding, which can be insufficient when encountering issues such as tightly packed diacritics, unusual spacing, or noisy backgrounds. To overcome this, I plan to design a character-level recognition model, where each character (glyph) is processed individually. This granularity is expected to enable:

- Finer-grained feature extraction, capturing subtle distinctions in Vietnamese diacritics;
- Improved generalization in multi-line or semi-structured layouts;

- Localized correction, allowing partial word recovery in corrupted segments.

This architecture would require careful adaptation of the decoding scheme and training labels but could significantly improve robustness in noisy or semi-formal document settings.

On the other hand, Reinforcement Learning for OCR Decoding, while supervised learning has proven effective for OCR, it often lacks adaptability when encountering unseen or degraded data. As an alternative, I propose integrating Reinforcement Learning (RL) into the recognition pipeline.

Using policy gradient or actor-critic algorithms, the model could iteratively improve its decoding policy based on historical performance. Such a setup enables dynamic correction mechanisms that go beyond greedy decoding or beam search, particularly in cases of partial occlusion, irregular fonts, or misaligned annotations.

By pursuing these directions, this research aspires not only to improve current Vietnamese OCR systems but also to contribute to a broader class of adaptable, error-resilient document understanding tools suitable for Southeast Asian scripts and multilingual environments.

# REFERENCES

[1] Smith, R. (2007). An Overview of the Tesseract OCR Engine. *Ninth International Conference on Document Analysis and Recognition (ICDAR)*, 629–633.

[2] Islam, N., Islam, Z., & Noor, N. (2017). A Survey on Optical Character Recognition System. *Journal of Information Technology & Software Engineering*, 7(2), 1–6.

[3] Duong, K. N., Pham, D. S., & Tran, H. T. (2022). Deep Learning-based Vietnamese Text Recognition from Complex Background Images. *International Journal of Pattern Recognition and Artificial Intelligence*, 36(10).

[4] Bui, M. H., Vu, X. T., & Tran, D. T. (2021). *MC-OCR Challenge: Mobile-Captured Image Document Recognition for Vietnamese Receipts.*

[5] Du, Y., Li, C., Guo, R., & Wang, X. (2021). PP-OCRv2: Bag of Tricks for Ultra Lightweight OCR System. *arXiv preprint arXiv:2109.03144*.

[6] Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L. C., Tan, M., ... & Adam, H. (2019). Searching for MobileNetV3. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 1314–1324.

[7] Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, 1137–1143.

[8] Levenshtein, V. I. (1966). Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet Physics Doklady*, 10(8), 707–710.

[9] D. Liu et al., “PaddleOCR: A Practical, High-Performance OCR System,” *Proceedings of the International Conference on Document Analysis and Recognition*, 2020.

[10] S. Baek, M. Kwon, and E. Kim, “Character Region Awareness for Text Detection,” CVPR, 2019.

[11] H. Howard et al., “Searching for MobileNetV3,” *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[12] M. Jaderberg et al., “Spatial Transformer Networks,” NeurIPS, 2015.

[13] SDSV_AICR Team, “Winning Solution for MCOCR 2021,” *GitHub Repository*, 2021.

[14] B. Shi, X. Bai, and C. Yao, “An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2298–2304, 2017.

[15] A. D. Le, H. T. Nguyen, M. Nakagawa, “End-to-End Recognition System for Recognizing Offline Unconstrained Vietnamese Handwriting,” *arXiv* preprint arXiv:1905.05381, 2019.

[16] C. Carbune *et al.*, “ICFHR 2018 Competition on Vietnamese Online Handwritten Text Recognition (VOHTR2018),” *ICFHR Workshop*, 2018.

[17] A. Vaswani et al., “Attention Is All You Need,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[18] VietOCR Contributors, “VietOCR: Open-source OCR Toolkit for Vietnamese and English,” *GitHub Repository*, 2020.

[19] S. Pan and Q. Yang, “A Survey on Transfer Learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[20] J. Brownlee, *Cross-Validation for Predictive Modeling*, Machine Learning Mastery, 2018.

[21] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” *arXiv preprint arXiv:1804.02767*, 2018.

[22] W. Liu, D. Anguelov, D. Erhan *et al.*, “SSD: Single Shot MultiBox Detector,” *European Conference on Computer Vision (ECCV)*, pp. 21–37, 2016.

[23] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. “Real-time Scene Text Detection with Differentiable Binarization.” In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 2020.

[24] Kuang, Zhanghui, et al. “MMOCR: A Comprehensive Toolbox for Text Detection, Recognition and Understanding.” *arXiv preprint* arXiv:2108.06543, 2021.

[25] Kim, Geewook et al. “Donut: Document Understanding Transformer without OCR.” *arXiv preprint* arXiv:2111.15664, 2022.

[26] Li, Yiheng et al. “TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models.” *arXiv preprint* arXiv:2109.10282, 2021.

[27] Dosovitskiy, Alexey et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.” *International Conference on Learning Representations (ICLR)*, 2021.

[28] Hamdi, Ahmed et al. “VISTA-OCR: Towards generative and interactive end to end OCR models.” *arXiv preprint* arXiv:2404.03621, 2024.

[29] Wang Peng, An Yang et al. “OFA: Unifying Architectures, Tasks, and Modalities Through a Simple Sequence-to-Sequence Learning Framework” *ICML2022*.

[30] Dropbox Tech Blog. “Inside Dropbox’s Document Scanner.” *Dropbox.tech*, 2017.

[31] Signzy. “OCR Pipeline Built Using Deep Learning.” *Signzy.com*, 2018.

[32] Neptune.ai. “How to Build Deep Learning-Based OCR Systems: Lessons from the Field.” *Neptune Blog*, 2025.

[33] Buslaev, Alexander et al. “Albumentations: Fast and Flexible Image Augmentations.” *Information*, 2020.

[34] Fogel, Shahar, et al. “ScrabbleGAN: Semi-Supervised GAN for Text Recognition.” *CVPR*, 2020.

[35] Xu, Yiheng et al. “LayoutXLM: Multimodal Pre-training for Multilingual Visually-rich Document Understanding.” *ACL*, 2021.

[36] Karatzas, Dimosthenis et al. “ICDAR 2013 Robust Reading Competition.” *Proceedings of ICDAR*, 2013.

[37] M. Sandler et al., "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4510–4520, 2018.