THE UNIVERSITY OF DANANG
DANANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

GRADUATION PROJECT THESIS

MAJOR: INFORMATION TECHNOLOGY

SPECIALTY: SOFTWARE ENGINEERING

PROJECT TITLE:

AUDIO CONVERSION APPLICATION FOR IMAGE-BASED DOCUMENTS

Instructor: **Ph.D. PHAM CONG THANG**
Student: **NGUYEN PHAM NAM ANH**
Student ID: **102200162**
Class**:** **20TCLC_DT4**

Da Nang, 06/2024

THE UNIVERSITY OF DANANG
DANANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

GRADUATION PROJECT THESIS

MAJOR: INFORMATION TECHNOLOGY

SPECIALTY: SOFTWARE ENGINEERING

PROJECT TITLE:

AUDIO CONVERSION APPLICATION FOR IMAGE-BASED DOCUMENTS

Instructor: **Ph.D. PHAM CONG THANG**
Student: **NGUYEN PHAM NAM ANH**
Student ID: **102200162**
Class**:** **20TCLC_DT4**

Da Nang, 06/2024

DA NANG UNIVERSITY
UNIVERSITY OF SCIENCE AND TECHNOLOGY
INFORMATION TECHNOLOGY FACULTY

SOCIALIST REPUBLIC OF VIETNAM
Independence - Freedom - Happiness

GRADUATION PROJECT COMMENT

I. **General information:**

1. Student name: Nguyen Pham Nam Anh
2. Class: 20TCLC_DT4 Student ID: 102200162
4. Topic title: Audio conversion application for image-based documents
5. Instructor: Pham Cong Thang Academic title/ degree: Doctor of Philosophy

II. **Reviews of graduation project**

1. About the urgency, novelty, usability of the topic: (2 points)

……
……

2. About the results of solving the tasks required by the project: (4 points)

……
……

3. About the form, structure and layout of the graduation project: (2 points)

……
……

4. The topic includes scientific value/article/problem solving of the enterprise or school: (1 point)

……
……

5. Existing shortcoming need to be supplemented or modified:

……
……

III. **Spirit and attitude of the student (1 point):**

……
……

IV. **Evaluation:**

1. Evaluation point: …/10
2. Suggest: Defense permitted/ Edit to defend/ Defense not permitted

Da Nang, …., June, 2024

Instructor

SUMMARY

Topic title: Audio Conversion Application For Image-based Documents

Student name: Nguyễn Phạm Nam Anh

Student ID: 102200162 Class: 20TCLC_DT4

In today's digital age, managing and accessing documents efficiently is crucial. With the rise in image-based documents, there is a growing need for a tool that can effortlessly convert these images into accessible audio formats. This is where "Everead" comes in – an innovative app designed to transform image-based documents into clear, listenable audio files. Perfect for users on the go, individuals with visual impairments, or anyone looking to maximize productivity, "Everead" ensures that your documents are always just a listen away, providing convenience and accessibility like never before.

DA NANG UNIVERSITY
UNIVERSITY OF SICIENCE AND TECHNOLOGY
INFORMATION TECHNOLOGY FALCUTY

THE SOCIALIST REPUBLIC OF VIETNAM
Independence - Freedom - Happiness

GRADUATION PROJECT REQUIREMENTS

Student Name: Nguyễn Phạm Nam Anh Student ID: 102200162

Class: 102200162 Faculty: Information Technology Major: Software Engineering

1. Topic title: Audio conversion application for image-based documents

2. Project topic: ☐has signed intellectual property agreement for final result

3. Initial figure and data:

The initial figure was borrowed from BKAI – a competition about machine learning for student of Hanoi University of Science and Technology. Data for this project was mostly sourced from Google Image.

4. Content of the explanations and calculations:

- Introduction section: introduction about the topic
- Chapter 1: Application Overview and requirements
- Chapter 2: System analysis and design
- Chapter 3: System implementation
- Chapter 4: Application demo
- Chapter 5: Conclusion

5. Drawings, charts (specify the types and sizes of drawings):

- Use case Diagrams
- Activity Diagrams
- Architecture structure Diagrams

6. Name of instructor: Pham Cong Thang

7. Date of assignment: .../.../2024

8. Date of completion: .../.../2024

Da Nang, …., June, 2024

Head of Division……. **Instructor**

PREFACE AND ACKNOWLEDGEMENT

The exponential growth of the technology community and the increasing demand for efficient document management have driven the development of this application. With enthusiasm and determination, I have strived to create a product that not only meets the needs of our users but also delivers exceptional experiences. My goal is to ensure that users not only utilize the product functionally but also genuinely appreciate its design flow and underlying philosophy.

I would like to extend my deepest appreciation to my instructor, Ph.D. Pham Cong Thang, for his invaluable advice and comments, which have helped improve my application day by day. I am also grateful to the Faculty of Information Technology at Danang University of Science and Technology for providing an excellent environment and conditions for students to study and develop themselves gradually.

Although I have put forth my best effort in this project, it is inevitable that some mistakes or incompletions remain. I hope to receive comments and suggestions from lecturers to further improve my graduation thesis.

ASSURRANCE

I understand the University's policy about anti-plagiarism and guarantee that:

- The contents of this thesis project are performed following the guidance of Ph.D. Pham Cong Thang.
- All the references, which I used in this thesis, are quoted with the author's name, project's name, time, and location to publish clearly and faithfully.
- The contents of this project are my own work and have not been copied from other sources or been previously submitted for award or assessment.

Student Performed

Nguyen Pham Nam Anh

TABLE OF CONTENT

LIST OF TABLES

PICTURES

LIST OF SYMBOLS, ACRONYM

ITEMS	DESCRIPTION
API	Application Programming Interface
OCR	Optical Character Recognition
DB	Differentiable Binarization
Http	Hyper text transfer protocol
NoSQL	Non Structure Query Language

INTRODUCTION

1. Reason for Doing the Thesis

In today's digital age, people frequently encounter text within images, whether they are documents, signs, or handwritten notes. This presents challenges for accessibility, especially for visually impaired individuals and those who prefer auditory learning. To address this need, there is a growing demand for a platform that can efficiently convert image-based text into audio format. That's where our "Audio Conversion App" comes in – a solution designed to make text from images accessible and convenient for all users.

2. Scope and Objective

Motivated by this need, I have chosen to focus on developing an audio conversion app for image-based documents. Our objective is to create an application that transforms text found in images into clear, high-quality audio. This app aims to improve accessibility, provide convenience for users on the go, and offer an efficient way to consume text-based content through audio.

3. Methods

- **Observation Method**: Observing the general use cases and challenges faced by individuals dealing with text in images.
- **Theories Analyzing Method**: Researching related academic papers and analyzing current systems and technologies used in text-to-speech conversion.
- **Experimental Method**: Extracting features and functionalities from existing applications to determine essential capabilities for the audio conversion app.
- **Modeling Method**: Using insights from the above methods to build a comprehensive model, represented through diagrams and figures, to guide the development of the final application.

4. Structure of the Thesis

- Chapter 1: Problem Definition
- Chapter 2: Theoretical Foundation
- Chapter 3: System Implementation
- Chapter 4: Experimental Results and Application Demo
- Chapter 5: Conclusion

Chapter 1: PROBLEM DEFINITION

1.1. Overview

1.1.1. Context

The digital age has brought about an influx of image-based text from various sources like scanned documents, photos of printed materials, and handwritten notes. This text, while visually accessible, poses challenges for those who need or prefer auditory information.

1.1.2. Importance

Accessing text from images through audio can significantly enhance accessibility for visually impaired individuals and offer convenience for others who are constantly on the move.

1.2. Introduction to Image-To-Audio

The concept of image-to-audio conversion involves transforming text contained within images into spoken words, leveraging technologies such as Optical Character Recognition (OCR) and Text-to-Speech (TTS). This process facilitates the accessibility of textual information for users who may have visual impairments, those who prefer auditory consumption, and individuals who need to access information hands-free while multitasking.

1.2.1. Optical image recognition (OCR)

- **Functionality**: OCR technology scans images to detect and extract text, converting it into a machine-readable format. This involves identifying characters and words from various image types, including scanned documents, photographs, and handwritten notes.
- **Challenges**: OCR must accurately process text from different fonts, sizes, and orientations, and cope with varying image qualities. The effectiveness of OCR can be influenced by factors such as lighting conditions, background noise in images, and the presence of complex layouts.

1.2.2. Text to speech (TTS)

TTS technology converts written text into spoken words, allowing users to listen to the content. This involves selecting appropriate voices, ensuring clear pronunciation, and adjusting the speech rate to suit user preferences.

1.3. Problem Statement

The development and implementation of Image to audio application for Vietnamese documents face significant challenges due to the unique characteristics of the Vietnamese language. These challenges hinder the accurate extraction and conversion of text from images, thereby affecting the usability and accessibility of OCR applications for Vietnamese-speaking users. The key issues include:

1.3.1. Complex Diacritical System

Vietnamese language utilizes a wide range of diacritical marks that are essential for proper pronunciation and meaning. The presence of multiple diacritical marks on a single letter can complicate the recognition process, especially when these marks are faint, poorly printed, or handwritten.

1.3.2. Script Complexity

The Vietnamese script, although based on the Latin alphabet, includes additional characters and specific combinations unique to the language. This complexity requires OCR systems to be finely tuned and specifically trained to handle Vietnamese text accurately.

1.3.3. Diverse Font and Style Variations

Vietnamese documents are presented in various fonts and styles, including formal printed documents and informal handwritten notes. The wide range of typographic variations demands an adaptable OCR system capable of recognizing text across different formats consistently.

1.3.4. Intricate Document Layouts

Many Vietnamese documents feature complex layouts with multiple columns, mixed text orientations, and embedded images or tables. OCR systems need to accurately interpret these layouts to ensure correct text extraction without losing the document’s context.

1.3.5. Language-Specific Contextual Challenges

Vietnamese has many homonyms and context-dependent meanings, making it crucial for OCR systems to incorporate contextual understanding to accurately recognize and interpret the text. Without this, the extracted text may be incorrect or meaningless.

1.3.6. Lack of Integrated Text-to-Audio Conversion

Existing solutions typically require multiple steps and separate applications to convert Vietnamese text from images into audio. This fragmented process results in a cumbersome and inefficient user experience, deterring users from utilizing these tools effectively.

1.4. Related works

Some works have been done to implement an OCR system:

1.4.1. Convolutional Recurrent Neural Networks (CRNN)

- **Overview:** A CRNN (Convolutional Recurrent Neural Network) merges the capabilities of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), usually utilizing Long Short-Term Memory (LSTM) networks. This architecture is crafted to recognize text within images by leveraging CNNs for extracting spatial features and RNNs for capturing sequential dependencies.
- **Strengths:**
    - **End-to-End Learning:** CRNN can learn directly from raw image data without the need for manual feature extraction.
    - **Handling Sequence Data:** The combination of CNNs and RNNs allows CRNN to effectively handle sequences of characters, making it suitable for recognizing text lines or words.
    - **Flexibility:** CRNN can be trained on a wide variety of text styles and layouts, making it adaptable to different OCR tasks.
- **Drawbacks:**
    - **Complex Training:** Training CRNN models can be computationally intensive and requires large amounts of annotated data.
    - **Performance with Complex Layouts:** CRNN may struggle with documents that have complex layouts or significant variations in text orientation and size.

1.4.2. Connectionist Temporal Classification (CTC)

- **Overview:** CTC is often used in conjunction with CRNNs to handle the alignment problem between input sequences and output labels. It allows the model to predict the sequence of characters without requiring pre-segmented input data.
- **Strengths:**
    - **Alignment-Free Recognition:** CTC allows for flexible recognition of sequences without requiring explicit segmentation, which is particularly useful for handwriting recognition and continuous text recognition.
- **Drawbacks:**
    - **Training Complexity:** Training models with CTC loss can be challenging and requires careful tuning of hyperparameters.
    - **Decoding Complexity:** The decoding process can be computationally intensive, especially for long sequences.

1.5. Problem Solution

To address these issues, a Vietnamese document-based image-to-audio mobile app will be developed, leveraging advanced OCR and TTS technologies specifically tailored for the Vietnamese language. This app aims to:

- Enhance Accessibility for Vietnamese Users:

  Provide visually impaired Vietnamese users with an intuitive and reliable means to access text within images through high-quality audio conversion, enhancing their ability to engage with written content.
- Streamline Efficiency:

  Automate the process of text extraction and audio conversion, reducing the time and effort required for manual transcription of Vietnamese text from images.
- Improve Accuracy in Vietnamese Text Recognition:

  Utilize specialized OCR models trained on Vietnamese text, capable of accurately recognizing and extracting text despite the complexity of diacritical marks and unique character combinations.
- Provide a Seamless User Experience:

  Integrate the processes of text detection, recognition, and audio conversion into a single, cohesive application, offering a streamlined and user-friendly experience tailored for Vietnamese-speaking users.

Chapter 2: THEORETICAL FOUNDATION

2.1. Chapter overview

This chapter provides an overview of the key theoretical concepts and technologies that underpin the development of a Vietnamese document-based image-to-audio mobile app. It covers three main areas:

- Introduction to Text Detection and Text Recognition
- Text Detection using Differentiable Binarization (DB) Model
- Text Recognition using Transformer Model
- Building a Image to audio system
- Image to audio pipeline
- Use case diagram
- Activity diagram

2.2. Introduction to Text Detection and Text Recognition

Based on the requirements of this problem, there are two essential steps that cannot be omitted: Text Detection and Text Recognition. In practice, some modern approaches have attempted to combine these two tasks into one. However, this does not yield results that are comparable to tackling these two tasks independently:

- Text Detection:

  This step involves identifying and locating the areas within an image where text is present. Accurate text detection is crucial as it sets the foundation for the subsequent recognition process. For Vietnamese documents, this step must be capable of handling various document layouts and text orientations.

- Text Recognition

  Once the text areas are detected, the next step is to recognize and convert the text into a machine-readable format. Given the complexity of the Vietnamese language, including its diacritical marks and unique character combinations, specialized models and algorithms are required to ensure high accuracy in text recognition.

By approaching Text Detection and Text Recognition as two separate, independent tasks, the solution can leverage specialized techniques and optimizations for each step. This separation allows for more precise detection and accurate recognition of Vietnamese text, ultimately improving the performance and reliability of the OCR system.

2.3. Text Detection using Differentiable Binarization (DB):

2.3.1. Methodology

The architecture of our Differentiable Binarization is depicted in Fig. 2.1. Initially, the input image passes through a feature-pyramid backbone. Then, the pyramid features are up-sampled to a uniform scale and combined to generate a feature map F. This feature map F is subsequently used to predict both the probability map P and the

threshold map T. Following this, the approximate binary map B is derived using P and F.

During the training phase, supervision is applied to the probability map, the threshold map, and the approximate binary map, with the probability map and the approximate binary map receiving identical supervision. In the inference phase, bounding boxes can be effortlessly extracted from either the approximate binary map or the probability map through a box formulation module.

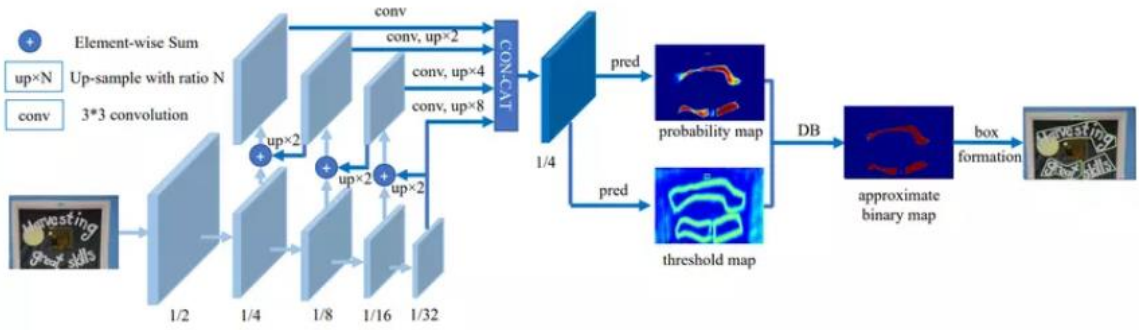


Figure 2.1. DB Architecture

2.3.2. Standard Binarization

Given a probability map P of dimensions R^{HxW} produced by a segmentation network, where R, H, and W represent the number of channels, height, and width of the map, respectively, it is essential to convert it into a binary map P_{binary} of the same dimensions. In this binary map, pixels with a value of 1 are considered valid text areas. This binarization process can be described as follows:

$$|B_{i,j} = \begin{cases} 1 & \text{if } P_{i,j} >= t \\ 0 & \text{otherwise.} \end{cases}$$

where t is the predefined threshold, and (i,j) indicates the coordinate point in the map.

2.3.3. Diffentiable Binarization

Standard binarization has a drawback: it is challenging to find an appropriate threshold value t. To address this issue, we use a method called Differentiable Binarization (DB), which allows for the computation of derivatives, thereby integrating seamlessly into the training process.

The approximate binary map $B_{i,j}$ is defined as:

$$\hat{B}_{i,j} = \frac{1}{1 + e^{-k(P_{i,j} - T_{i,j})}}$$

where B^ is the approximate binary map, T is the adaptive threshold map learned by the model, and k is the amplification factor, typically set to 50. This method uses an adaptive threshold, which not only helps in distinguishing text from the background but also aids in separating connected text regions.

The improvement in performance through DB can be explained by the backpropagation process. We define the DB function $f(x) = \frac{1}{1+e^{-kx}}$, where $x = P_{i,j} - T_{i,j}$. The positive label loss l^+ and the negative label loss l^- are expressed as follows:

$$l^+ = -\log\left(\frac{1}{1+e^{-kx}}\right)$$

$$l^- = -\log\left(1 - \frac{1}{1+e^{-kx}}\right)$$

The derivatives of these loss functions, calculated using the chain rule, are:

$$\frac{\partial l^+}{\partial x} = -kf(x)e^{-kx}$$

$$\frac{\partial l^-}{\partial x} = kf(x)$$

The graphs of the loss functions and their derivatives are shown below.

Thanks to the parameter k, the model can optimize predictions and more clearly distinguish between text and background regions.

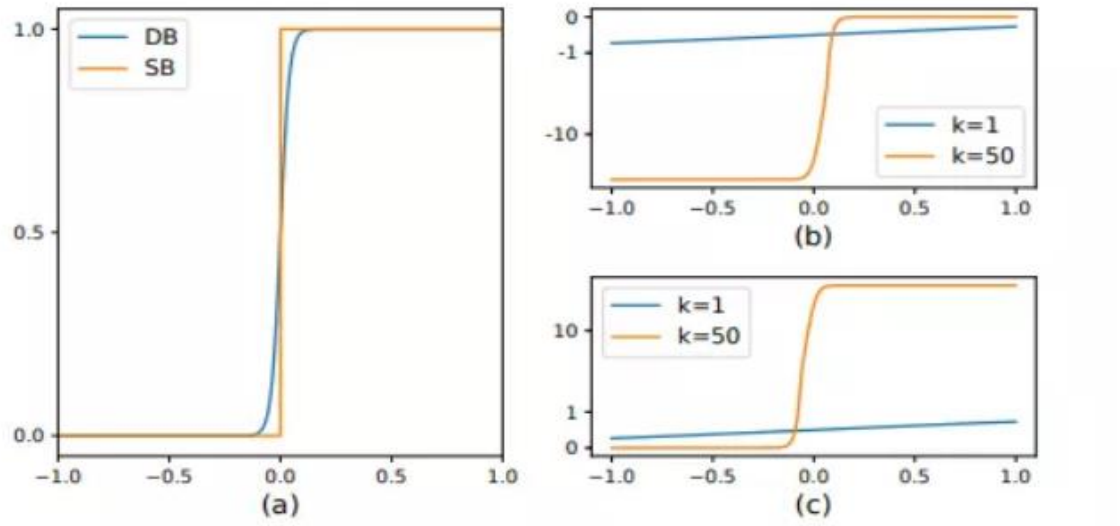


Figure 4: Illustration of differentiable binarization and its derivative. (a) Numerical comparison of standard binarization (SB) and differentiable binarization (DB). (b) Derivative of l_+. (c) Derivative of l_-.

Figure 2.2 Graph of DB loss function

2.3.4. Optimization

The loss function for the model is the weighted sum of three loss functions: the probability map loss L_s the binary map loss L_b and the threshold map loss L_t:

$$L = L_s + \alpha \times L_b + \beta \times L_t$$

The value of α is set to 1.0 and β to 10.

The loss function used here is Binary Cross-Entropy (BCE) loss for both L_s and L_b. To address the imbalance between positive and negative pixels (since non-text pixels typically outnumber text pixels in an image), the authors employ hard negative mining in the BCE loss by selecting hard negative samples (i.e., those negative samples that the model finds difficult to classify) instead of using all negative samples.

$$L_s = L_b = \sum_{i \in S_l} y_i \log x_i + (1 - y_i) \log(1 - x_i)$$

where S_l is the sampling set with a positive to negative ratio of 1:3. The threshold map loss L_t is calculated as the total L_1 distance between the predictions and the labels:

$$L_t = \sum_{i \in R_d} |y_i^* - x_i^*|$$

where R_d is the set of points within the text region, and y_i^* is the label for the threshold map.

2.3. Text Recognition using Transformer Model

2.3.1. Overview

Like other machine translation models, the transformer model's overall architecture comprises two main components: the encoder and the decoder. The encoder learns the vector representation of a sentence, aiming to encapsulate the complete information of that sentence. The decoder then converts this representation vector into the target language. What sets the transformer model apart from traditional sequential models are two key structures: Multi-Head Attention and Positional Encoding.

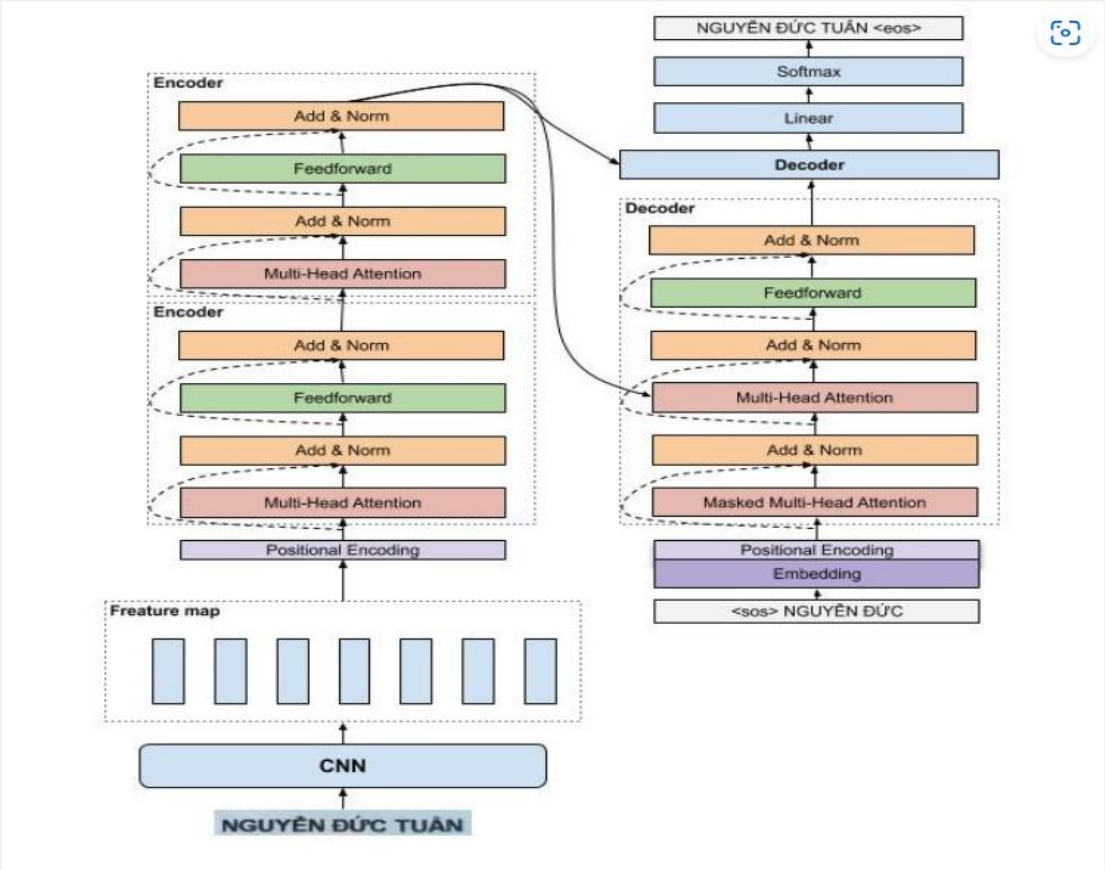


Figure 2.3. Transformer model

2.3.2. Encoder

The encoder of the transformer model consists of multiple identical encoder layers. Each encoder layer includes two main components: multi-head attention and a feedforward network, along with skip connections and normalization layers.

The first encoder layer receives a matrix representation of the words, augmented with positional information through positional encoding. This matrix is then processed by the multi-head attention mechanism. Essentially, multi-head attention is an extension of self-attention, enabling the model to focus on various patterns by utilizing multiple self-attention mechanisms simultaneously.

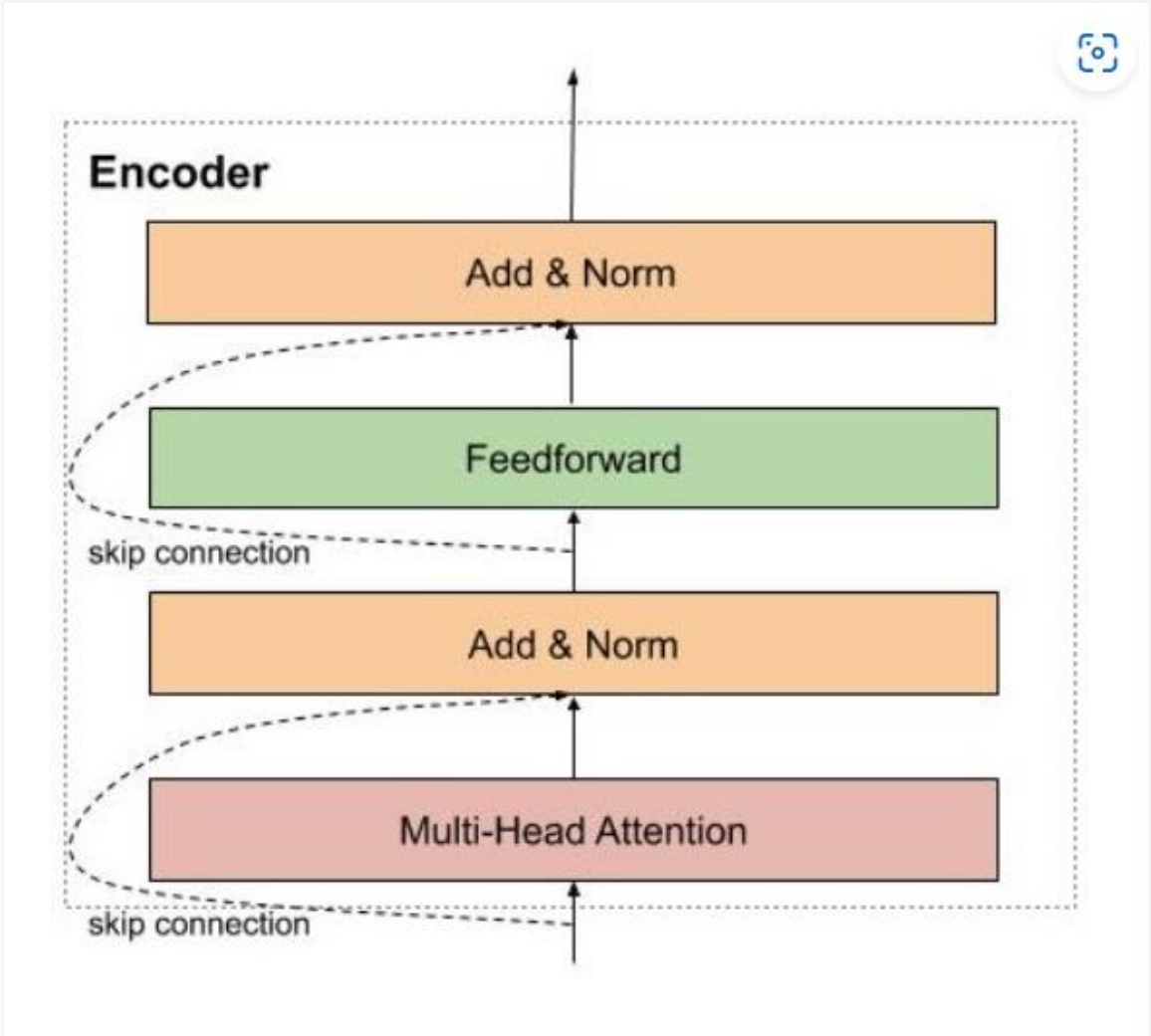


Figure 2.4. Transformer model encoder

2.3.3. Multi-head attention

We want the model to learn various types of relationships between words. With each self-attention mechanism, we learn one type of pattern. Therefore, to expand this capability, we simply add more self-attention mechanisms. This means we need multiple query, key, and value matrices. As a result, the key, query, and value weight matrices will have an additional depth dimension.

Multi-Head Attention allows the model to simultaneously focus on different observable patterns, such as:

- Attention to the preceding word of a given word
- Attention to the following word of a given word
- Attention to related words of a given word

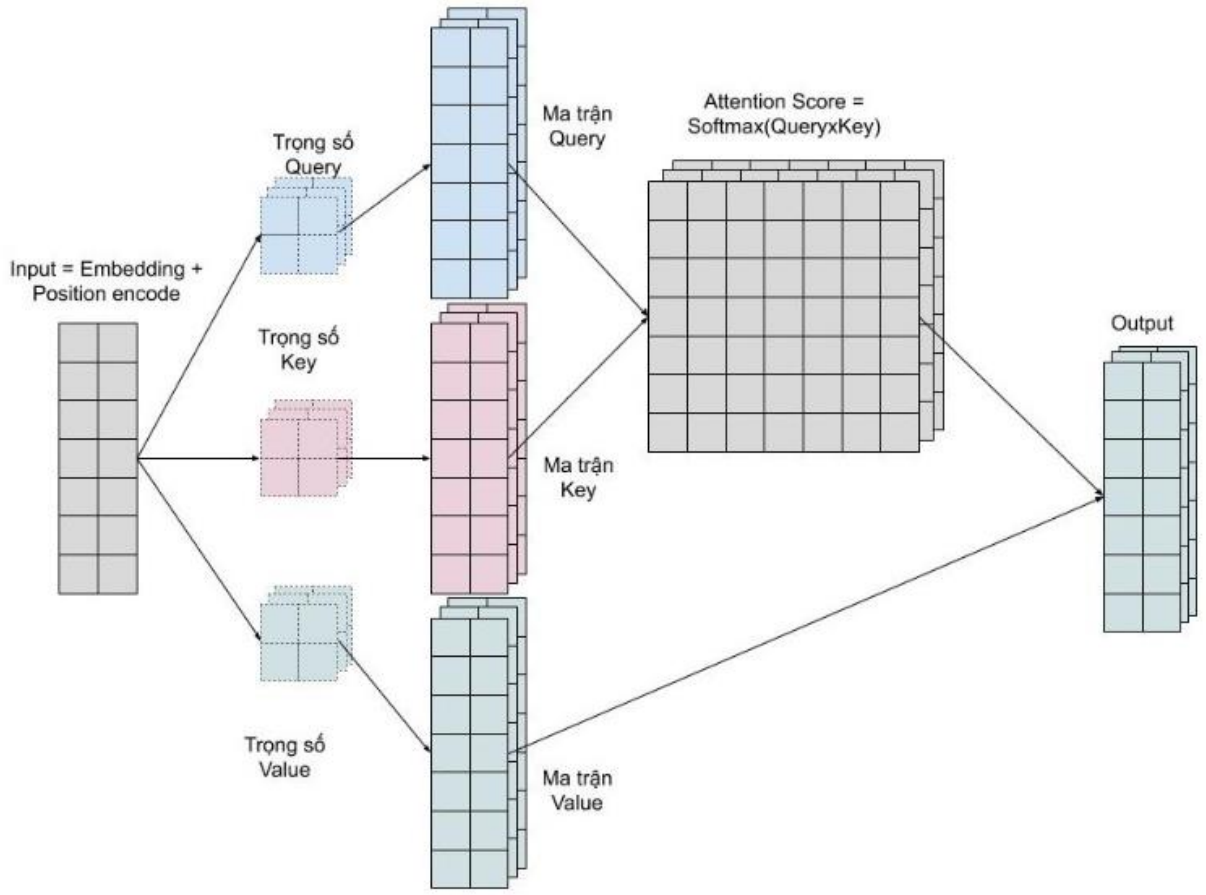


Figure 2.5. Multi-head Attention

2.3.4. Decoder

The decoder's role is to transform the source sentence vector into the target sentence. To do this, the decoder receives two vectors from the encoder: the key and the value. The architecture of the decoder closely resembles that of the encoder, with one key difference: an additional multi-head attention layer in the middle. This extra layer is designed to learn the relationships between the word being translated and the words in the source sentence.

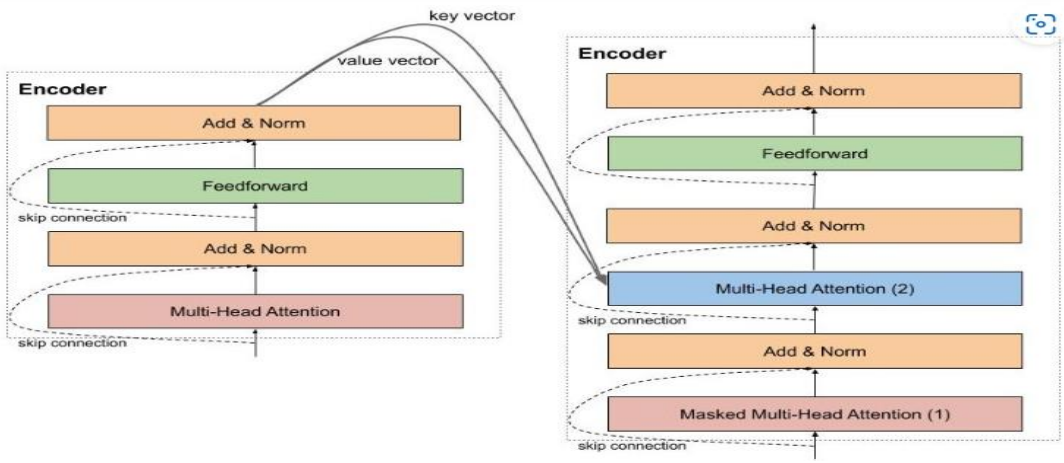


Figure 2.6. Decoder

The Masked Multi-Head Attention, as discussed earlier, functions to encode the words of the target sentence during the translation process. However, during implementation, it is crucial to mask out future words that the model has not yet translated. This can be done by multiplying with a vector containing 0s and 1s.

Additionally, the decoder includes another multi-head attention layer that focuses on the words from the encoder. This layer receives the key and value vectors from the encoder, along with the output from the previous decoder layer. This design allows the model to compare the correlation between the word being translated and the words in the source sentence effectively.

2.3.5. Positional Encoding

The position and order of words in a sentence are essential for any language model, whether in NLP or CV. Models like RNNs or LSTMs use sequential processing to learn the positions of words in a text. However, as mentioned earlier, to overcome the long training times caused by sequential processing, the Transformer model eliminates this entirely. So how does the model learn positional information? By encoding positional information into each word of the sentence, a technique known as Positional Encoding.

A good positional encoding meets the following criteria:

Each time-step must have a unique encoding: If two different time-steps have the same encoding, it will cause confusion regarding word positions.

The distance between the embeddings of two positions in sentences of different lengths must be consistent.

It should be able to represent positions for sentences longer than those seen during training.

The Transformer achieves all these expectations with its positional encoding method. The formula proposed by the authors in the paper is as follows:

$$PE[pos, 2i] = \sin(pos/10000^{2i/d_{model}})$$

$$PE[pos, 2i+1] = \cos(pos/10000^{2i/d_{model}})$$

Where pos is the current position, d_{model} is the fixed size of the model, and i is the dimension.

To explain further, let's define $w = \frac{1}{10000^{2i/d_{model}}}$:

$$PE[pos, 2i] = \sin(w \cdot pos)$$

$$PE[pos, 2i+1] = \cos(w \cdot pos)$$

Now, let's evaluate this encoding method against the criteria mentioned above:

- **Each time-step must have a unique encoding**: As i increases, w decreases gradually, approaching 0. Thus, each position i will have a different representation due to different pos values. Although sine and cosine functions are periodic, the non-fixed decreasing w ensures different values for each i.

Furthermore, identical pos values in sentences of different lengths will still have unique embeddings.

- **The distance between embeddings of positions in sentences of different lengths must be consistent**: Since each position i has a different embedding in the same sentence and the same embedding in sentences of different lengths, the distance between them remains constant regardless of sentence length.
- **Ability to represent positions for sentences longer than those seen during training**: Since sine and cosine are periodic functions with a period of $2k\pi$, we can easily compute w and determine pos values, enabling the representation of far positions even beyond training.

The following illustration shows positional encoding:

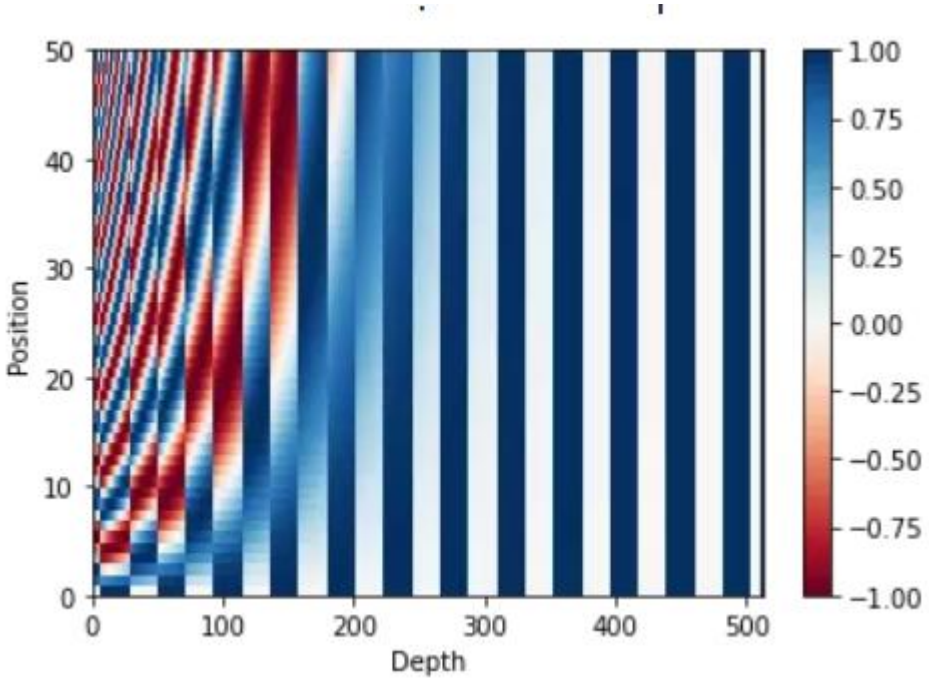


Figure 2.7. Positional Encoding

As i gets deeper and larger, the sine and cosine values approach 0 and 1, making the embedding positions more similar. Consequently, the columns become more uniform in color, showing less variation.

In conclusion, the Transformer's positional encoding method effectively meets all the criteria for a robust encoding, ensuring unique and consistent positional representations even for longer sentences.

2.4. Building an Image to audio system:

To achieve this, I propose 2 steps:

First step: Collecting datasets for text images, this could be achieved by browsing the Internet or using available famous OCR datasets like MNIST (Modified National Institute of Standards and Technology database), IAM Handwriting Database, COCO-Text, SynthText… Since this project main target is Vietnamese, we should collect pictures of Vietnamese text with diverse fonts and layouts. For text recognition task, we can cut the text from the available dataset for better inferences. After that, we

will start training using DB for Text Detection and TransformerOCR for text recognition.

Second step: The input of the system will be an image. First, it will pass through the text detection model to detect the text areas. After that, the image will be cropped into smaller images which only contain the text so that the text recognition model will be able to decode. The result will then be reorganized into a coherence texts, keeping the order from left to right according to the original input. Finally, we will use a text-to-speech API to get the audio from the decrypted text.

2.5. Image to audio Pipeline

The main pipeline of our Image to audio system:

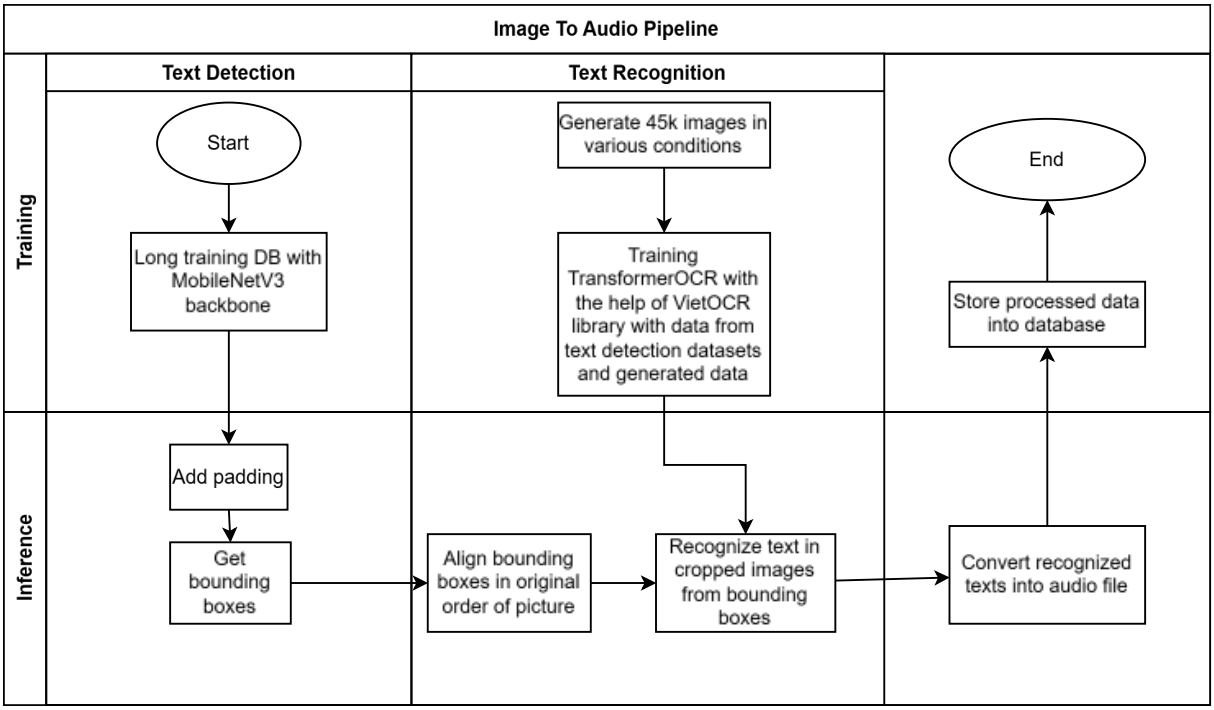


Figure 2.8 System pipeline

2.6. Use case diagram

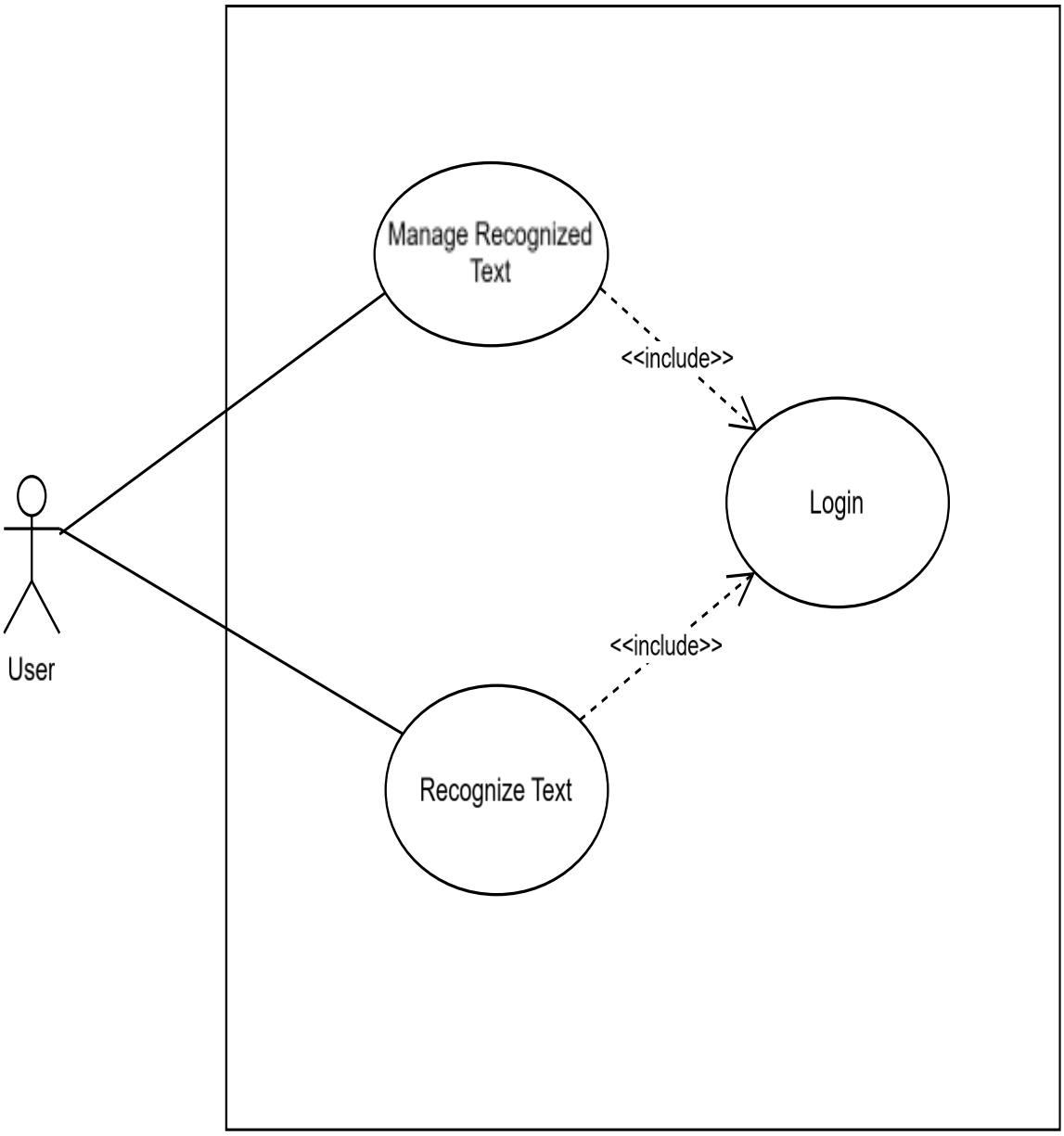


Figure 2.9 General use case

2.6.1. Authentication

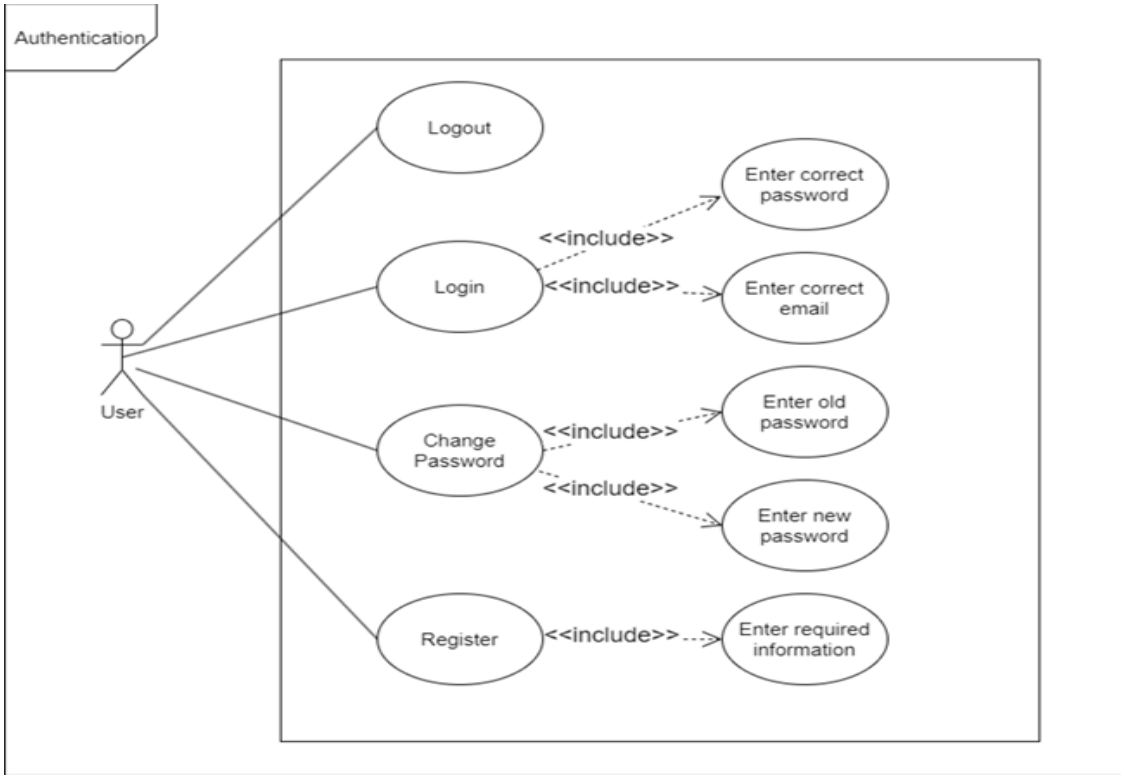


Figure 2.10 Authentication

2.6.2. Recognize Text

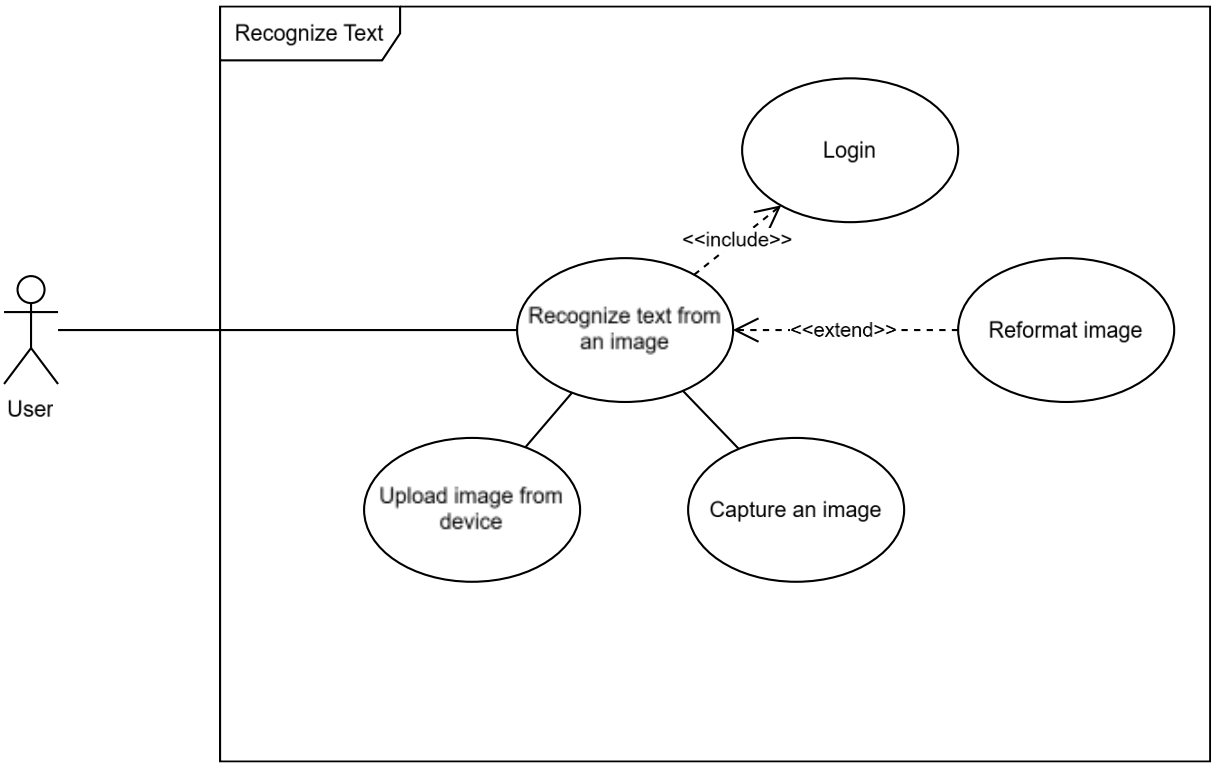


Figure 2.11 Recognize Text

2.6.3. Manage recognized texts

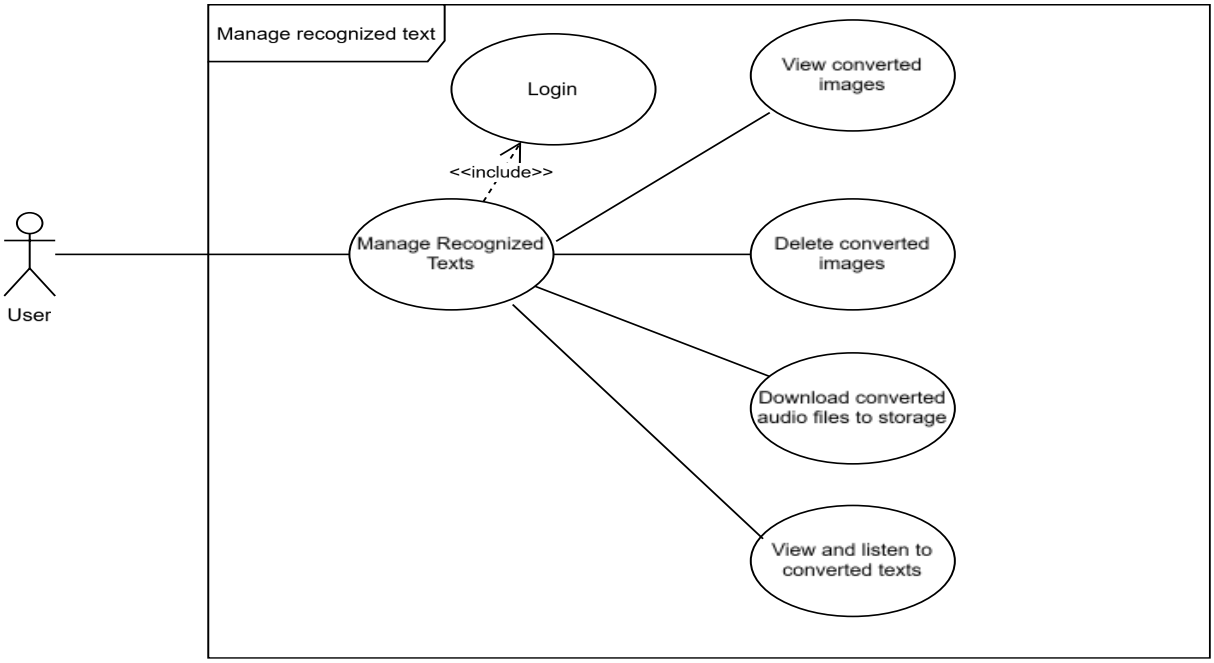


Figure 2.12 Mange recognized texts

2.7. Activity diagram

2.7.1. Login activity diagram

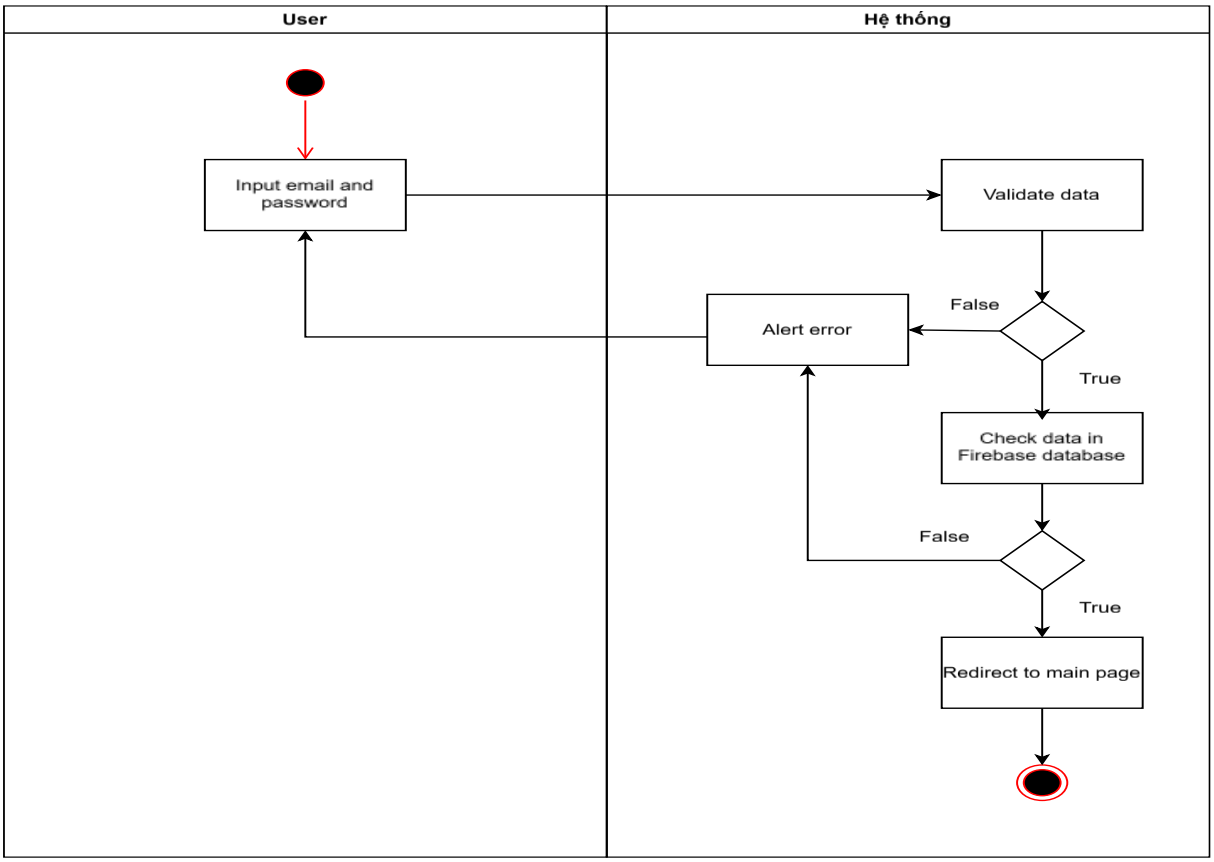


Figure 2.13 Login activity diagram

2.7.2 Logout activity diagram

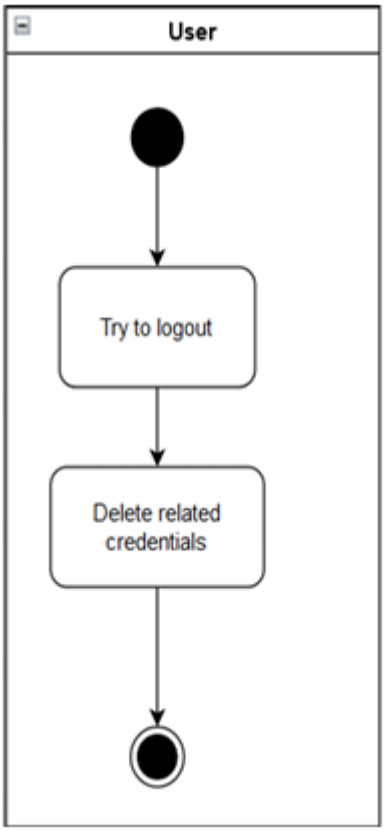


Figure 2.14 Logout activity diagram

2.7.3. Process image activity diagram

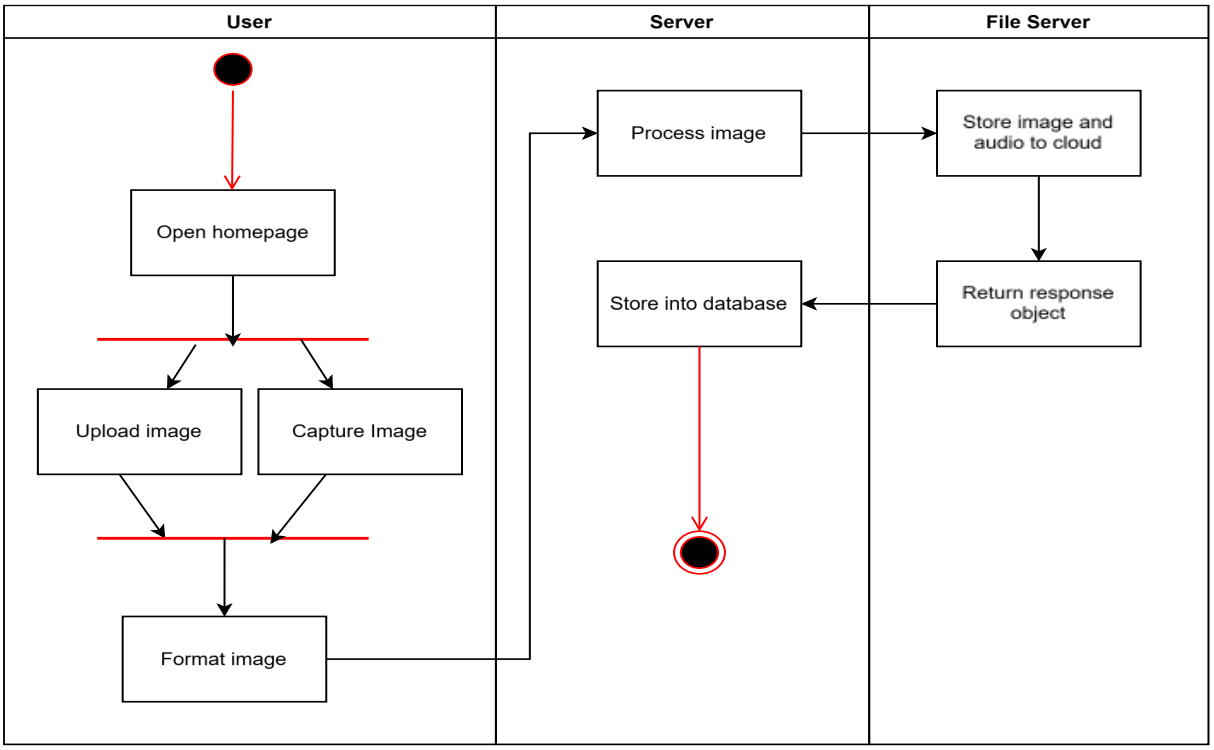


Figure 2.15 Process image activity diagram

2.7.4. View processed images activity diagram

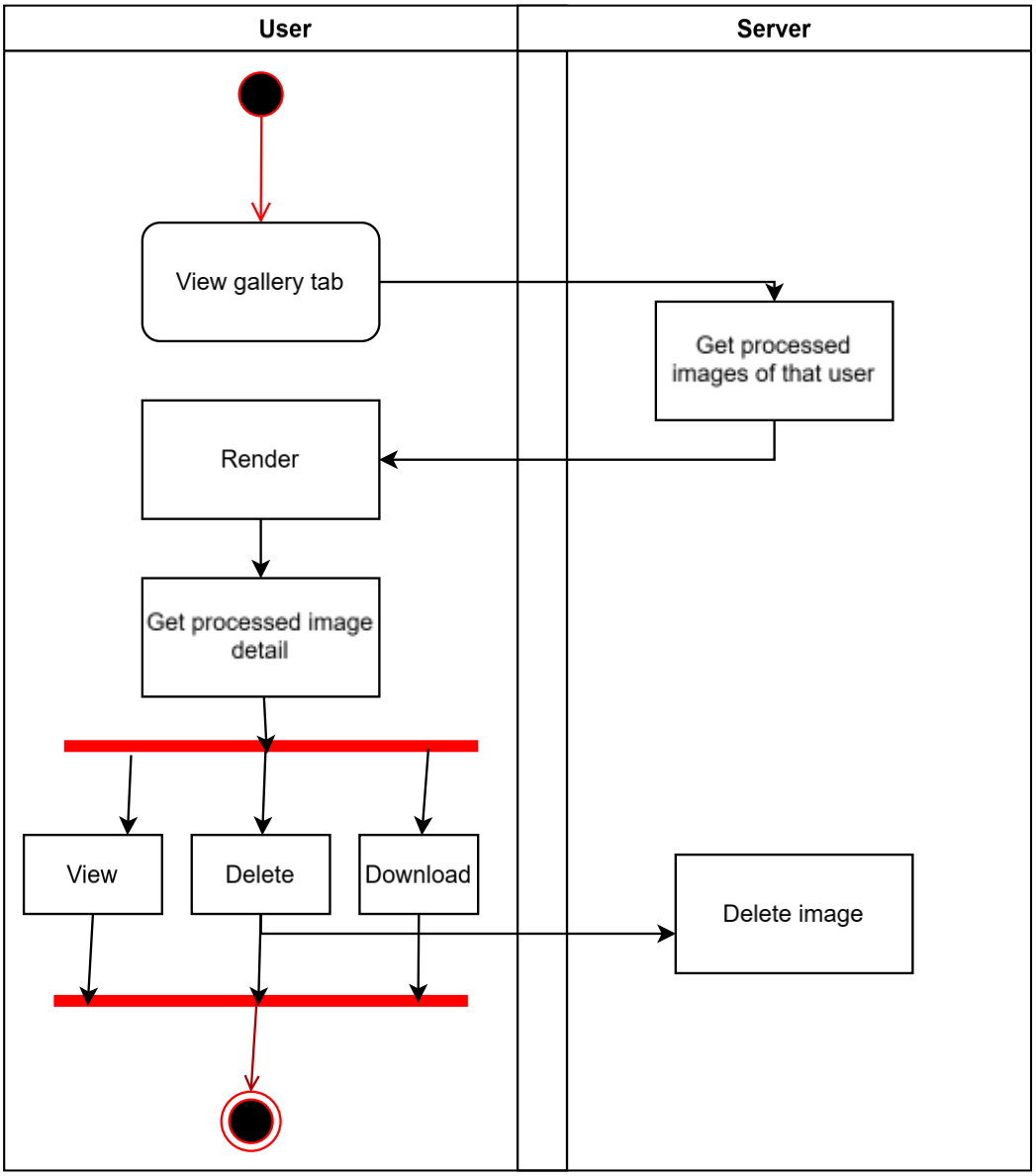


Figure 2.16 View processed images activity diagram

Chapter 3: SYSTEM IMPLEMENTATION

3.1. System Architecture

3.1.1. Overall architecture

The overall architecture of the application is presented as below:

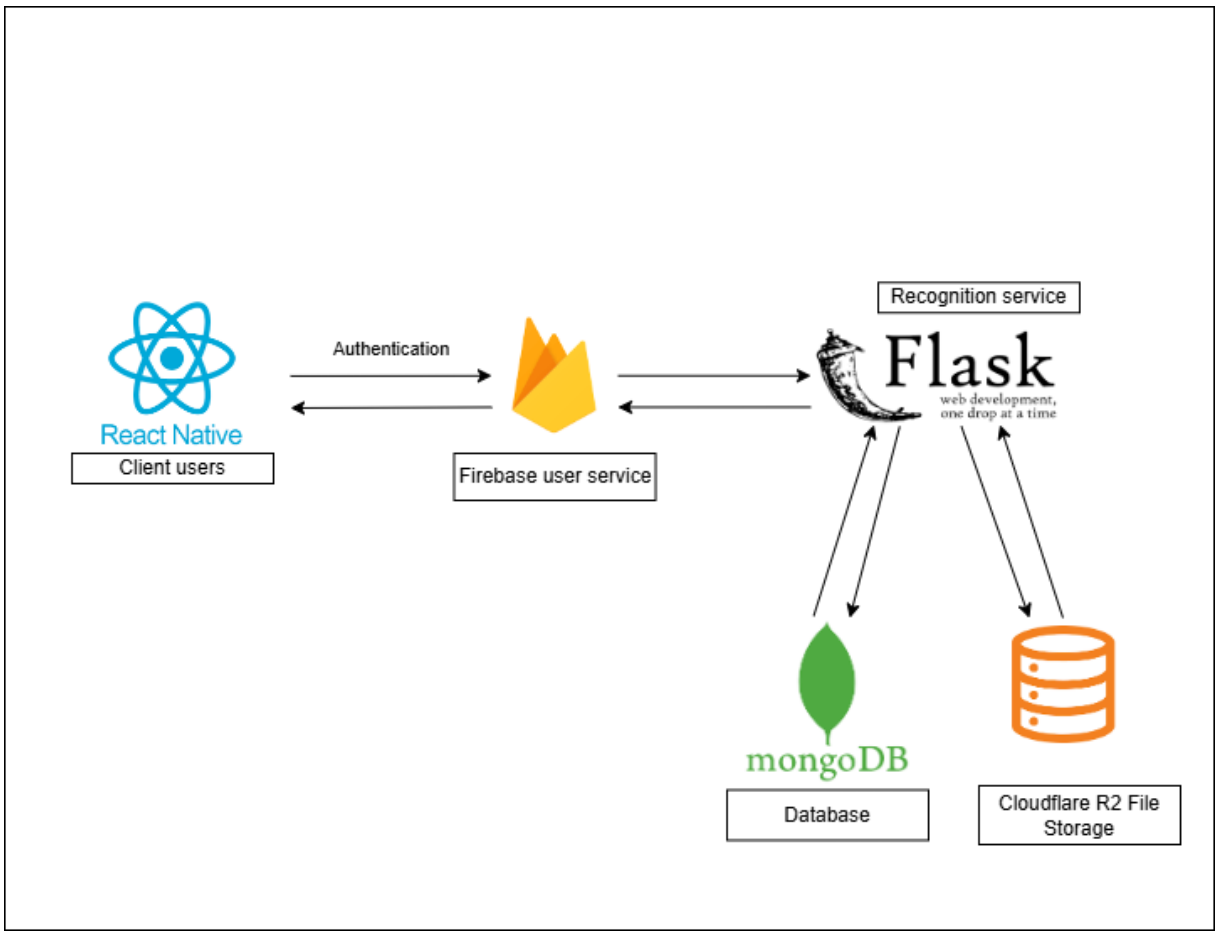


Figure 3.1 Application architecture

At first it can visualize the overview of our system so that we can deep down into how we implemented this. As we may see, our main system consists of a React Native client app, a server which contains our OCR model and a Firebase backend for authentication purpose. Besides, we also serve images and audio files through CDN which belongs to an external cloud service – Cloudflare R2.

3.2. Backend technology stack:

3.2.1. Python

Before we start implementing our system out of the paper. We need to decide which language that can help us develop the product as fast as we need but still keep the

best structure for it. And so on Python - is our choice for both client and backend languages.

Python is a versatile and powerful programming language created by Guido van Rossum and first released in 1991. It is known for its simplicity and readability, making it an excellent choice for beginners and experienced developers alike. Python is widely used for a variety of applications, ranging from web development and data science to machine learning, automation, and even game development.

Python combines the elegance of object-oriented programming with the flexibility of a modern programming language. It supports a wide range of programming paradigms, including object-oriented programming, functional programming, and procedural programming. Python offers features such as automatic memory management through garbage collection, dynamic typing, and a robust exception handling mechanism to ensure reliable and secure application development.

3.2.2. Flask Framework

To host our application with the main features of taking image from a Http request and pass it to our machine learning model with Python, the most optimal choice would be Flask.

Flask is a lightweight and flexible web framework for Python, developed by Armin Ronacher. It is part of the Pallets Projects and was first released in 2010. Flask is designed to make getting started quick and easy, with the ability to scale up to complex applications. It is widely used for building web applications, from simple prototypes to full-featured websites and services.

Flask combines the power of Python with a simple and straightforward API, making it an excellent choice for developers who want to create web applications with minimal setup and configuration. Flask follows the WSGI (Web Server Gateway Interface) specification and is based on the Werkzeug WSGI toolkit and Jinja2 template engine.

3.2.3. Firebase Backend Authentication Service

Our application is a mobile app. So for the authentication of the system, I will use Firebase Authentication as the way for user to save their previous uses in the system.

Firebase Authentication is a robust and flexible service provided by Google Firebase, designed to simplify the process of user authentication and identity management for developers. It offers a variety of methods to authenticate users, including email and password authentication, phone number authentication, and federated identity providers like Google, Facebook, and Twitter.

Firebase Authentication integrates seamlessly with other Firebase services and is widely used in mobile and web applications to handle user authentication with minimal effort and maximum security. It supports client-side SDKs for multiple platforms, including iOS, Android, and web, as well as server-side SDKs for languages such as Node.js, Java, Python, and more.

3.3. Building Client UI:

3.3.1. React Native

React Native is an open-source framework developed by Facebook for building mobile applications using JavaScript and React. It enables developers to create native mobile apps for both iOS and Android platforms using a single codebase, leveraging React, a popular JavaScript library for building user interfaces. Since its launch in 2015, React Native has rapidly become one of the most widely used frameworks for mobile app development due to its efficiency, performance, and ability to reuse code across platforms.

React Native combines JavaScript and JSX, a special markup language similar to XML. The framework facilitates communication between JavaScript-based threads and native app threads through a mechanism known as the "bridge." Despite being written in entirely different languages, JavaScript and native threads can interact bidirectionally thanks to this bridge feature.

Here's a great visualization of the bridge concept:

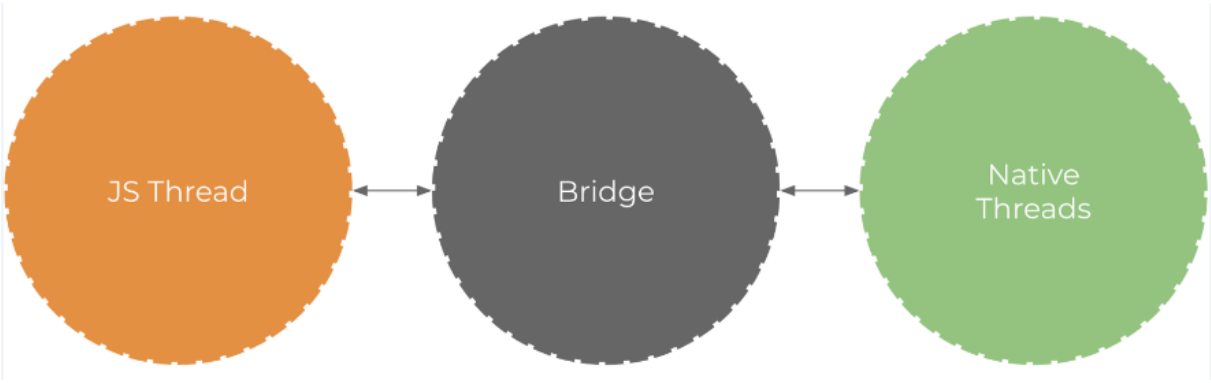


Figure 3.2 React Native Bridge

This means that if you already have a native iOS or Android app, you can still utilize its components or transition to React Native development. Moreover, React Native allows developers to create platform-specific versions of their apps for different mobile platforms, thereby enhancing the app's performance and user experience on each device.

For development purpose, for this application I will implement the React Native on Expo Go – a React Native development tool.

3.3.2. Expo Go

Expo Go is a development tool provided by Expo, designed to simplify the process of developing, testing, and deploying React Native applications. Expo Go allows developers to preview their apps on physical devices without needing to build a native binary each time. This makes the development process much faster and more convenient, especially when working on features that require frequent testing.

Expo Go is available for both iOS and Android, and it leverages the capabilities of the Expo platform to provide a streamlined and efficient development workflow.

3.4. Client-server communication protocol

3.4.1. Http Protocol

HTTP, or HyperText Transfer Protocol, is a protocol that sets the rules for transmitting various types of files such as images, text, audio, video, and more over the World Wide Web (WWW). It operates on a client-server request-response model. HTTP is an application layer protocol that typically communicates with the server using the Transmission Control Protocol (TCP). Being a stateless protocol, HTTP handles each request independently, without retaining any information about previous requests, which means the connection between the browser and the server is terminated once the transaction ends. HTTP utilizes methods to inform the server of the actions to be performed when the client sends a request. The most widely used HTTP methods are GET, POST, PUT, PATCH, and DELETE.

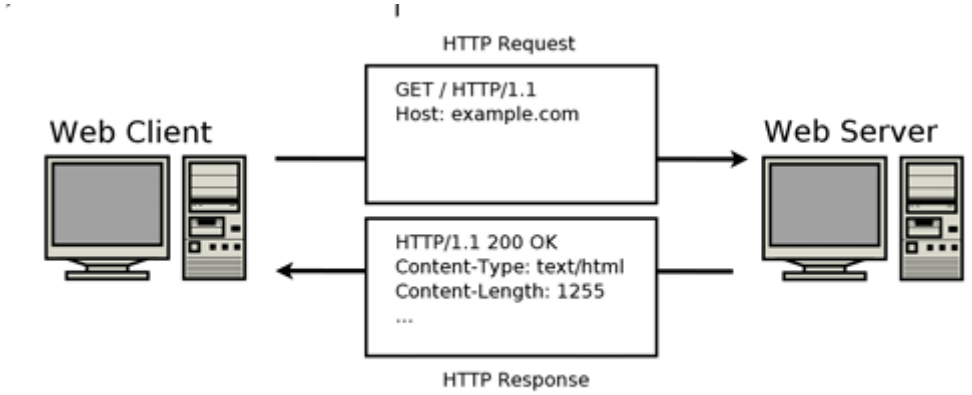


Figure 3.3 Http Protocol

3.5. Database

3.5.1. MongoDB

a. What is MongoDB

MongoDB, the most popular NoSQL database, is a schema-less database, which means the database can manage data without the need for a blueprint. Document, data in MongoDB is stored in documents with key-value pairs instead of rows and columns.

b. Why choose this

- MongoDB supports rich querying capabilities, including filtering, sorting, and aggregation, which is suitable for our read-heavy service.
- MongoDB is designed for horizontal scalability, allowing you to distribute data across multiple servers or clusters. This enables seamless scaling as your application grows, providing better performance and handling higher volumes of data and traffic.
- MongoDB is designed for store complex hierarchical data structures, nested arrays, which simplifies data representation and retrieval.

c. Document structure

In this application, image after processing will be stored into database in the following format:


```
_id: ObjectId('6668726789aaad530b64ba0b')
imageUrl : "https://pub-4745ad8a30b144ec808391a73f00b617.r2.dev/a782d0fd-0d87-4123…"
deviceId : "ed736e25-40b4-47a8-99d0-54a2b8fcea80"
audioUrl : "https://pub-4745ad8a30b144ec808391a73f00b617.r2.dev/8a58b542-b48c-4248…"
transcript : ""Tác giả Nhật dược yêu thích nhất tại Mỹ này có thể xuất bản ẩn danh t…"
name : "2024-06-11 15:50"
timestamp : 2024-06-11T15:50:36.417+00:00
imageFileKey : "a782d0fd-0d87-4123-b263-ad296abca465.jpg"
audioFileKey : "8a58b542-b48c-4248-872c-354508b7e387.mp3"
userId : "L9c8H3wh9xYR52rYaMz9G6jOo9a2"
```


Figure 3.4 Document structure

Field	Data type	Nullable	Default	Description
_id	ObjectId	False		Id of the record
imageUrl	String	False		Image url of the input image
deviceId	String	True		Id of the device the user input from
audioUrl	String	False		Audio url of the text after converting
transcript	String	True	Null	Transcript of the image
timestamp	DateTime	False	Now()	The time the image is processed
imageFileKey	String	False		Image file name
audioFileKet	String	False		Audio file name
userId	String	False		User Id of the authenticated user

Table 3.1. Description of each field in the document

3.6. Training method

3.6.1. Kaggle Notebook

Kaggle Notebooks, formerly known as Kaggle Kernels, are an integral part of the Kaggle platform, offering a powerful and interactive environment for data analysis, machine learning, and collaborative data science. Kaggle, a subsidiary of Google, is

widely recognized for its data science competitions, datasets, and educational resources. Kaggle Notebooks are hosted Jupyter Notebooks that provide a seamless way to write and execute code in languages like Python and R directly in the browser. It also provides user with up to 2 GPU-based accelerators, which improve the speed of the training process and allow distributed training, saving time considerably.

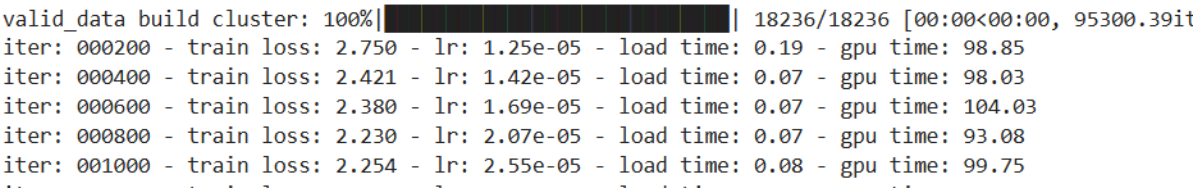


Figure 3.5 Training using Kaggle Notebook

3.7. Text Detection Training

3.7.1 Dataset

The dataset used for this task comprises 2500 meticulously curated images containing scene text in a variety of formats and layouts. These images are specifically selected to represent a broad spectrum of real-world scenarios where text appears, ensuring the dataset's comprehensiveness and utility in developing robust text detection models. The primary language featured in this dataset is Vietnamese, aligning with the core objective of the module, which is to enhance the detection and recognition of Vietnamese text in various contexts.

Dataset Composition

- **Diverse Sources**: The dataset includes images from a wide range of sources to capture the diversity of text appearances in everyday life. This includes printed text from books, newspapers, and magazines; handwritten notes and letters; digital text from screenshots of webpages and social media posts; and environmental text from billboards, shop signs, street signs, and advertisements. This variety ensures that the dataset covers different fonts, sizes, colors, and orientations of text, as well as various background complexities.
- **High-Quality Annotations**: Each image in the dataset is meticulously annotated with bounding boxes that accurately enclose the text regions. These annotations were manually created and cross-verified by native Vietnamese speakers to ensure precision. The bounding boxes are provided in a standardized format (such as JSON or XML), detailing the coordinates for each text region within the images. This careful annotation process ensures that the dataset is well-suited for training machine learning models for text detection.

To facilitate the effective training and validation of the text detection module, the dataset has been strategically divided:

- **Training Set**: 2200 images are designated for training. This subset encompasses a wide range of text types and scenarios to enable the model to learn the diverse characteristics and variations of Vietnamese text. By exposing the model to a large and varied training set, we aim to enhance its ability to generalize across different text appearances and contexts.

- **Validation Set**: 300 images are reserved for validation. This subset is used to evaluate the model's performance and ensure that it is accurately detecting and localizing text in unseen images. The validation set is carefully chosen to represent the same diversity as the training set, providing a reliable measure of the model's efficacy and helping to fine-tune its parameters.

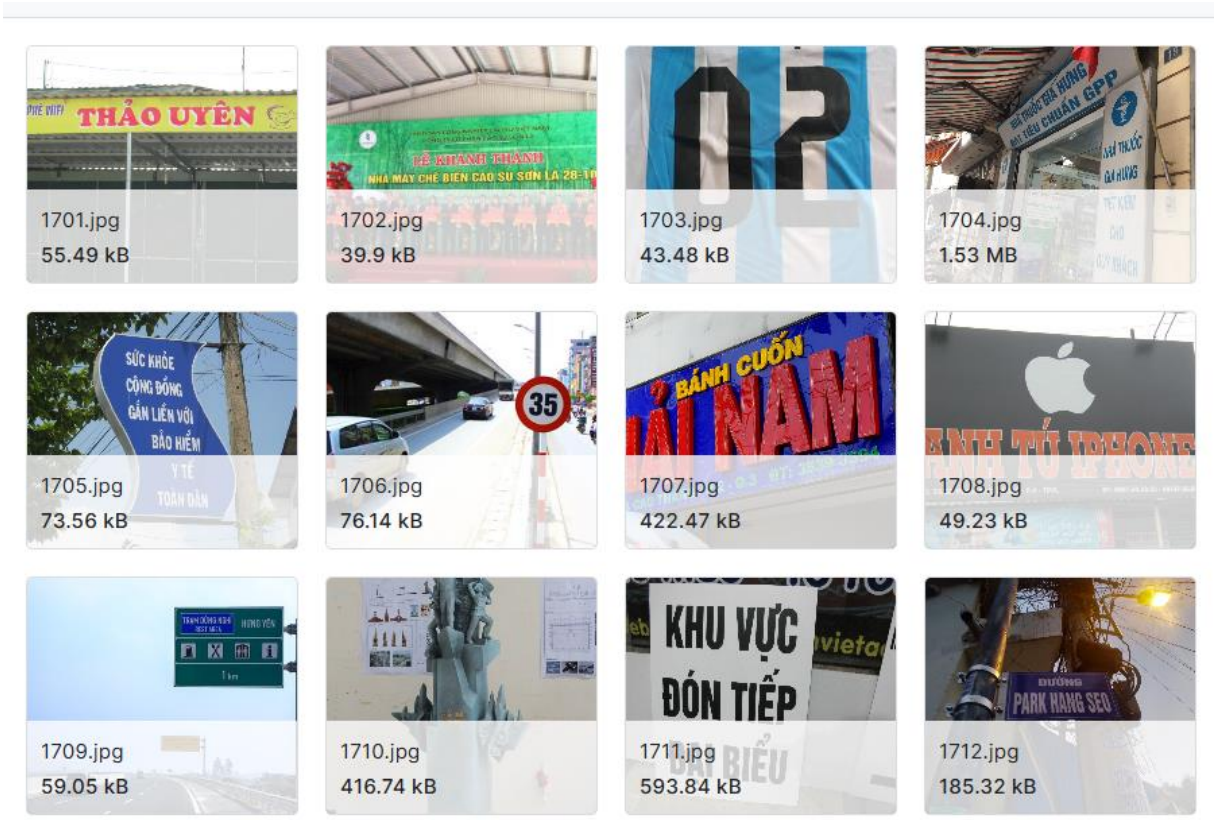


Figure 3.6 Text Detection Dataset

```
1003.jpg        [{"transcription": "PETRO", "points": [[207, 82], [285, 81], [277, 95], [200, 97]]},
1004.jpg        [{"transcription": "CHÙA", "points": [[275, 101], [445, 93], [448, 157], [287, 153]]}
1005.jpg        [{"transcription": "Lồng", "points": [[432, 126], [672, 107], [680, 273], [437, 278]]
1006.jpg        [{"transcription": "IU", "points": [[548, 436], [611, 426], [624, 468], [566, 475]]},
1007.jpg        [{"transcription": "Kim", "points": [[188, 122], [750, 170], [765, 393], [239, 510]]}
1008.jpg        [{"transcription": "KHU", "points": [[1746, 1517], [1889, 1521], [1888, 1594], [1747,
1009.jpg        [{"transcription": "AN", "points": [[288, 9], [337, 8], [337, 36], [279, 36]]}, {"tra
101.jpg [{"transcription": "###", "points": [[14, 16], [105, 7], [97, 52], [0, 55]]}, {"transcription
1010.jpg        [{"transcription": "CÚC", "points": [[62, 1], [128, 2], [126, 128], [63, 100]]}, {"tr
1011.jpg        [{"transcription": "CẢM", "points": [[311, 90], [425, 89], [427, 143], [311, 143]]},
1012.jpg        [{"transcription": "CGV", "points": [[997, 178], [1344, 137], [1291, 322], [1004, 370
1013.jpg        [{"transcription": "PHÒNG", "points": [[343, 84], [495, 113], [494, 187], [343, 160]]
1014.jpg        [{"transcription": "CUNG", "points": [[171, 177], [292, 172], [291, 206], [165, 217]]
1015.jpg        [{"transcription": "WC", "points": [[199, 68], [277, 62], [276, 83], [204, 89]]}, {"t
1016.jpg        [{"transcription": "PHÒNG", "points": [[1046, 194], [1378, 198], [1397, 332], [1059,
1017.jpg        [{"transcription": "CHÀO", "points": [[0, 141], [50, 100], [71, 129], [1, 177]]}, {"t
1018.jpg        [{"transcription": "PHỐ", "points": [[294, 64], [372, 64], [372, 98], [294, 95]]}, {"
1019.jpg        [{"transcription": "ĐÈ", "points": [[406, 230], [455, 230], [459, 331], [409, 330]]},
102.jpg [{"transcription": "Hello", "points": [[75, 127], [306, 127], [306, 209], [75, 209]]}, {"tran
```


Figure 3.7 Dataset label format

3.7.2. Training text detection with PaddleOCR library

PaddleOCR is an open-source Optical Character Recognition (OCR) tool developed by Baidu's PaddlePaddle team. It provides comprehensive support for detecting and recognizing text in various languages, including Chinese, English, and multiple other languages. PaddleOCR is built on the PaddlePaddle deep learning framework and is designed to offer high accuracy, efficiency, and ease of use for OCR applications.

Key features of PaddleOCR:

- **Multilingual Support**: PaddleOCR supports text recognition in over 80 languages, making it versatile for global applications.
- **High Accuracy**: Leveraging state-of-the-art deep learning models, PaddleOCR delivers high accuracy in text detection and recognition.
- **Comprehensive Pipeline**: It provides an end-to-end OCR pipeline, including text detection, recognition, and post-processing.
- **Lightweight and Efficient**: PaddleOCR is optimized for performance, offering lightweight models suitable for deployment on various devices, including mobile and edge devices.
- **Extensibility**: Users can customize and extend PaddleOCR to meet specific requirements by modifying the underlying models or integrating additional functionalities.

PaddleOCR offers pretrained models based on the DB architecture with MobileNetv3 as the backbone. To enhance its performance specifically for Vietnamese language tasks, I has fine-tuned these models using the dataset above. This approach aims to optimize the model's inference capabilities and accuracy in recognizing Vietnamese text across various contexts and formats.

3.8. Text Recognition

3.8.1 Dataset

This extensive dataset comprises 46,200 images, integrating 45,000 auto-generated text images with cut images processed from the text detection dataset.

Training Set

A total of 36,960 images are allocated for training, following an 8:2 split ratio of the 46,200 images dataset. This training set includes a balanced mix of auto-generated images and real-world scene text images, providing a comprehensive foundation for training the text recognition model. By exposing the model to a diverse array of text appearances, formats, and contexts, the training set aims to enhance the model's robustness and generalization capabilities, enabling it to effectively recognize and transcribe text across various scenarios.

Validation Set

The remaining 9,240 images are reserved for validation. This subset is used to evaluate the model's performance on unseen data, ensuring that it can accurately recognize and transcribe text in a wide range of conditions. The validation set is carefully selected to reflect the same diversity present in the training set, providing a reliable measure of the model's efficacy and helping to fine-tune its parameters for optimal performance.

Figure 3.8 Text Recognition Dataset

3.8.2. Training Text Recognition with VietOCR

VietOCR is an open-source Optical Character Recognition (OCR) tool specifically designed for recognizing text in Vietnamese. It provides a robust platform for converting scanned documents, images containing text, and other media into editable and searchable text formats. VietOCR is tailored to meet the specific linguistic nuances and challenges of Vietnamese text recognition, making it a valuable tool for a wide range of applications. This library has proven to be quite accurate in Vietnamese character recognition and also provided an accurate pretrained model, upon which I fine-tune my model with the Text Recognition dataset.

Backbone	Config	Precision full sequence	time
VGG19-bn - Transformer	vgg_transformer	0.8800	86ms @ 1080ti
VGG19-bn - Seq2Seq	vgg_seq2seq	0.8701	12ms @ 1080ti

Figure 3.9 The accuracy of VietOCR model

3.8. Audio module

For this application audio conversion purpose, I use an open-source text-to-speech library, gTTS

3.8.1. gTTS

gTTS (Google Text-to-Speech) is a powerful Python library that provides an interface to Google Translate's text-to-speech API. This tool allows developers to convert written text into spoken words in various languages with high-quality, natural-sounding speech. gTTS is widely used in applications requiring text-to-speech functionality, such as voice assistants, language learning tools, and accessibility features for visually impaired users.

Key Features

- **Multilingual Support**: gTTS supports multiple languages and accents, making it versatile for global applications. It includes popular languages like English, Spanish, French, German, Chinese, Japanese, and many others.
- **Ease of Use**: The library is designed to be user-friendly and easy to integrate into Python projects. With a few lines of code, developers can convert text to speech and save the output as an audio file.
- **Customization**: Users can customize various aspects of the speech output, including speed (slow or fast), to match their specific needs.

Chapter 4: EXPERIMENTAL RESULT AND APPLICATION DEMO

4.1. Text Detection Module

4.1.1. Training result

After training using PaddleOCR with DB model and MobileNetv3 backbone, the receiving results are as follow:

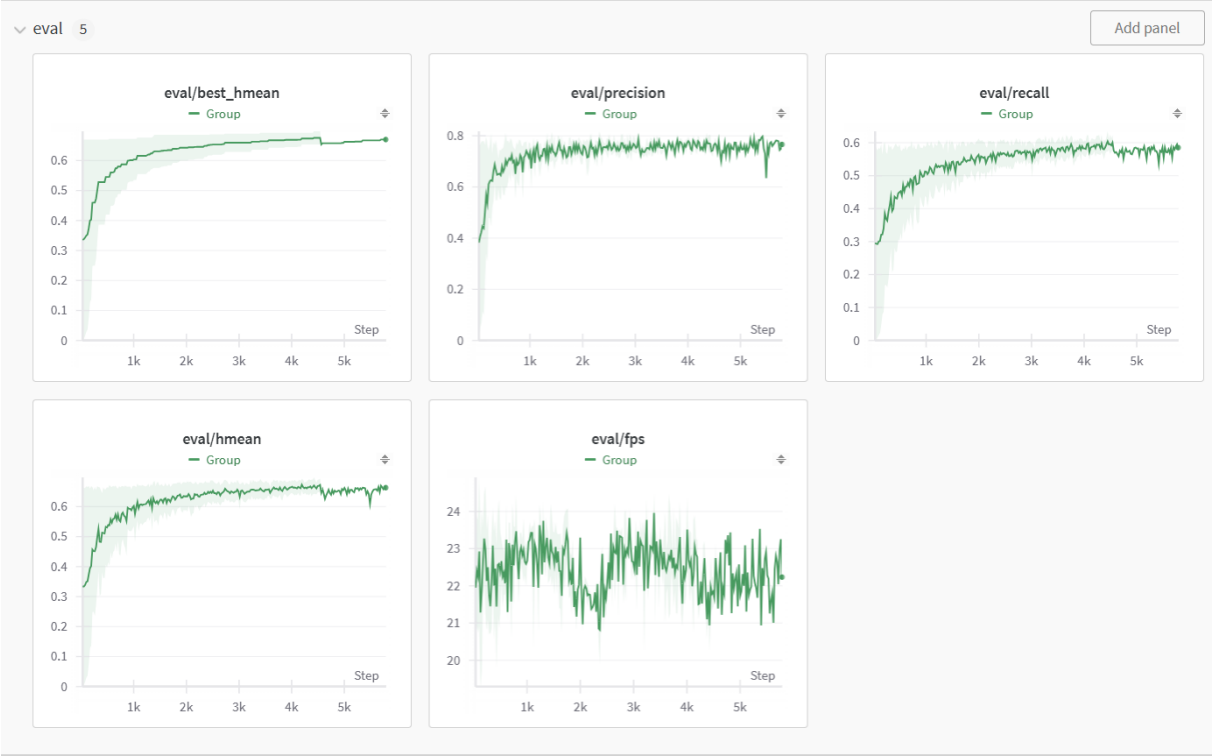


Figure 4.1 Evaluation result

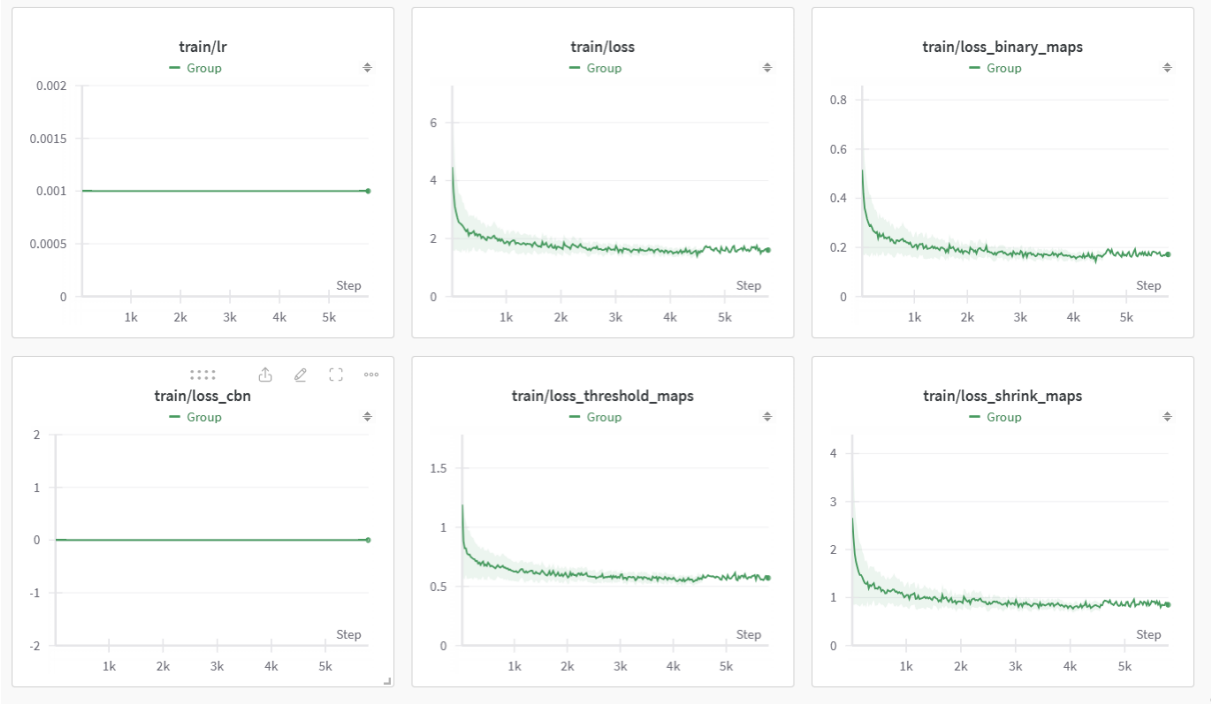


Figure 4.2 Training loss

After training the model, the evaluation process resulted in a precision score of 81.79%. This indicates that the model correctly identified 81.79% of the relevant instances out of all the instances it predicted as relevant.

4.1.2 Inference result

The model proved to be quite efficient in detecting texts, both in Vietnamese and English. The bounding boxes generated are accurate and suit the need of the project to detect word-by-word.

Here is the example inference result:

VĂN BẢN

CHIẾC LƯỢC NGÀ

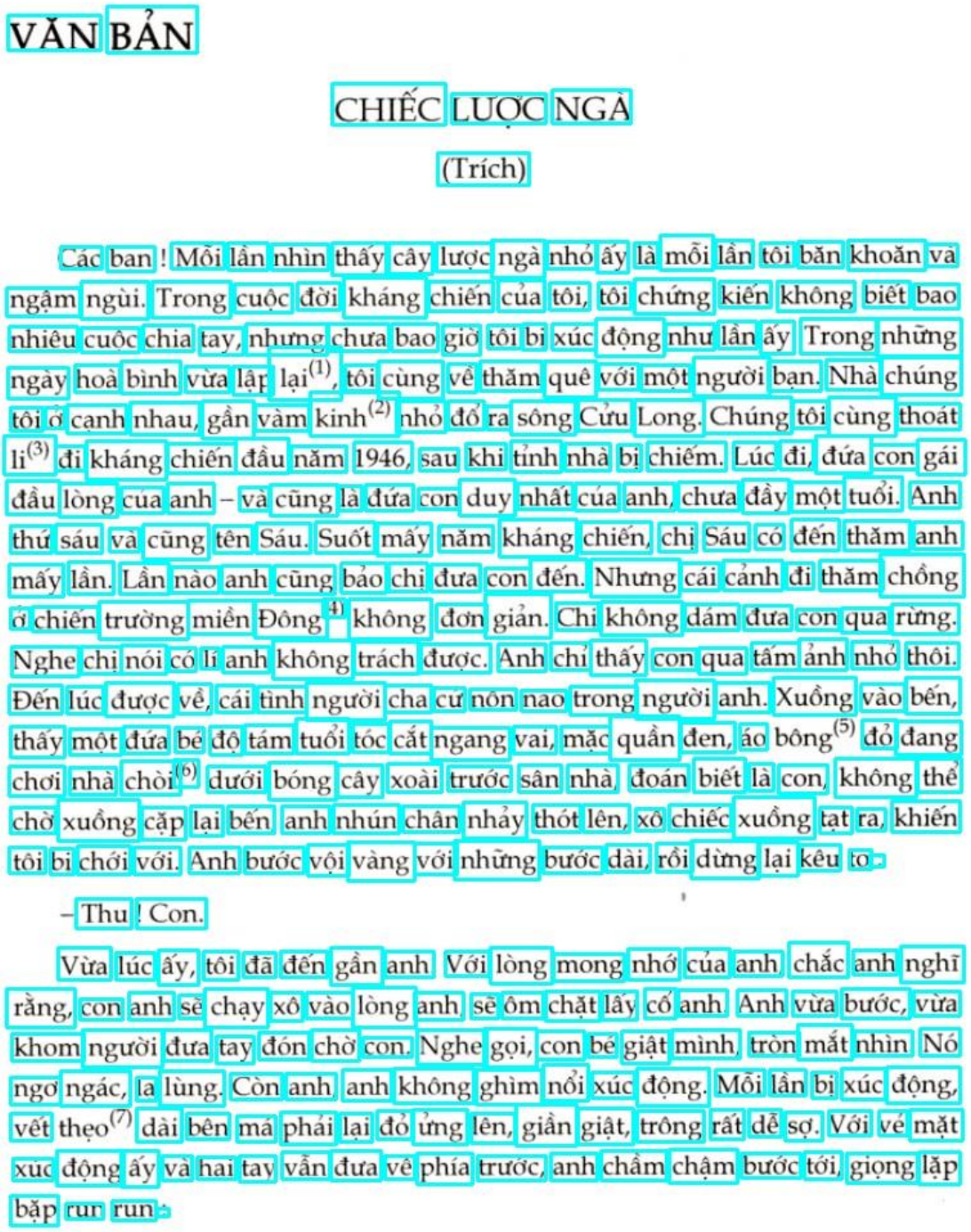
(Trích)

Các bạn ! Mỗi lần nhìn thấy cây lược ngà nhỏ ấy là mỗi lần tôi băn khoăn và ngậm ngùi. Trong cuộc đời kháng chiến của tôi, tôi chứng kiến không biết bao nhiêu cuộc chia tay, nhưng chưa bao giờ tôi bị xúc động như lần ấy. Trong những ngày hoà bình vừa lập lại(1), tôi cùng về thăm quê với một người bạn. Nhà chúng tôi ở cạnh nhau, gần vàm kinh(2) nhỏ đổ ra sông Cửu Long. Chúng tôi cùng thoát li(3) đi kháng chiến đầu năm 1946, sau khi tỉnh nhà bị chiếm. Lúc đi, đứa con gái đầu lòng của anh – và cũng là đứa con duy nhất của anh, chưa đầy một tuổi. Anh thứ sáu và cũng tên Sáu. Suốt mấy năm kháng chiến, chị Sáu có đến thăm anh mấy lần. Lần nào anh cũng bảo chị đưa con đến. Nhưng cái cảnh đi thăm chồng ở chiến trường miền Đông(4) không đơn giản. Chị không dám đưa con qua rừng. Nghe chị nói có lí anh không trách được. Anh chỉ thấy con qua tấm ảnh nhỏ thôi. Đến lúc được về, cái tình người cha cứ nôn nao trong người anh. Xuống vào bến, thấy một đứa bé độ tám tuổi tóc cắt ngang vai, mặc quần đen, áo bông(5) đỏ đang chơi nhà chòi(6) dưới bóng cây xoài trước sân nhà, đoán biết là con, không thể chờ xuống cặp lại bến, anh nhún chân nhảy thót lên, xô chiếc xuồng tạt ra, khiến tôi bị chới với. Anh bước vội vàng với những bước dài, rồi dừng lại kêu to :

– Thu ! Con.

Vừa lúc ấy, tôi đã đến gần anh. Với lòng mong nhớ của anh, chắc anh nghĩ rằng, con anh sẽ chạy xô vào lòng anh, sẽ ôm chặt lấy cổ anh. Anh vừa bước, vừa khom người đưa tay đón chờ con. Nghe gọi, con bé giật mình, tròn mắt nhìn. Nó ngơ ngác, lạ lùng. Còn anh, anh không ghìm nổi xúc động. Mỗi lần bị xúc động, vết thẹo(7) dài bên má phải lại đỏ ửng lên, giần giật, trông rất dễ sợ. Với vẻ mặt xúc động ấy và hai tay vẫn đưa về phía trước, anh chầm chậm bước tới, giọng lặp bặp run run :

Figure 4.3 Text Detection Input



VĂN BẢN

CHIẾC LƯỢC NGÀ

(Trích)

Các bạn ! Mỗi lần nhìn thấy cây lược ngà nhỏ ấy là mỗi lần tôi băn khoăn và ngậm ngùi. Trong cuộc đời kháng chiến của tôi, tôi chứng kiến không biết bao nhiêu cuộc chia tay, nhưng chưa bao giờ tôi bị xúc động như lần ấy. Trong những ngày hoà bình vừa lập lại[1], tôi cùng về thăm quê với một người bạn. Nhà chúng tôi ở cạnh nhau, gần vàm kinh[2] nhỏ đổ ra sông Cửu Long. Chúng tôi cùng thoát li[3] đi kháng chiến đầu năm 1946, sau khi tỉnh nhà bị chiếm. Lúc đi, đứa con gái đầu lòng của anh – và cũng là đứa con duy nhất của anh, chưa đầy một tuổi. Anh thứ sáu và cũng tên Sáu. Suốt mấy năm kháng chiến, chị Sáu có đến thăm anh mấy lần. Lần nào anh cũng bảo chị đưa con đến. Nhưng cái cảnh đi thăm chồng ở chiến trường miền Đông[4] không đơn giản. Chị không dám đưa con qua rừng. Nghe chị nói có lí anh không trách được. Anh chỉ thấy con qua tấm ảnh nhỏ thôi. Đến lúc được về, cái tình người cha cứ nôn nao trong người anh. Xuống vào bến, thấy một đứa bé độ tám tuổi tóc cắt ngang vai, mặc quần đen, áo bông[5] đỏ đang chơi nhà chòi[6] dưới bóng cây xoài trước sân nhà, đoán biết là con, không thể chờ xuống cặp lại bến anh nhún chân nhảy thót lên, xô chiếc xuồng tạt ra, khiến tôi bị chới với. Anh bước vội vàng với những bước dài, rồi dừng lại kêu to :

– Thu ! Con.

Vừa lúc ấy, tôi đã đến gần anh. Với lòng mong nhớ của anh chắc anh nghĩ rằng, con anh sẽ chạy xô vào lòng anh, sẽ ôm chặt lấy cổ anh. Anh vừa bước, vừa khom người đưa tay đón chờ con. Nghe gọi, con bé giật mình, tròn mắt nhìn. Nó ngơ ngác, la lùng. Còn anh, anh không ghìm nổi xúc động. Mỗi lần bị xúc động, vết thẹo[7] dài bên má phải lại đỏ ửng lên, giần giật, trông rất dễ sợ. Với vẻ mặt xúc động ấy và hai tay vẫn đưa về phía trước, anh chầm chậm bước tới, giọng lặp bặp run run :

Figure 4.4 Text Detection Output

4.2. Text Recognition Module

4.2.1. Training

After training TransformerOCR model using VietOCR on my dataset, The training progress are as followed:

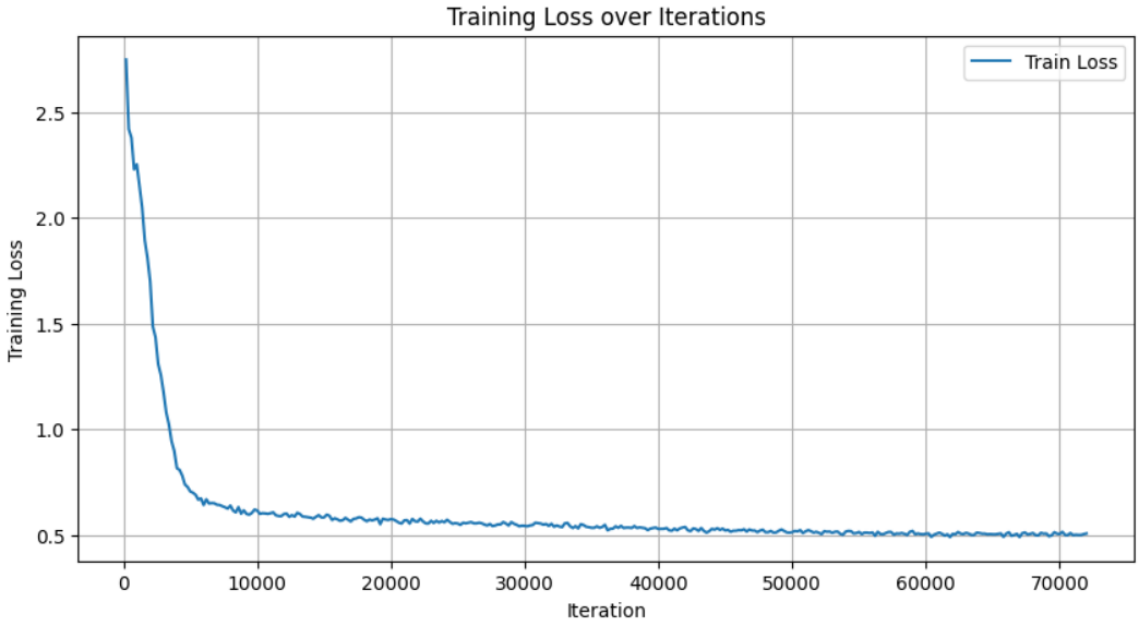


Figure 4.5 Training loss

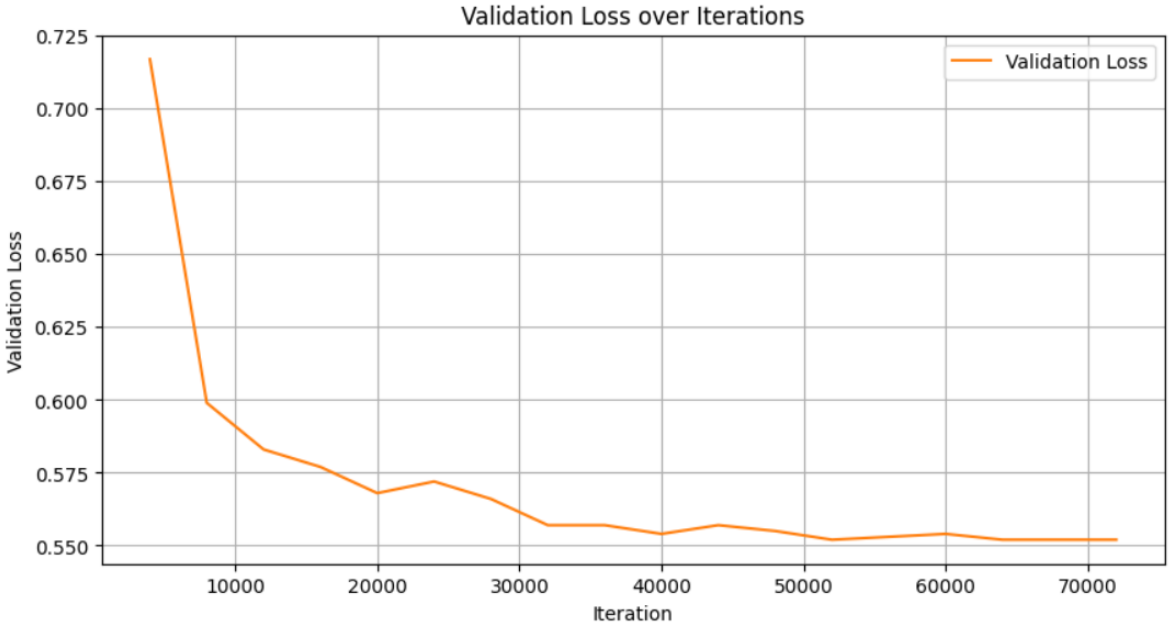


Figure 4.6 Validation Loss

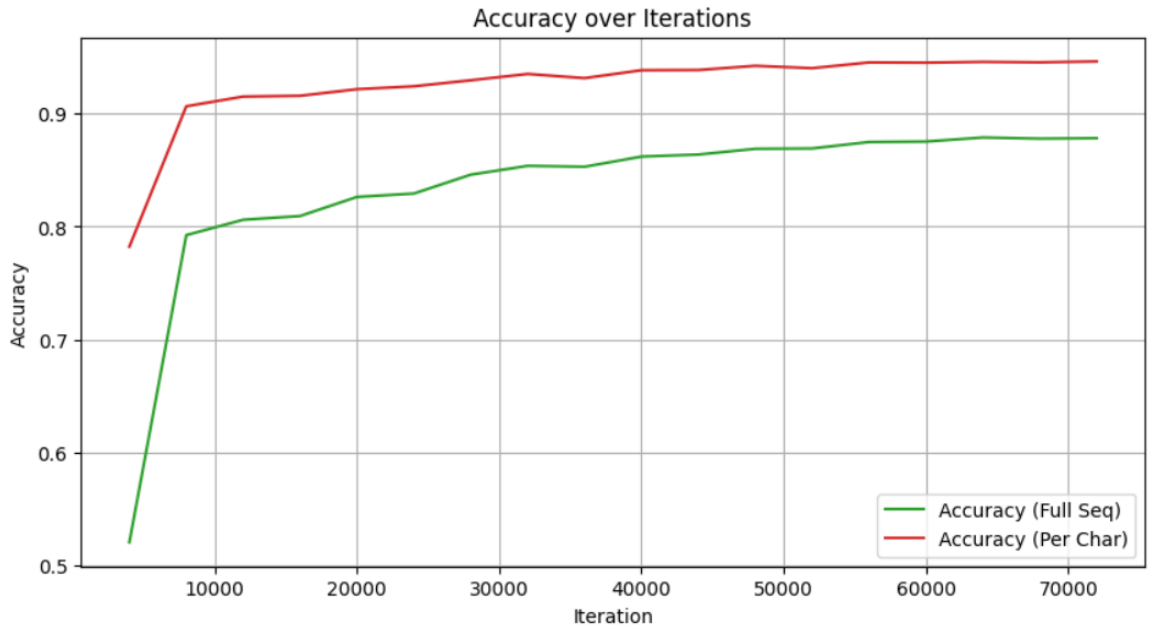


Figure 4.7 Accuracy

As the graph illustrates, the accuracy of the model for individual characters is remarkably high, reaching up to 95%. In contrast, the accuracy for entire sequences is 87%. This discrepancy highlights the model's proficiency at recognizing and correctly identifying single characters, while also demonstrating a strong performance in handling complete sequences, albeit with slightly reduced accuracy.

These results prove that the model is highly capable and reliable for tasks involving character recognition. Its ability to achieve a 95% accuracy rate for individual characters suggests a robust understanding of the finer details within the text. Meanwhile, the 87% accuracy for whole sequences indicates that the model effectively manages more complex tasks involving the context and structure of text sequences, though there is a slight drop in accuracy compared to single characters.

Overall, the model's performance metrics indicate a high level of precision and reliability, making it well-suited for practical applications in text detection and recognition across different languages, including Vietnamese and English.

4.2.2 Inference

The inference result of this module is quite good, as illustrated below:

Figure 4.8 Text Recognition Input

```
$ python utils/recognition.py
('Giới thiệu sương sương một chút, ONNX Runtime là bộ công cụ giúp tăng tốc', 0.9267798645855629)
```

Figure 4.9 Text Recognition Output

4.3. Image to audio module

Thanks to the high accuracy of both the Text Recognition Module and the Text Detection Module, the Image to Audio Module has demonstrated exceptional performance. The Text Recognition Module, with an individual character accuracy of up to 95%, ensures that characters are accurately identified, contributing to the overall precision of the text extraction process. Similarly, the Text Detection Module effectively identifies and isolates text within images, providing a reliable foundation for subsequent recognition tasks.

The combination of these two modules allows the Image to Audio Module to convert text from images into speech with high fidelity. The precise detection and recognition of text ensure that the spoken output is accurate and intelligible, enhancing the user experience. This integrated system proves to be particularly useful in applications such as assisting visually impaired individuals, real-time language translation, and automated reading of documents.

Overall, the high accuracy of the Text Recognition and Text Detection Modules significantly boosts the efficacy of the Image to Audio Module, making it a powerful tool for converting visual text data into audio format.

Example output based on the input of Text Detection Module

```
Cropping images took 0.46 seconds.
Predicting text took 50.49 seconds.
Organizing text took 0.00 seconds.
VĂN BẢN
CHIẾC LƯỢC NGÀ
(Trích)
Các bạn Mỗi lần nhìn thấy cây lược ngà nhỏ ấy là mỗi lần tôi băn khoăn và
ngậm ngùi. Trong cuộc đời kháng chiến của tôi, tôi chứng kiến không biết bao
nhiêu cuộc chia tay nhưng chưa bao giờ tôi bị xúc động như lần ấy. Trong những
ngày hoà bình vừa lập lại tôi cùng về thăm quê với một người bạn. Nhà chúng
tôi ở cạnh nhau gần vàm kinh (2) nhỏ đổ ra sông Cửu Long- Chúng tôi cùng thoát
li(3) đi kháng chiến đầu năm 1946, sau khi tỉnh nhà bị chiếm. Lúc đi, đứa con gái
đầu lòng của anh - và cũng là đứa con duy nhất của anh, chưa đầy một tuổi. Anh
thứ sáu và cũng tên Sáu. Suốt mấy năm kháng chiến, chị Sáu có đến thăm anh
mấy lần. Lần nào anh cũng bảo chị đưa con đến. Nhưng cái cảnh đi thăm chồng
ở chiến trường miền Đông (4) không đơn giản. Chị không đám đưa con qua rừng
Nghe chị nói có lí anh không trách được. Anh chỉ thấy con qua tấm ảnh nhỏ thôi.
Đến lúc được VỀ, cái tình người cha cứ nôn nao trong người anh. Xuồng vào bến
thấy một đứa bé độ tám tuổi tóc cắt ngang vai, mặc quần đen áo đỏ đang
chơi nhà chòi(6) dưới bóng cây xoài trước sân nhà, đoán biết là con, không thế
chờ xuồng cặp lại bến anh nhún chân nhảy thót lên, XÔ chiếc xuồng tạt ra, khiến
tôi bị chới với. Anh bước vội vàng với những bước dài, rồi dừng lại kêu to
Thu !Con.
Vừa lúc ấy, tôi đã đến gần anh. Với lòng mong nhớ của anh chắc anh nghĩ
rằng, con anh sẽ chạy XÔ vào lòng anh sẽ ôm chặt lấy Cổ anh. Anh vừa bước, vừa
khom người đưa tay đón chờ con. Nghe gọi, con bé giật mình tròn mắt nhìn. Nó
ngơ ngác, lạ lùng Còn anh anh không ghìm nổi xúc động. Mỗi lần bị xúc động,
vết thẹo(7) dài bên má phải lại đỏ ửng lên, giần giật, trông rất đẽ SỢ Với vẻ mặt
xúc động ấy và hai tay vẫn đưa về phía trước, anh chầm chậm bước tới, giọng lặp
bặp run run
```

Figure 4.10 Example output

Despite the relatively high processing time, the system proves to be quite efficient in converting text to audio with accurate inference. The robust performance of the Text

Recognition and Text Detection Modules ensures that the extracted text is precise, which is critical for the subsequent audio conversion process.

4.4. Application Demo

4.4.1. Login

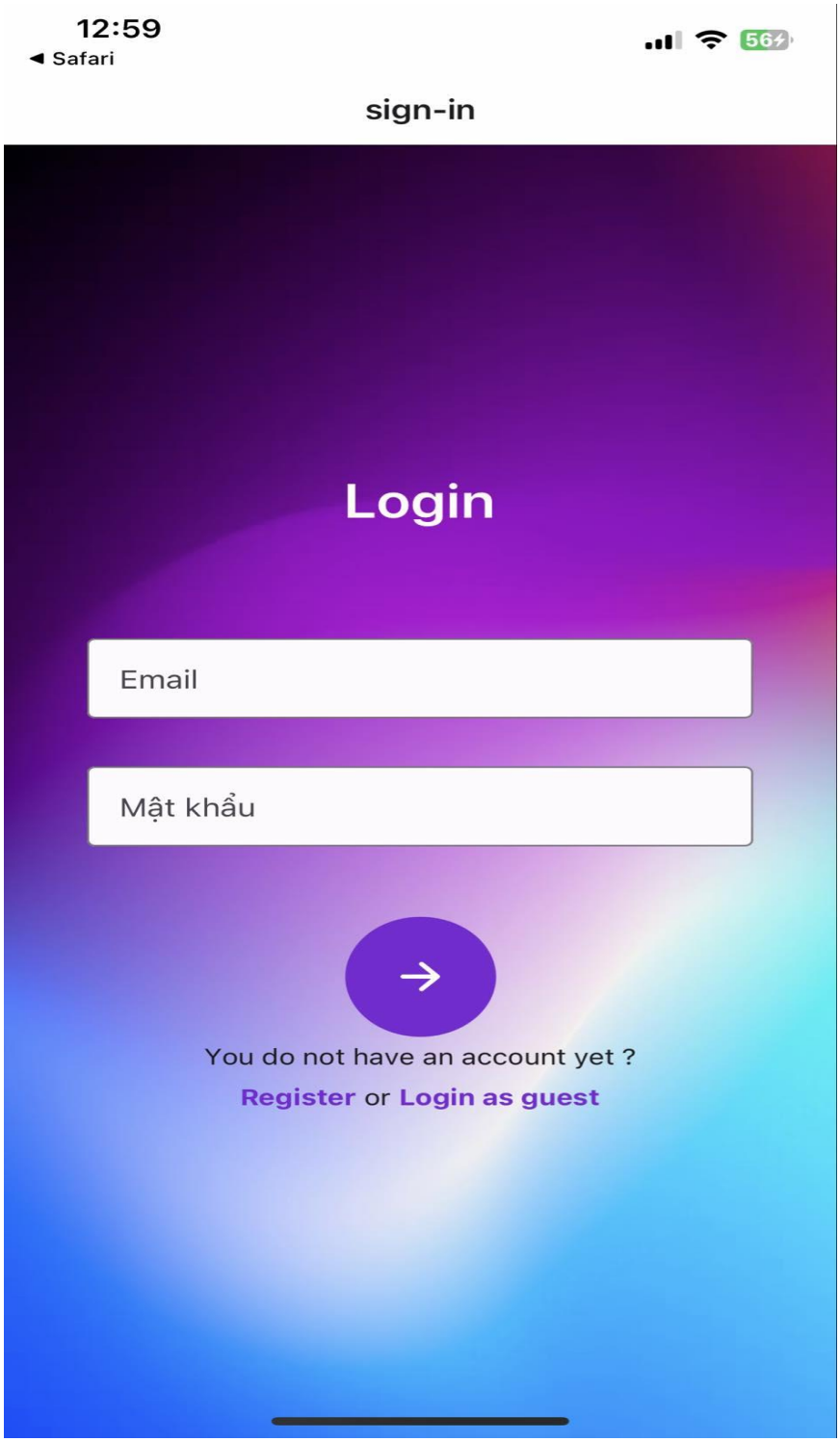


Figure 4.11 Login screen

4.4.2. Register

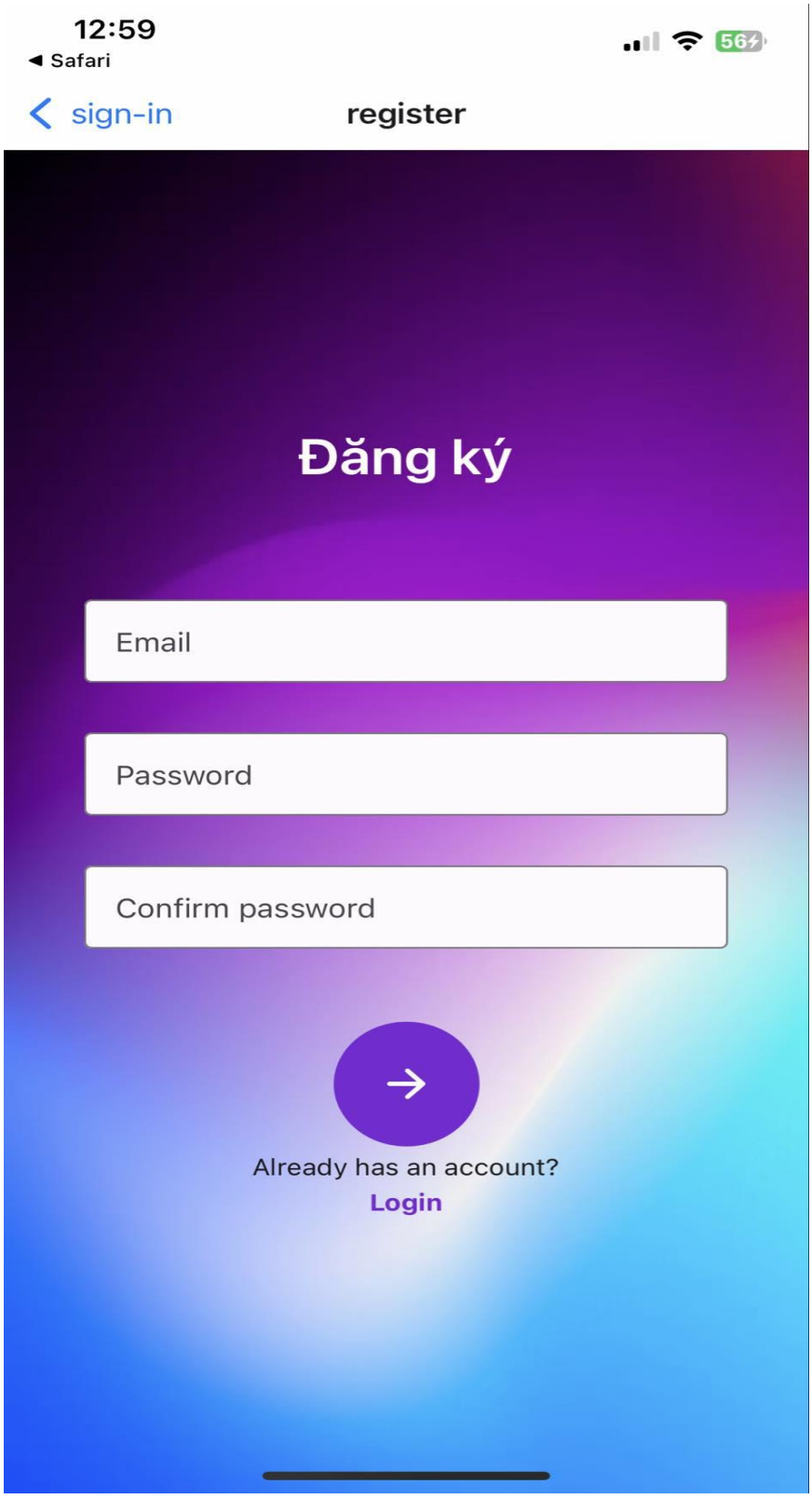


Figure 4.12 Register screen

4.4.3 Capture or upload image screen

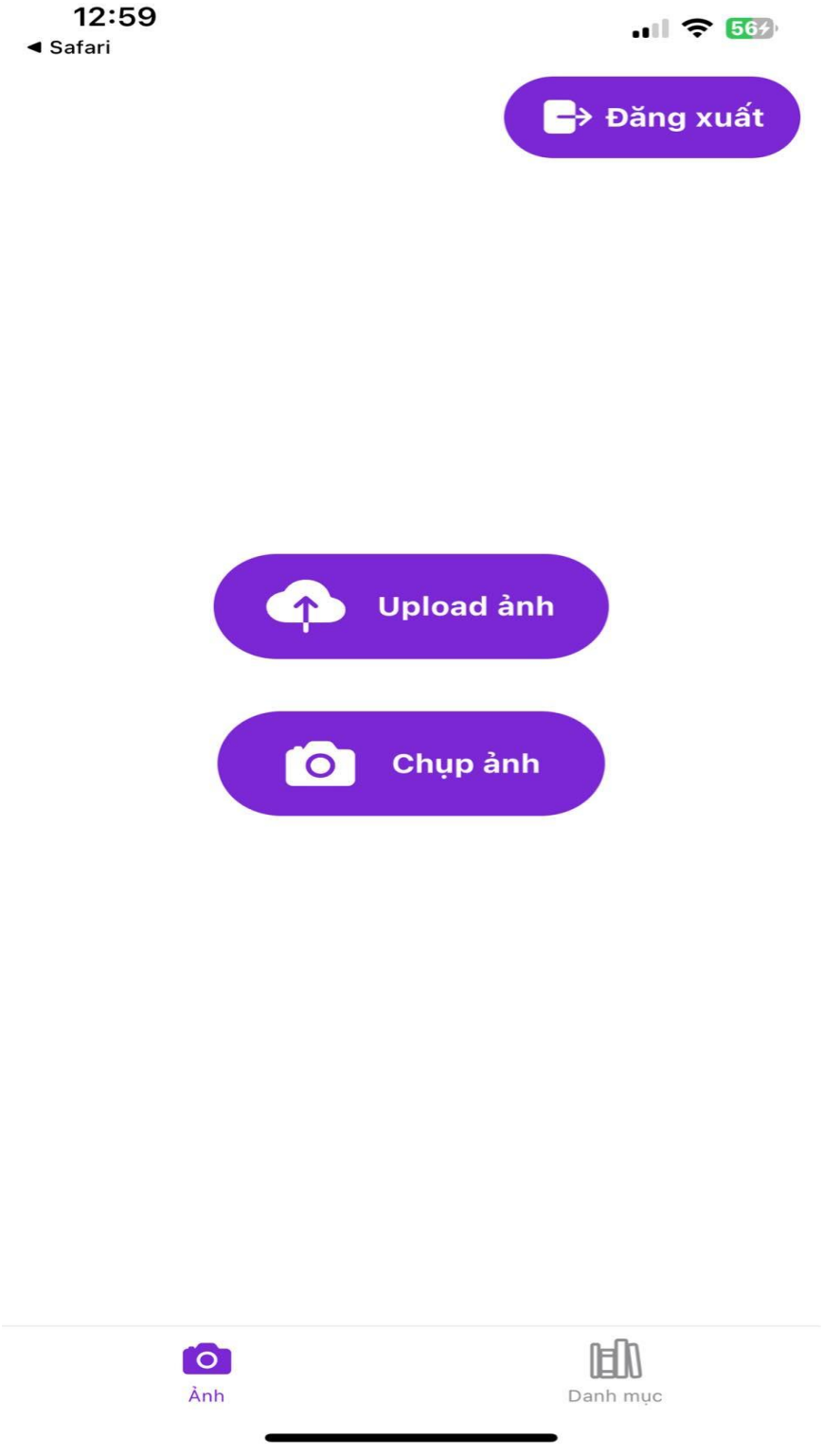


Figure 4.13 Capture or upload image screen

4.4.4 Format image screen

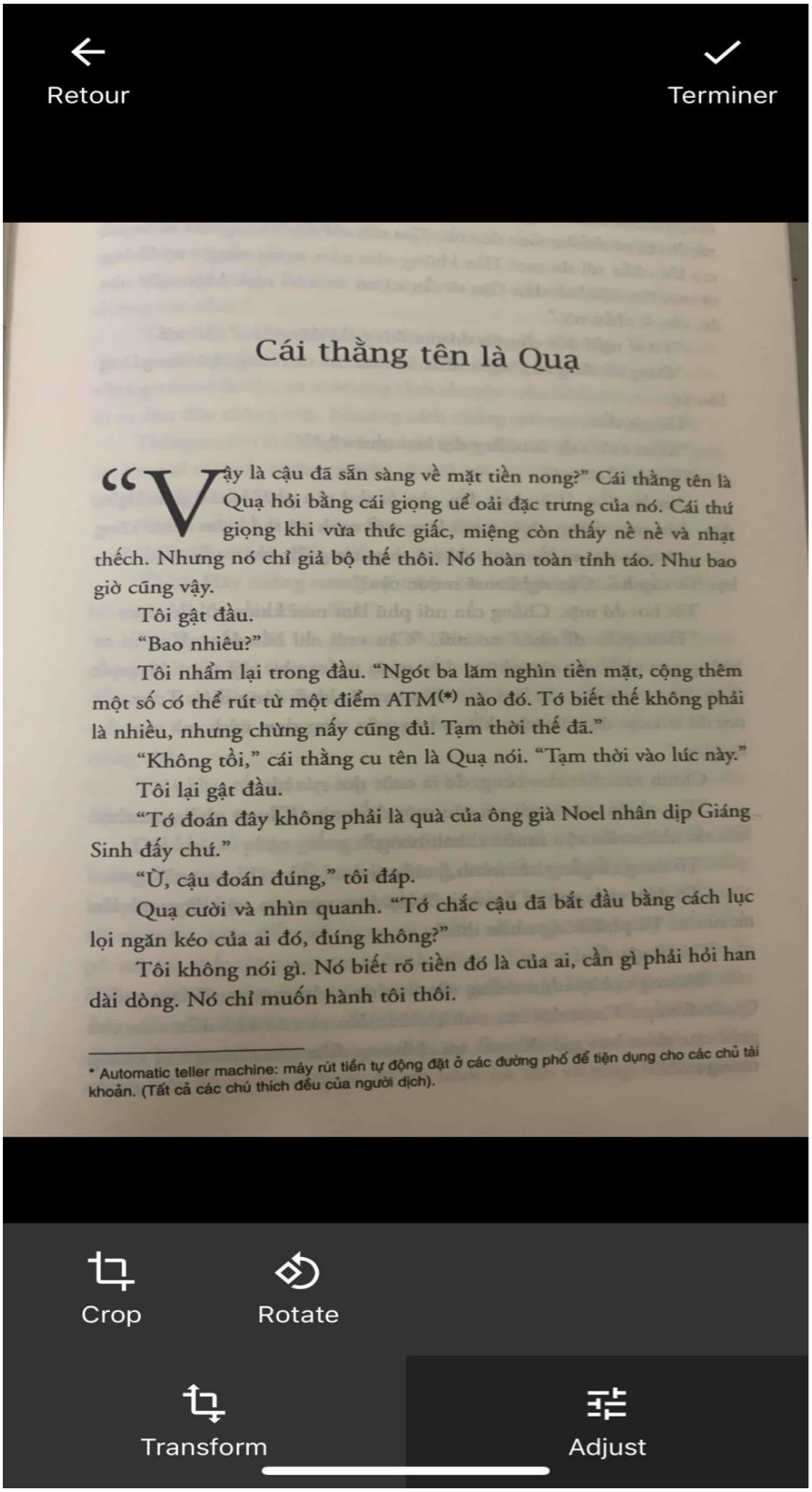


Figure 4.14 Format image

4.4.5. Document list screen

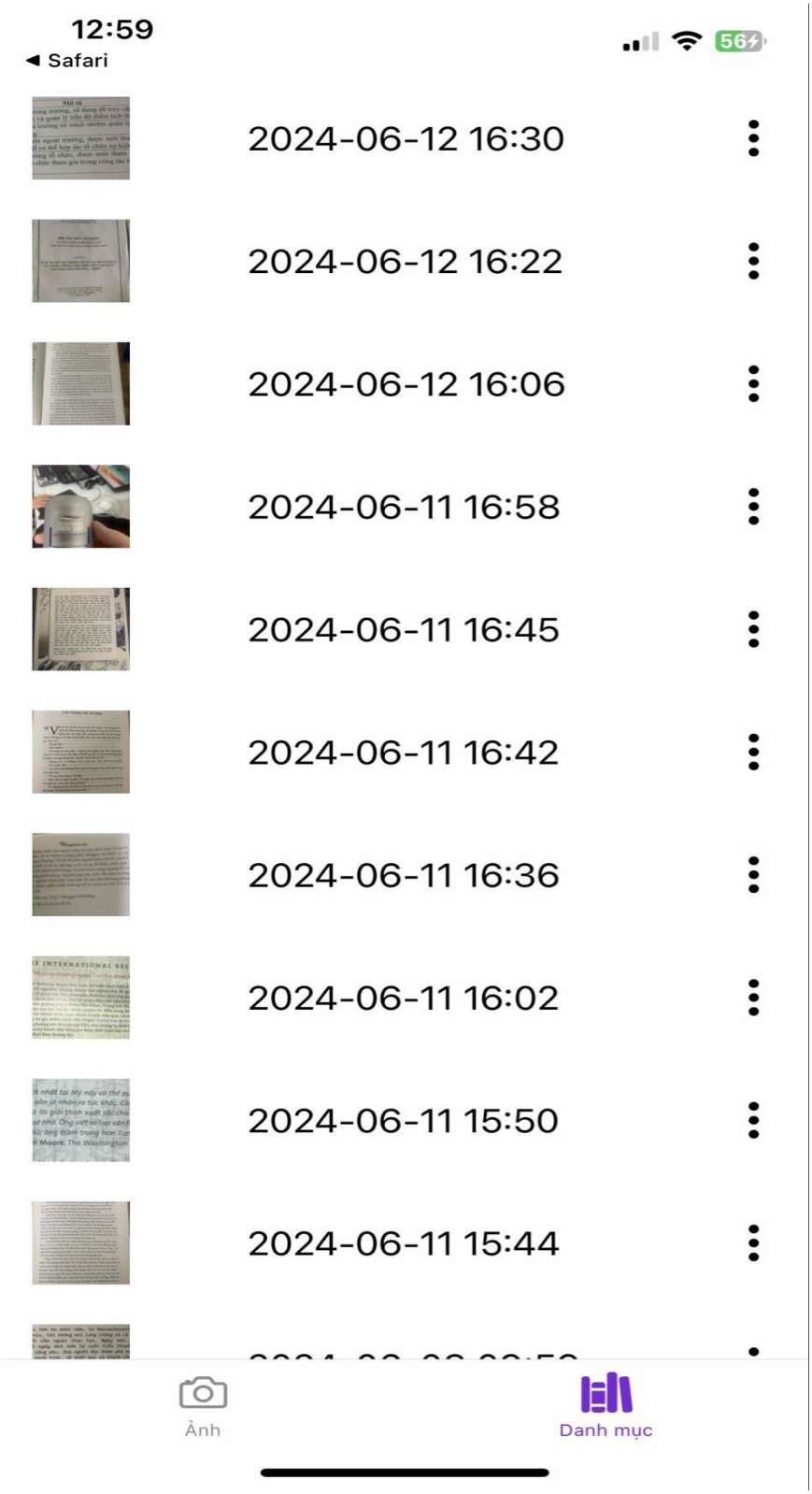


Figure 4.15 Document list screen

4.4.6. Document detail screen

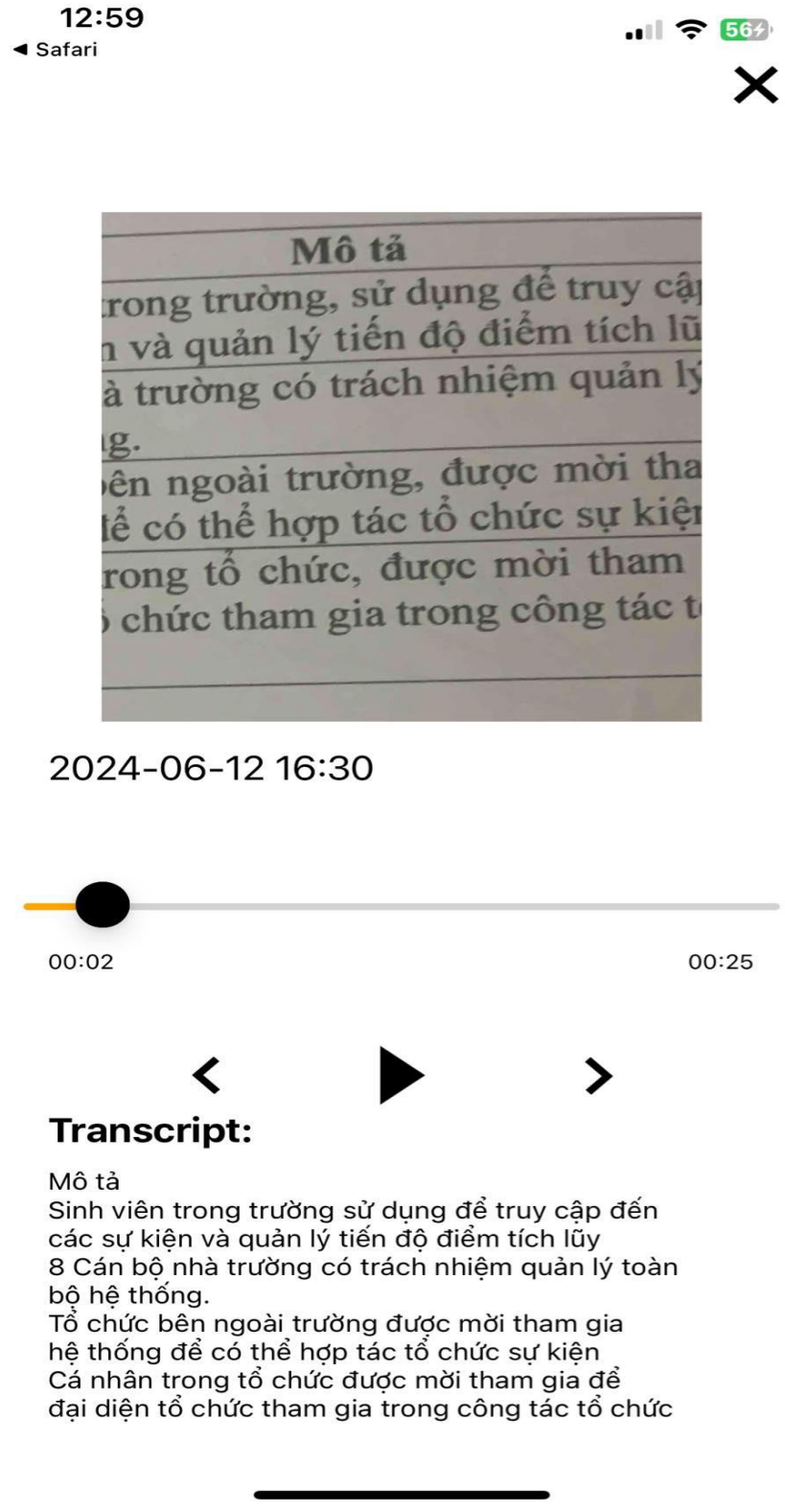


Figure 4.16 Document detail screen

Chapter 5: CONCLUSION

5.1. Result

After building this project, I have identified several pros and cons in our application:

Pros:

- **User-Friendly Interface**:

The application is designed to be intuitive and easy to use, making it accessible for users of varying technical skills. The straightforward interface ensures that users can quickly learn how to operate the application without requiring extensive training or support.

- **High-Quality Inference Results**:

The inference results are impressive, producing comprehensible audio files that users can easily hear and understand. This high level of accuracy in converting text to speech significantly enhances the user experience, particularly for those who rely on audio outputs for accessibility reasons.

Cons:

- **Long Processing Time**:

One of the main drawbacks is the prolonged processing time. The application takes a considerable amount of time to analyze and convert text from images into audio. This can be a significant inconvenience for users who need quick results or are working with large volumes of text.

- **Difficulty with Complex Layouts**:

The application struggles with reorganizing text when the layout of the page is too complicated. This limitation means that documents with intricate designs, such as those with multiple columns, tables, or non-linear text arrangements, may not be processed accurately. As a result, the output may not reflect the intended structure and flow of the original document.

- **Limitation to Straight Images**:

The current implementation only works effectively with straight images. If an image is tilted, skewed, or captured at an angle, the application may fail to detect and recognize the text correctly. This constraint limits the application's usability in real-world scenarios where perfect image alignment cannot always be ensured.

- **UI/UX:**
  The current UI/UX of the application still need refinement to bring the best experience for the user

5.2. Future work

Given the disadvantages identified in the evaluation part of this project, the future work will focus on addressing these limitations to enhance the application's performance and usability. The proposed improvements and extensions are as follows:

- Optimizing processing time either by using GPU accelerators or by refining the algorithms.
- Training the model to be able to comprehend complex layout texts or words in different angles
- Optimize code for saving infrastructure cost as well as handling more user.
- Redesign the UI/UX with helps and feedbacks of designer

REFERENCES

[1] Real-time Scene Text Detection with Differentiable Binarization, [1911.08947] Real-time Scene Text Detection with Differentiable Binarization (arxiv.org)

[2] Transformer-based Optical Character Recognition 2109.10282 (arxiv.org)

[3] Recognize Vietnamese with VietOCR Nhân dang tiếng Viêt cùng với Transformer OCR (viblo.asia)

[4] React Native with Expo Go Expo Documentation

[5] What is OCR? What is Optical Character Recognition (OCR): [2024 update] (ubiai.tools)

[6] Introduction to Flask Introduction to Web development using Flask - GeeksforGeeks

[7] Introduction to React Native Introduction · React Native

[8] BKAI-NAVER Challenge 2022 - Vietnamese Scene Text Detection and Recognition AIHUB.ML - Competition

[9] PaddleOCR for Vietnamese Sử dụng thư viện PaddleOCR trong bài toán nhận dang chữ Tiếng Việt - THI GIÁC MÁY TÍNH (thigiacmaytinh.com)

[10] Introduction to MongoDb MongoDB Tutorial (tutorialspoint.com)

[11] Authentication with Firebase Authentication with Firebase and React.js: A Complete Guide (tutorialspoint.com)

[12] Upload file with CloudFlare R2 Cloudflare R2 | Zero Egress Fee Object Storage | Cloudflare | Cloudflare