Vietnamese Automatic Speech Recognition Utilizing Auditory and Visual Data

Thinh Duong, Minh Nguyen, Khanh Pham

Supervisor: Dr. Hai Le Thanh

DEPARTMENT OF ITS

FPT UNIVERSITY HO CHI MINH

A final year capstone project submitted in partial fulfillment of the requirement for the Degree of Bachelor of Artificial Intelligent in Computer Science

August 2024

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to our supervisor, Professor Hai Le Thanh, for his invaluable guidance and support throughout the duration of this thesis. His invaluable guidance, drawn from his wealth of experience and profound expertise, significantly enhanced the completion of this study. We extend our appreciation to Professor Trung Nguyen Quoc and Professor Tien Nguyen Quoc for their constructive feedback during the review process, which greatly contributed to enhancing the quality of this thesis. Additionally, we wish to acknowledge Thinh Duong Tan and Duy Tran Nguyen Nhut for providing the essential dataset of audio and visual data of Vietnamese speakers.

AUTHOR CONTRIBUTIONS

This thesis, which includes the stages of research, data preparation, method exploration, training, evaluation, experimentation, analysis, demo software development, documentation, and report writing, was conducted through the collaboration of Thinh Duong, Minh Nguyen, and Khanh Pham. As co-authors, Thinh Duong, Minh Nguyen, and Khanh Pham take full responsibility for all aspects of this work. The contributions of each author collectively ensured the research was completed comprehensively and effectively, resulting in a robust and accurate Audio-Visual Automatic Speech Recognition (AV-ASR) system for Vietnamese. The authors worked together to ensure that all aspects of the research were executed with meticulous detail.

Thinh Duong led the research and design of the Audio-Visual Automatic Speech Recognition (AV-ASR) system and was responsible for developing the deep learning models. Additionally, Thinh Duong conducted data analysis and performance evaluation, set up experiments, and analyzed the results, including testing and optimizing models and algorithms.

Minh Nguyen was in charge of data collection and preparation for the research, which involved gathering audio and visual data from various sources. Minh Nguyen also handled data preprocessing and the construction of training datasets. Furthermore, Minh Nguyen contributed to the development and deployment of deep learning models and the creation of the demo software.

Khanh Pham also played a role in data collection and preparation, similar to Minh Nguyen, by gathering audio and visual data from diverse sources. Khanh Pham was involved in data preprocessing and constructing the training dataset. Additionally, Khanh Pham contributed to data analysis, the development of the demo software, and supported the writing of research report sections.

ABSTRACT

Automatic Speech Recognition (ASR) has become a crucial research area with numerous practical applications in daily life. However, traditional ASR systems face significant challenges when dealing with non-ideal audio conditions. This study introduces a novel approach that combines audio and visual data for Vietnamese speech recognition, utilizing Large Language Models (LLMs) to enhance performance and contextual understanding. This research is the first in Vietnam to explore automatic speech recognition using both audio and visual data. We developed the ViAVSP-LLM system, consisting of two main components: the Audio-Visual Hidden Unit BERT (AV-HuBERT) encoder and the VinaLLaMA decoder. AV-HuBERT learns self-supervised representations for audio-visual speech by masking multi-stream video input and predicting automatically discovered multi-modal hidden units, fine-tuning iteratively. Recognizing that there is redundant information in the input frames, we propose a novel deduplication method to reduce embedded visual features using visual speech units. Through the proposed deduplication and Low-Rank Adaptation (LoRA), VinaLLaMA, a large language model tailored for Vietnamese, can be fine-tuned efficiently. On the VASR dataset (1,045 hours), ViAVSP-LLM achieved a Word Error Rate (WER) of 10.04% on the validation set and 12.03% on the test set, along with a Character Error Rate (CER) of 7.2% on the test set. We conducted experiments to compare the performance of the ViAVSP-LLM system with audio-only models. The results show that combining audio and visual data significantly improves the accuracy the speech recognition system. As part of this thesis project, a Minimum Viable Product (MVP) was developed to test audio-visual speech-to-text recognition. The research results open new avenues for speech recognition technology applications in various fields, from daily communication to assisting people with disabilities, while also providing a solid foundation for future research in this area. The report also outlines the limitations of this project and proposes future work to further develop and enhance speech recognition models.

Keywords: *Audio-Visual Speech Recognition, Audio-Visual Hidden Unit BERT, Large Language Model, LLaMA, Low-Rank Adaptation*

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Automatic Speech Recognition (ASR) has become an important research field with many practical applications in daily life. ASR applications include intelligent virtual assistants (such as Siri, Google Assistant), automated call systems, customer service via phone, and smart home devices [1, 2, 3, 4, 5]. The development of deep learning technologies and signal processing has brought significant advancements to ASR systems, improving their accuracy and performance. However, developing an effective ASR system for Vietnamese, a language with a complex phonetic structure and numerous dialectal variations, remains a challenge [6]. In Vietnamese, similar sounds can have different meanings depending on the context, making speech recognition more complicated. Additionally, the presence of ambient noise in real-world environments, such as traffic noise, other people's speech, or environmental factors, also significantly impacts the performance of traditional audio-based ASR systems [7]. Addressing these challenges requires innovative solutions to enhance the accuracy and robustness of ASR systems under diverse real-world conditions.

To tackle these challenges, combining audio and visual data has been proposed as a new research direction, offering opportunities to improve ASR accuracy [8]. Information from visual data, such as lip movements and mouth features of speakers, can provide additional context and aid in more accurate speech recognition, especially in noisy environments. This research focuses on developing a multi-modal automatic speech recognition system for Vietnamese, utilizing both audio and visual data. The goal is to build a model capable of accurately recognizing words and sentences in Vietnamese, even in challenging environmental conditions. This study not only contributes to the field of automatic speech recognition by providing an effective solution for Vietnamese but also opens up new research avenues in integrating multimedia data to enhance the performance of ASR systems. Developing a multi-modal ASR system for Vietnamese holds scientific significance and can have a profound impact on daily life, improving user support services and enhancing human-machine interaction in Vietnamese contexts.

This Introduction chapter lays the foundation for exploring the subsequent chapters, providing an overview of the purpose, objectives, and research methods used in this study. Following this introduction, the next sections will provide an overview of the project, followed by an exploration of what motivated the researchers to conduct this study both practically and technically. Subsequently, specific objectives will be outlined, and the structure of this report will be briefly described to lead into the following sections.

1.1 Overview

Speech is conveyed not only through sound but also through the movements of the mouth, which play a crucial role in human communication. With the increasing recognition of the importance of visual data in speech recognition, image-based processing technologies have been rapidly developing [9]. For example, Visual Speech Recognition (VSR) allows for the recognition of spoken words solely by observing lip movements, without accessing audio [10]. Recent advances in the field also highlight the growing interest in multimodal approaches that combine both audio and visual data to improve speech recognition accuracy. These approaches leverage the complementary nature of audio and visual cues to create more robust and reliable speech recognition systems. By integrating audio-visual data, these systems can better handle challenging conditions such as noisy environments or when one modality is partially occluded.

Recently, the application of Large Language Models (LLMs) has garnered significant attention due to their powerful and flexible context modeling capabilities [11]. Motivated by the recent successes of LLMs, we investigate whether the rich context modeling abilities of LLMs can be applied to audio-visual speech processing and alleviate the ambiguity of homophenes.

In this study, we propose a novel framework named Vietnamese Audio-Visual Speech Processing integrated with LLMs (ViAVSP-LLM), which employs integrated learning between visual speech and LLM's contextual space. ViAVSP-LLM uses a self-supervised model to convert visual speech into phoneme-level representations, facilitating the linkage of phonetic information to text. To alleviate computational load during training, we use a redundancy reduction method that reduces the input sequence length for LLMs. Specifically, we use visual speech units, which are phoneme-level representations from the self-supervised model, as indicators of redundant information between sequences. Visual speech features assigned to the same unit are averaged to minimize redundant processing and improve computational efficiency. Finally, ViAVSP-LLM is trained simultaneously to perform Audio-Visual Speech Recognition (AV-ASR) with a single model, marking the first study of its kind in Vietnam. The following section, Motivation, will further elucidate why Audio Visual Speech Recognition was chosen for this project in both practical and technical motivation.

1.2 Motivation

1.2.1 Practical motivation

In today's rapidly advancing digital age, Automatic Speech Recognition (ASR) systems play a crucial role across various aspects of life. From facilitating everyday personal communication to specialized applications in healthcare, education, and customer service, ASR has significantly enhanced efficiency and convenience [12, 13, 14, 15]. For instance, in education, such a system can effectively support remote teaching and learning by automatically converting speech into text, thereby improving accessibility and teaching efficiency. In healthcare, ASR integrated with audio and visual data can automate medical record documentation processes, enhance the ability to detect health issues from speech, and convert them into easily manageable and analyzable data. The rapid development of information technology has also unlocked new potential applications for ASR, from automating workflow processes to improving user experiences and developing data-driven intelligent services [16, 17]. For current ASR applications, handling and recognizing speech in noisy environments remains a major challenge [18]. By integrating visual data, ASR systems can utilize mouth features and lip movements to enhance speech recognition accuracy, especially in challenging conditions like noisy spaces

[19].

Applying ASR to Vietnamese still faces significant challenges. Previous research has predominantly focused on languages and cultures such as English, optimizing ASR systems for their phonetic characteristics and vocabulary. Yet, this approach cannot fully translate to Vietnamese due to its unique tonal, rhythmic features, and pronunciation variations. This poses considerable challenges in developing and deploying effective and accurate ASR systems for Vietnamese [20]. In this context, developing an efficient ASR system for Vietnamese by integrating audio and visual data offers practical benefits.

1.2.2 Technical motivation

The technical motivation behind this research stems from the unique challenges and opportunities presented by the Vietnamese language and the integration of multimodal data for speech recognition [21]. Vietnamese, with its complex phonological structure and numerous dialectal variations, poses significant difficulties for traditional ASR systems that rely solely on audio data. These systems often struggle with accurately recognizing and transcribing speech, especially in noisy environments where a single audio signal may not be sufficient for ensuring accuracy.

One of the primary technical drivers is leveraging both audio and visual data to create a more robust and reliable speech recognition system [22, 23]. By incorporating visual cues from lip movements, the system can better disambiguate phonemes and tones, which are critical in a tonal language like Vietnamese. This multimodal approach not only helps in enhancing the recognition accuracy in ideal conditions but also significantly improves performance in noisy environments where audio signals alone might be insufficient. Another technical motivation is addressing the high costs and large data requirements associated with current ASR models. Traditional models require extensive transcribed video data for training, which is both time-consuming and costly to collect. Therefore, this research develops a self-supervised learning framework capable of effectively utilizing unlabeled audio-visual data. The goal is to reduce dependence on large labeled datasets, making the training process more efficient and scalable.

This project aims to confront these technical challenges by exploring the effectiveness of multimedia speech recognition methods. The objective of the research is to advance state-of-the-art AV-ASR technology and contribute to the development of more accurate and efficient speech recognition systems for Vietnamese, while also expanding the applicability of this technology in various real-world conditions.

1.3 Project Objectives and Chapter Overview

This research aims to develop an effective Automatic Speech Recognition (ASR) system for Vietnamese by integrating audio and visual data, thereby overcoming the limitations of current ASR systems. The goal is to create an advanced ASR system capable of performing well in various real-world scenarios. Additionally, this study represents a significant advancement in multimodal data integration, opening new research directions and providing innovative solutions to current technical challenges.

Following an in-depth exploration, the project's objectives will include:

- **Objective 1:** Investigate and create an innovative ASR model tailored specifically for the Vietnamese language, integrating both auditory and visual cues. Specifically, the de-

velopment of the ViAVSP-LLM system includes two main components: the Audio-Visual Hidden Unit BERT (AV-HuBERT) encoder and the VinaLLaMA decoder. AV-HuBERT learns self-supervised representations for audio-visual speech by masking multistream video inputs and automatically predicting discovered multimodal hidden units, refining them iteratively. We utilize a deduplication method to reduce embedded visual features by using visual speech units. Through deduplication and the proposed Low-Rank Adaptation (LoRA), VinaLLaMA, a large language model specifically designed for Vietnamese, can be fine-tuned efficiently. The performance will be evaluated on a separate test dataset, with metrics including accuracy and processing speed. We will compare the performance of the integrated audio-visual ASR system with systems using only audio or visual data to determine the specific improvements and benefits of combining both types of data. The experimental results will be analyzed to gain a better understanding of the AV-ASR system's performance under various conditions and to identify factors influencing accuracy and efficiency.

- **Objective 2:** Design and implement a versatile transcription system capable of generating accurate transcripts from auditory and visual input modalities, leveraging the proposed ASR model. We will develop a demo version of the AV-ASR system on a web platform, allowing users to easily access and utilize functions such as recording, uploading videos, and receiving text output. This will enable users to directly experience the system's features and evaluate the effectiveness of combining audio and visual data. User feedback will be collected to refine and improve the system before its official deployment.

In summary, this project involves researching and developing an automatic speech recognition system that integrates audio and visual data on Vietnamese Automatic Speech Recognition datasets. Specifically, the project's contributions include:

1. Experiment with parameters and optimize the ViAVSP-LLM system on the VASR dataset.
2. Compare and analyze the performance of these experiments.
3. Provide detailed insights and recommendations for future research and development efforts in the field of automatic speech recognition (ASR), with the aim of improving the quality and efficiency of speech recognition systems.
4. Develop a Vietnamese speech recognition system in the form of a demo.

The structure of this report is as follows: Chapter 2, we delve into existing literature on audio visual speech recognition, and integration of LLMs into speech models. In Chapter 3 provides an overview of the project. Details regarding data preprocessing, the architecture of the approaches, implementation setup, and metric evaluation are covered in Chapter 4. Chapter 5 presents both quantitative and qualitative results. In Chapter 6, we discuss these findings, offer insights, limitations and propose recommendations for future research. Finally, Chapter 7 wraps up the report. Additionally, Appendix A contains information of the demo for this thesis project. Appendix B includes supplementary materials such as source code, datasets, and model training results.

Chapter 2

RELATED WORKS

The Related Works chapter provides an overview of previous studies in the field of Automatic Speech Recognition (ASR) using audio and visual data, with a particular focus on research related to Vietnamese. Additionally, it examines the integration of large language models (LLMs) into ASR systems to enhance system performance and contextual understanding. The goal of this chapter is to offer a comprehensive view of advancements in this field, forming a foundation for further research and development.

2.1 Datasets

One significant obstacle impeding the progress of Vietnamese Automatic Speech Recognition (ASR) utilizing audio and visual data is the lack of sufficiently large and diverse Vietnamese audio-visual speech recognition (AV-ASR) datasets. Initial datasets primarily focused on the English language, with limited sample sizes and vocabulary. For instance, datasets like AVLetters [24], and CUAVE [25] contain fewer than 50 speakers and have restricted vocabulary. AVLetters and CUAVE include utterances of the English alphabet and the digits 0-9. Spanning 26 classes representing the letters A-Z, AVletters contains 780 examples of 10 people saying each of the letters 3 times. The CUAVE dataset has 10 classes (digits 0–9) and contains videos of 36 people saying each of the digits 5 times, so in total there are 180 examples per digit. Previous datasets such as OuluVS2 [26] attempted to expand the vocabulary and provide multi-view videos from different angles. The OuluVS2 contains 52 speakers saying 10 utterances, 3 times each, so in total there are 156 examples per utterance. However, they still fell short in reflecting real-world speech with unrestricted vocabulary. Recent efforts to develop larger and more natural speech datasets have been made with LRW [27], LRS2-BBC [28], LRS3-TED [29], and MV-LRS [30], primarily focusing on the English language. Meanwhile, datasets like CI-AVSR [31], LRW-1000 [32], GLips [33], and MISP2021 [34] have started to utilize automated processes to collect and label non-English speech data, including Cantonese, Mandarin, and German. However, these efforts remain limited and do not adequately meet the requirements for AV-ASR research and development for Vietnamese. To address the challenges of speech recognition in the context of the Vietnamese language, with its complex and diverse phonetic system, Thinh Duong and colleagues recently introduced VASR, a large-scale audio-visual dataset of Vietnamese speakers. VASR boasts a vocabulary of over 7,500 words, capturing more than one thousand hours of recorded speech from over 400 Vietnamese speakers.

2.2 Models

2.2.1 Audio-Visual Speech Recognition

The strong correlation between video and audio modalities provides an effective means for self-supervised learning on videos, a topic that has been explored in numerous prior works and remains an active area of research. Auto-AVSR focuses on audio-visual speech recognition with automatic labels, reducing the labor-intensive manual labeling process and enhancing model performance [35]. By utilizing automatic transcriptions of unlabeled datasets, Auto-AVSR's researchers expanded the training dataset size and enabled the model to achieve state-of-the-art performance on the LRS2 and LRS3 datasets. This demonstrates that increasing the training dataset size can reduce Word Error Rate (WER) even when using noisy transcriptions. Another approach to self-supervised learning for combined audio and visual speech representations is RAVEn [36]. The results show that RAVEn outperforms other self-supervised methods for visual speech recognition (VSR) and, when combined with self-training, it surpasses semi-supervised methods using large amounts of data. This indicates that learning robust speech representations entirely from raw video and audio is feasible. Additionally, some studies, such as WLAS [30], aim to recognize phrases and sentences from a speaking face, with or without audio, in open environments. The WLAS model trained on the LRS dataset with over 100,000 natural sentences from British television outperforms previous works and even surpasses a professional lip-reader on BBC television videos, demonstrating that visual information enhances speech recognition performance even in the presence of audio. The LCB-net work [37] proposes a long-context bias network for audio-visual speech recognition (AVSR) to leverage long-context information in videos. This model applies a dual-encoder architecture to model audio bias and long-context simultaneously, using a bias prediction module with binary cross-entropy (BCE) loss to identify biased phrases, improving the model's generalization and robustness.

Our AVSR model is based on research into Audio-Visual Hidden Unit BERT (AV-HuBERT) [38], a self-supervised representation learning framework for audio-visual speech, masking multi-stream video inputs and predicting automatically discovered multimodal hidden units refined iteratively. AV-HuBERT is trained with a masked prediction loss similar to AV-BERT [39], forcing the model to learn structure in multimodal inputs and demonstrating better resilience to clustering tasks than unmasked cluster prediction. While AV-BERT focuses on learning multimodal environment embeddings at the utterance level, serving as a global context for ASR, AV-HuBERT learns audio-visual speech representations at the frame level and pre-train a model that can be fine-tuned for downstream tasks with any modality. Overall, training and optimizing AV-ASR models still face numerous challenges. The process is more complex and costly compared to models using only audio or visual data. Specifically for Vietnamese, research on integrating audio and visual data in ASR systems remains limited. The lack of high-quality data for Vietnamese language are significant barriers in this field.

2.2.2 Integration of LLMs into Speech Models

While leveraging the extensive language knowledge and contextual understanding inherent in LLMs, several studies have sought to seamlessly integrate text-based knowledge with other modalities. For example, researchers introduced the VSP-LLM framework [40], which integrates LLMs into visual speech processing to enhance context modeling capabilities. This ability is particularly crucial for ambiguous lip movements, such as homophones, where understanding the context can help distinguish between words that have the same lip movements but produce different sounds. VSP-LLM is designed to perform multitasking in both visual speech

recognition and translation, where input prompts guide the type of task to be executed. This approach has demonstrated the superior effectiveness of LLMs in translating lip movements on a small dataset. AudioPaLM [41], a large language model that integrates speech understanding and generation capabilities, combines a text-based language model (PaLM-2) with an audio-based language model (AudioLM) into a multimodal architecture capable of processing and generating both text and speech. AudioPaLM inherits the ability to retain information about tone and speaker identity from AudioLM and linguistic knowledge from PaLM-2, enhancing performance in speech translation tasks. Research has shown that initializing AudioPaLM with weights from a text-based large language model can enhance speech processing capabilities, leveraging extensive text training data to support speech tasks. Fathullah et al. expanded the capabilities of LLMs by directly attaching a small audio encoder to perform speech recognition [42]. By aligning a sequence of audio embeddings with text token embeddings, the LLM can be transformed into an automatic speech recognition (ASR) system. Experiments on the Multilingual LibriSpeech (MLS) dataset demonstrated that integrating a conformer encoder into LLaMA-7B allowed the model to outperform monolingual models and perform well in multilingual speech recognition, even though LLaMA was primarily trained on English text. Furthermore, Speech-LLaMA [43], a novel approach that integrates acoustic information into text-based LLMs, uses Connectionist Temporal Classification and a simple audio encoder to map compressed acoustic features into the continuous semantic space of LLMs. The study showed significant improvements over baseline models in multilingual speech translation tasks, confirming the potential of decoder-only architectures for speech-to-text conversion tasks. The integration of LLMs into speech systems for the Vietnamese language remains scarce. VinaLLaMA [44] is a large language model for Vietnamese, developed based on LLaMA-2 with the addition of 800 billion training tokens. VinaLLaMA not only exhibits fluency in Vietnamese but also possesses a deep understanding of Vietnamese culture, making it a highly localized model. Therefore, we will utilize VinaLLaMA as the LLM to integrate into our speech system.

Chapter 3

PROJECT MANAGEMENT PLAN

The Project Management Plan chapter serves as a comprehensive roadmap for the successful execution of this thesis project. Within this chapter, an overview of the project will be provided, followed by detailed information regarding the project's objectives, scope, and schedule.

3.1 Overview

This is the graduation project of a student majoring in Artificial Intelligence at FPT University, Summer 2024 semester. The project information includes:

- Information about supervisors:

	Full name	**Email**	**Title**
Supervisor 1	Le Thanh Hai	hailt56@fe.edu.vn	Dr.

Table 1. **Supervisor information.**

- Information about the project team:

	Full name	**Student ID**	**Email**	**Role**
Student 1	Duong Tan Thinh	SE160970	thinhdtse160970@fpt.edu.vn	Leader
Student 2	Nguyen Van Minh	SE162009	minhnvse162009@fpt.edu.vn	Member
Student 3	Pham Hong Duyen Khanh	SE160477	khanhphdse160477@fpt.edu.vn	Member

Table 2. **Student information.**

3.2 Project Scope and Objectives

The objective of this project is to develop an efficient Automatic Speech Recognition (ASR) system for Vietnamese by integrating audio and visual data. This integration aims to overcome the limitations of current ASR systems by incorporating environmental information from speakers, facial features, and lip movements to improve accuracy and performance in noisy environments. Additionally, a Minimum Viable Product (MVP) has been developed as part of this graduation project to demonstrate the practical application of the ASR system integrating audio and visual data in fields such as education, healthcare, and customer service.

Through these efforts, the project contributes to advancing ASR technology for Vietnamese, enhancing its practical applications, and exploring new avenues for research in multimedia data integration and its utilization.

3.3 Project Schedule

Table 3 provides an overview of the timeline for this project. The schedule outlines the estimated durations for each phase, which may be subject to adjustments based on project progress and requirements.

Task name	**Priority**	**Start date**	**End date**	**Status**
Find documents	High	01/05/2024	07/05/2024	Done
Review papers	Medium	05/05/2024	14/05/2024	Done
Review and analyze dataset	Low	15/05/2024	18/05/2024	Done
Research on methods	High	15/05/2024	20/05/2024	Done
Prepare data and source code	High	21/05/2024	04/06/2024	Done
Experiment models	High	05/06/2024	15/07/2024	Done
Create Web demo	High	16/07/2024	31/07/2024	Done
Write report	High	01/08/2024	10/08/2024	Done
Review Report	Medium	11/08/2024	20/08/2024	Done

Table 3. **Project schedule.** Overview of the timeline for the Vietnamese Automatic Speech Recognition Utilizing Audio and Visual Data thesis

Chapter 4

MATERIALS AND METHODS

The Materials and Methods section provides a comprehensive blueprint of the research methodology employed to achieve the study's objectives. By meticulously detailing data characteristics, preprocessing techniques, architectural frameworks, evaluation metrics, and experimental procedures, this section ensures the research's transparency, reproducibility, and rigor. This explicit documentation facilitates a deep understanding of the research process, enabling critical evaluation and potential replication by the broader scientific community, thereby advancing knowledge in the field.

4.1 Data

The VASR dataset, utilized in this study, is a comprehensive multi-modal resource specifically designed for speech recognition research, with a particular focus on the Vietnamese language. The following sections will provide detailed insights into the dataset's properties, the methods used for data splitting, and the sampling techniques employed. This thorough examination will underscore the dataset's robustness and suitability for advancing the field of speech recognition.

4.1.1 Data Properties

An overview of the dataset is provided in Table 4. This dataset comprises over 1,000 hours of labeled speaker videos sourced from more than 400 social media channels, covering a wide range of topics. The dataset includes a diverse array of speakers in terms of age, gender, and makeup, ensuring that models trained on this data are robust across different demographic groups. Additionally, the variety of speakers and camera setups makes VASR a comprehensive resource for advancing speech recognition research and development. The following subsections will delve into the major aspects of the dataset in detail.

Transcript

The VASR dataset provides Vietnamese transcripts for each video clip. Despite each video being only 3 seconds long, they contain, on average, 13.86 words per instance, offering ample context for the model's comprehension. As shown in Figure 1, the distribution of text length follows a normal distribution, ensuring that the dataset accurately reflects real-world conversational speech patterns. This normal distribution of text length highlights the dataset's ability to capture natural language variability, making it a reliable resource for developing robust

Video frame rate (fps)	25
Audio sampling rate (Hz)	16,000
Resolution (pixels)	96x96
Video duration (s)	3
Number of hours	1,045
Average number of words per example	13.86
Vocabulary	7,528

Table 4. **Dataset Overview.** Vocabulary refers to unique words in a dataset.

speech recognition models.

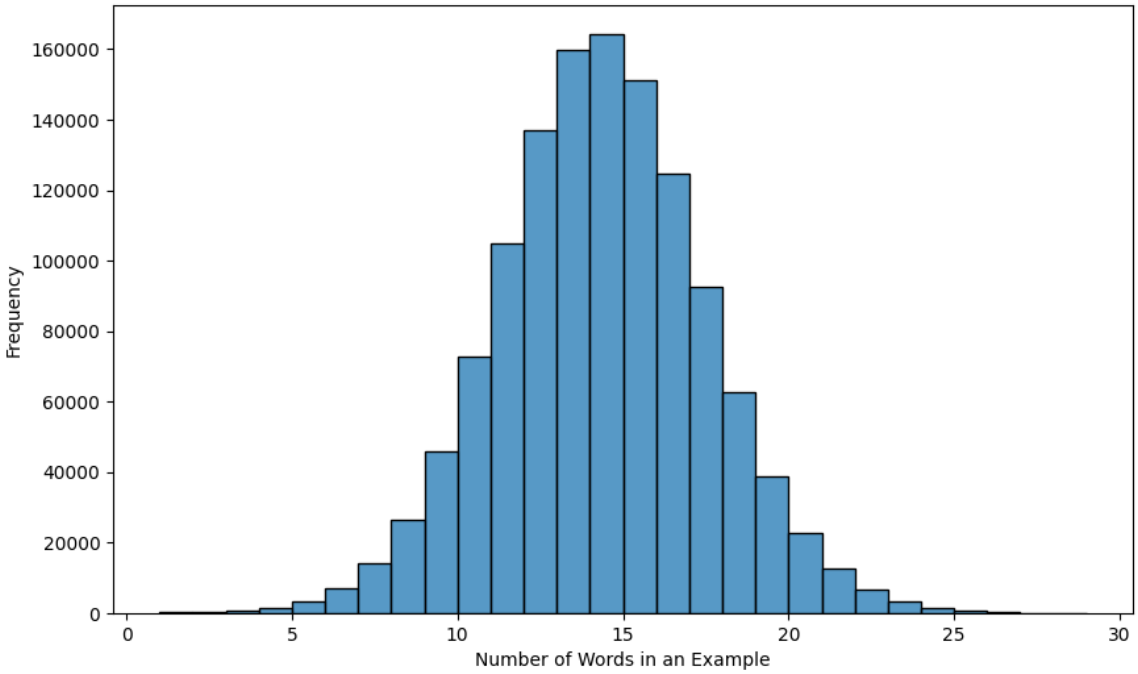


Figure 1. **Distribution of the Number of Words per Data Instance in the Entire Dataset.**

Topics

The dataset also encompasses a broad spectrum of topics, covering at least 11 distinct areas, as shown in Figure 2. This diversity in subject matter enhances the dataset's ability to mirror real-world conversational speech. By including such a wide range of topics, the dataset supports the development of a versatile speech recognition model, ensuring that the model can generalize effectively across different domains and contexts.

Camera Views

In terms of camera setups, the dataset captures different camera angles, as illustrated in Figure 3. Additionally, the Figure 3 also shows that various lighting conditions are available in the dataset. The variability of camera setups increases the complexity of the dataset and introduces challenges that help models achieve robustness in real-life speech recognition scenarios.

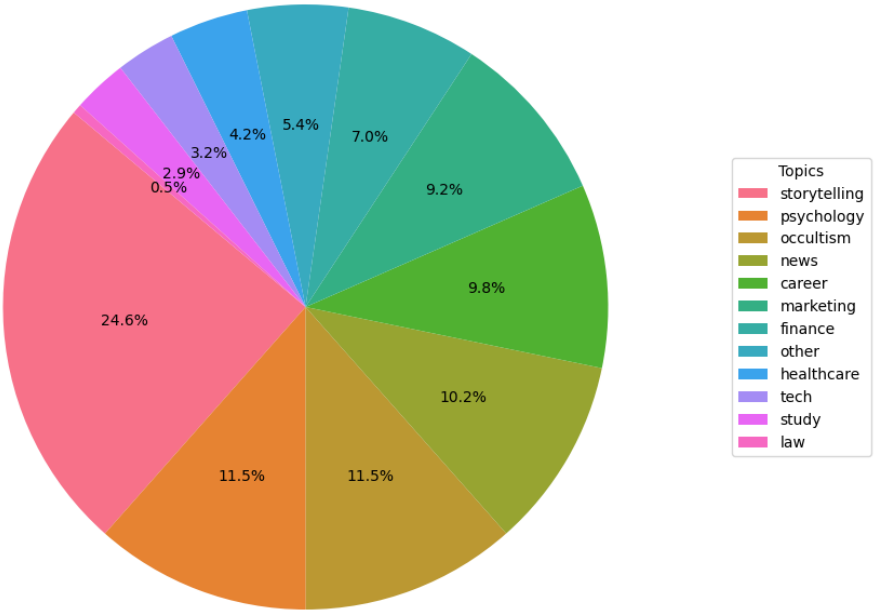


Figure 2. **Distribution of Topics.**

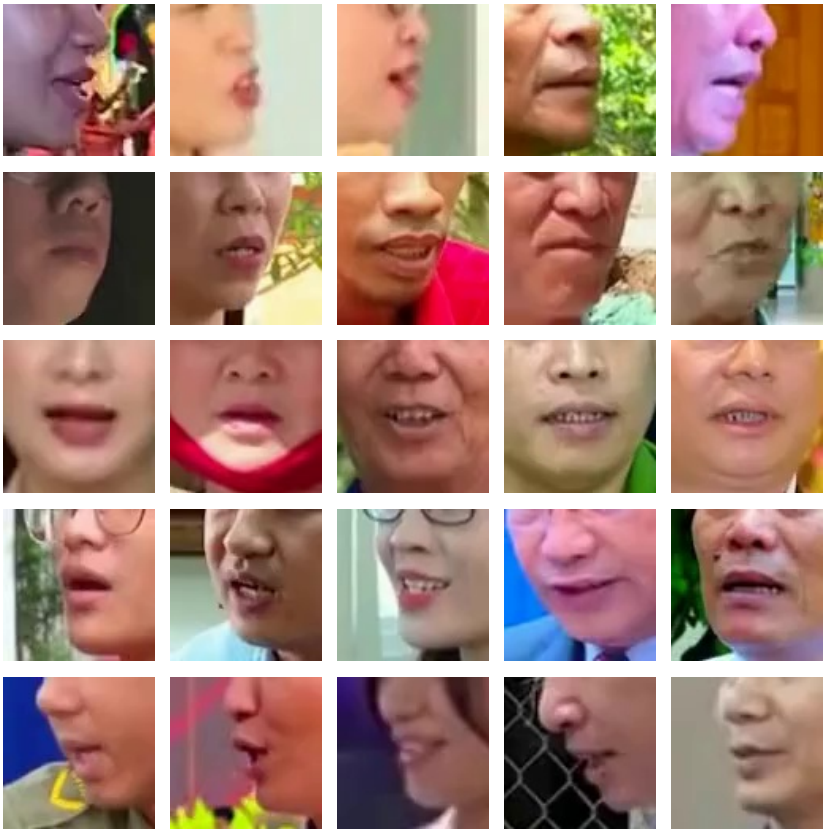


Figure 3. **Examples of Different Camera Views.** Top row: right profile; 2nd row: right three-quarter; 3rd row: frontal; 4th row: left three-quarter; Bottom row: left profile.

4.1.2 Data Splits

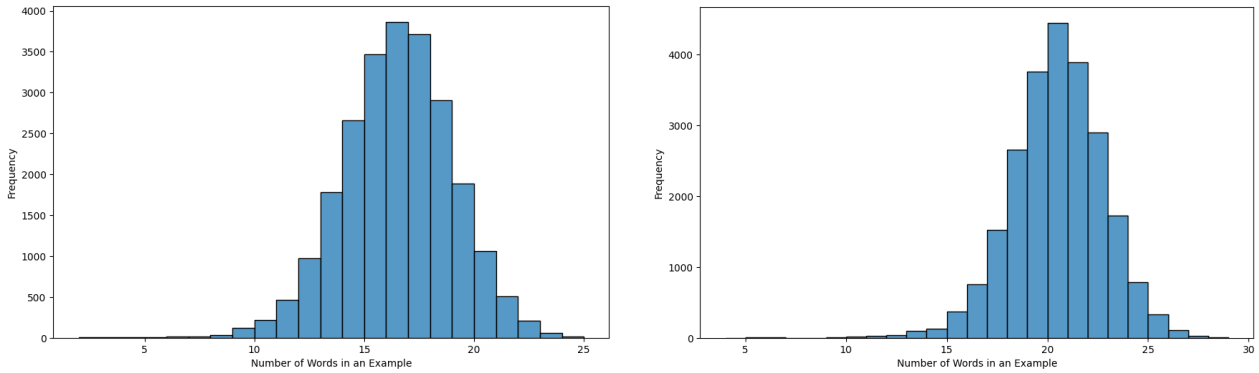
The dataset is divided into training, validation, and test splits with ratios of 96%, 2%, and 2%, respectively, as detailed in Table 5. This distribution is chosen for several reasons. First, to effectively transcribe speech from unseen speakers, the model requires a training set

	Training	Validation	Test
Number of samples	1,206,779	23,990	23,617
Number of hours	1,005	20	20
Average number of words	13.69	16.02	19.98
Vocabulary	7,497	3,704	3,939
Unique vocabulary	2,947	14	17

Table 5. **Statistics of Splits.** Unique vocabulary refers to words that appear exclusively in a specific subset.

with a wide variety of individuals. Second, given the computational intensity of the evaluation process, increasing the size of the validation and test sets would significantly extend evaluation time during training. Moreover, following the advice on data splits from Dr. Andrew Ng [45], these percentages are reasonable for estimating model performance given the training set contains over a million examples.

Additionally, the validation and test sets are made speaker-exclusive, meaning that speakers in these sets do not appear in the training set or in each other, as referenced in [46]. Although this strategy results in slightly different distributions of the number of words per data instance in the validation and test sets (Figure 4), it ensures the robustness of the model in recognizing speech from unseen speakers.



(a) (b)

Figure 4. **Distributions of the Number of Words per Data Instance in Evaluation Sets.** Figure (a) represents the complete validation set, while figures (b) corresponds to the test set.

4.1.3 Data Sampling

The substantial size of the training set, the computational demands of the model, and the time constraints of this thesis present significant challenges, limiting the number of experiments that can be conducted. To address these issues, two subsets containing 100 and 200 hours of data—equivalent to approximately 10% and 20% of the full training set, respectively—have been extracted using stratified sampling method, as illustrated in Figure 5.

The primary objective of this sampling method is to maintain the original distribution of speakers. A critical factor in achieving this goal is the video source. Since each video source originates from a single social media channel and is typically speaker-exclusive, stratifying by

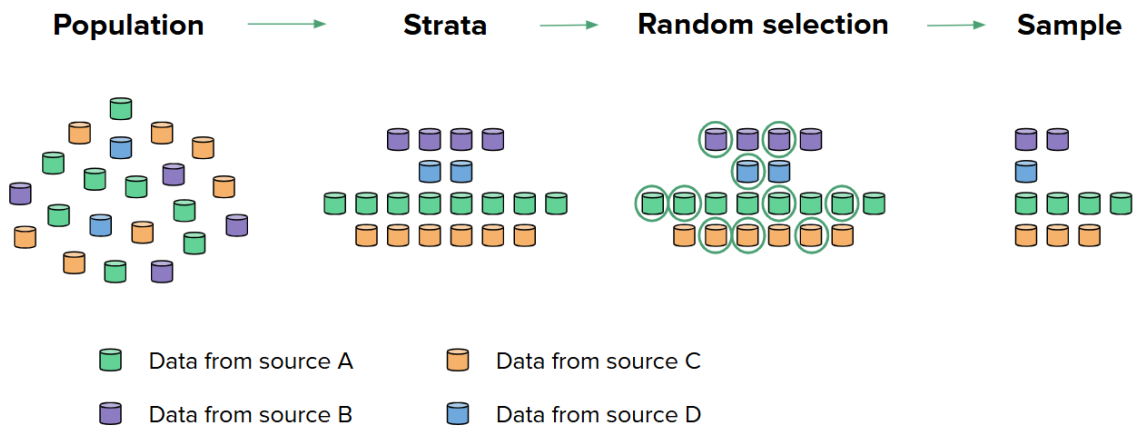


Figure 5. **Example of Stratified Sampling Method.** In this example, a sample comprising 50% of the population is selected.

video source ensures the preservation of speaker diversity. This approach also helps retain various speaker-related characteristics, such as speaking speed, dialects, and video quality, thereby ensuring that these attributes are consistently represented within the subsets.

As detailed in Table 6, although the 100-hour and 200-hour subsets represent only 10% and 20% of the entire training set, they still cover over 76% and 85% of the total vocabulary found in the full dataset, respectively. Moreover, the analysis in Figure 6 and Table 6 highlights that the distributions of word counts and the average number of words per data instance are remarkably consistent across the 1,000-hour, 100-hour, and 200-hour sets. Given the uniform video length of 3 seconds, the word count per instance also serves as an indicator of speaking speed, a crucial speaker-related characteristic. By maintaining speaker diversity through this sampling method, we preserve essential speaker attributes, such as speaking speed, which ensures the robustness and validity of our experimental outcomes. These findings confirm that experiments conducted with these subsets effectively reflect the key properties of the original full training set.

	100-hour	**200-hour**	**1,000-hour**
Number of samples	120,686	241,351	1,206,779
Number of hours	100	201	1,005
Average number of words	13.69	13.69	13.69
Vocabulary	5,725	6,369	7,497

Table 6. **Statistics of Training Subsets.** Unique vocabulary refers to words that appear exclusively in a specific subset.

4.1.4 Data Preprocessing

The preprocessing steps in this study build upon the methodologies established in prior works [38, 40]. For each video clip, it is recommended that the original video sources be resampled to 25 frames per second (fps). This frame rate offers sufficient temporal resolution to capture essential motion and scene changes, while maintaining computational efficiency. Human vision perceives motion smoothly at this rate, making it suitable for most applications. After resampling, the sequence of frames is processed by converting the images to grayscale. Grayscale images, with only one channel compared to the three channels in RGB images, reduce

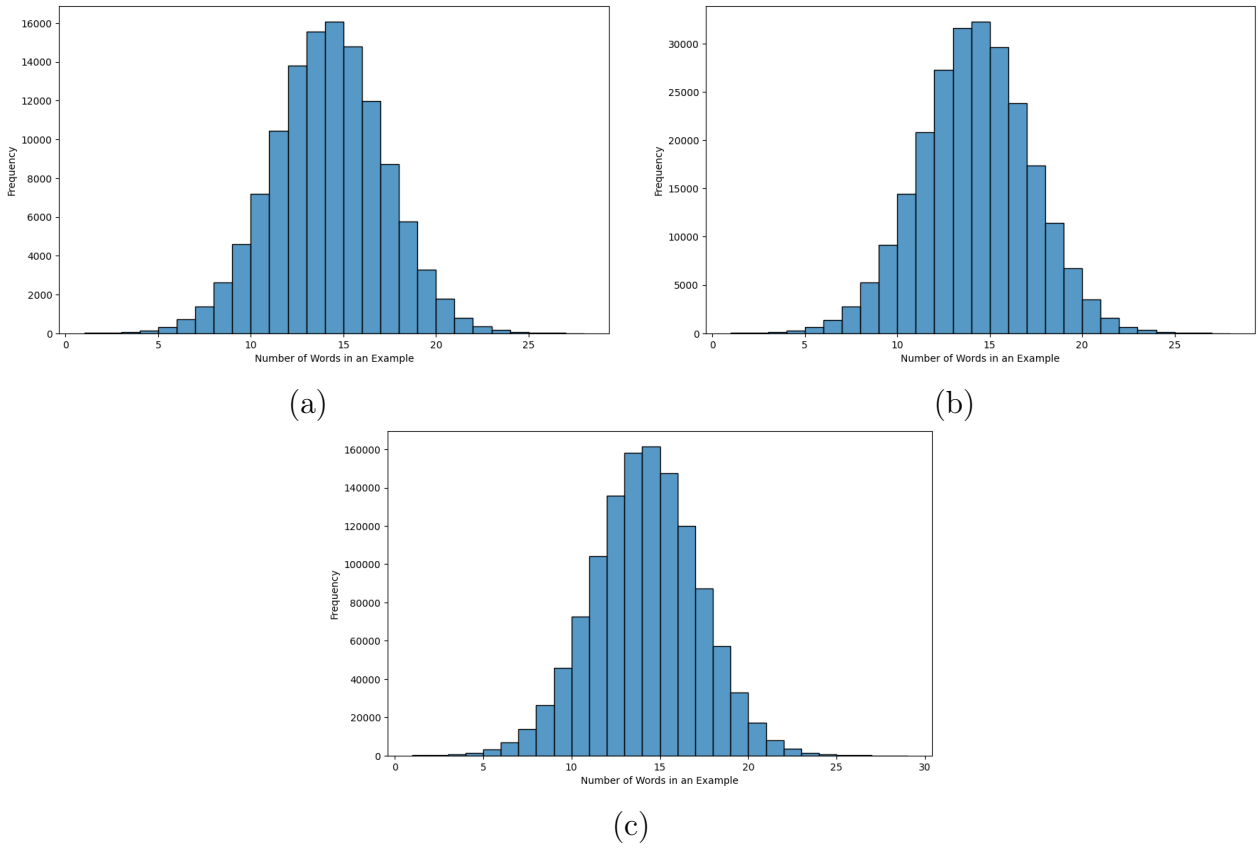


Figure 6. **Distributions of the Number of Words per Data Instance in training sets.** Figure (c) represents the complete training set (1,000 hours), while figures (a) and (b) correspond to the 200-hour and 100-hour training sets, respectively.

the dimensionality of the data, simplifying the model and decreasing computational complexity. This streamlining allows for faster training while still preserving critical information such as intensity and brightness, which are crucial for tasks like edge detection, texture analysis, and certain types of object recognition. During training, frames are randomly cropped to 88 $\times$ 88 pixels from the region of interest (ROI) and horizontally flipped with a probability of 0.5. For testing, an 88 $\times$ 88 pixel ROI is center-cropped without horizontal flipping.

For the associated audio data, a resampling rate of 16,000 Hz is utilized, which is a standard sampling rate for speech-related tasks in both traditional and deep learning models. Since the majority of speech information lies below 8 kHz, resampling to 16 kHz ensures that all essential information is retained, as per the Nyquist theorem. This reduces the amount of data while preserving critical speech information, thereby speeding up both training and inference. When processing the audio signal, which is a sequence of discrete numerical values representing the amplitude of the audio signal over time, it is essential to synchronize the audio with the video. The audio sequence, typically longer than the video frame sequence due to the difference in sampling rates, is processed by extracting 26-dimensional log filter-bank energy features at a 10 ms stride from the raw waveform. These features are used as input to the model, providing a time-frequency representation that is compact, robust, and perceptually relevant. The log Mel-frequency filter bank (logfbank) is particularly effective because it mimics the human ear's sensitivity to different frequencies by placing more filters in the lower frequency range and fewer in the higher range. This approach ensures that the audio representation is both noise-robust and aligned with human perception. Finally, to synchronize the audio with the video frames, the acoustic frames are stacked in groups of four to match the 25Hz sampling rate of the video.

4.2 Methods

Building upon prior research and various existing approaches to the Speech-to-Text problem [47], such as Audio-Based Speech Recognition [48], which relies on acoustic signals, and Lip Reading [49], which extracts information from visual frame sequences, we have developed a novel model named ViAVSP-LLM, base on existing model VSP-LLM [40] which uses visual in video data and integrate LLM to solve Speech-to-Text for English. This model is designed to leverage the strengths of both audio and visual data to improve transcription accuracy.

Our model incorporates two main components: the AV-HuBERT [38], which functions as the audio-visual encoder, and VinaLLaMA [44], which acts as the decoder. The AV-HuBERT is responsible for processing and encoding the multimodal input, capturing both audio and visual cues, while VinaLLaMA decodes this rich representation into text. To further enhance the model's performance, we introduce a deduplication step that efficiently reduces duplicated information within the encoded features, ensuring a more streamlined and accurate transcription process.

The overall architecture of ViAVSP-LLM is illustrated in Figure 7, providing a visual representation of how the different components interact. The following sections will delve into a detailed examination of this model, exploring how each part contributes to solving the Speech-to-Text problem and highlighting the innovative aspects of our approach.

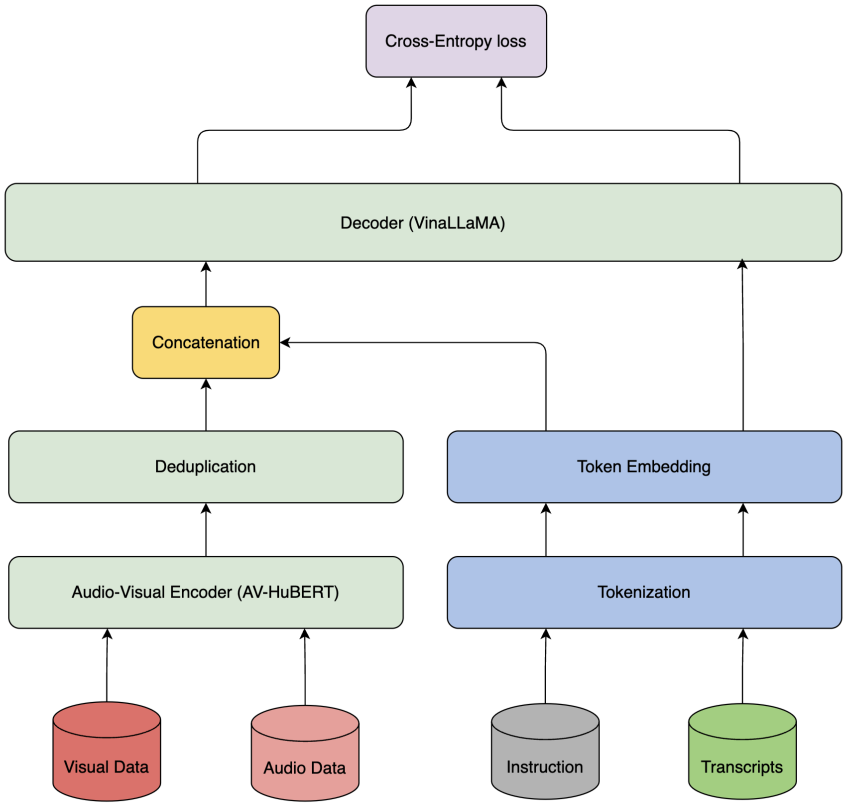


Figure 7. **Overview of ViAVSP-LLM architecture.**

4.2.1 Audio-Visual Encoder

Our primary goal is to leverage the advanced context modeling capabilities of large language models (LLMs) in audio-visual speech recognition. A critical aspect of this approach

is ensuring that the input video is represented in a way that aligns seamlessly with the linguistic content, enabling a robust association between the audio-visual input and the textual space of the pre-trained LLM. Drawing inspiration from recent advancements in self-supervised speech models, which have demonstrated strong correlations between learned representations and speech elements such as phonemes [50, 51], we extend the VSP-LLM framework [40] by employing AV-HuBERT [38] as our core audio-visual encoder. The detailed architecture of the audio-visual model is illustrated in Figure 8. The processing workflow of AV-HuBERT encompasses several steps, including feature extraction, modalities fusion, and encoding the audio-visual features.

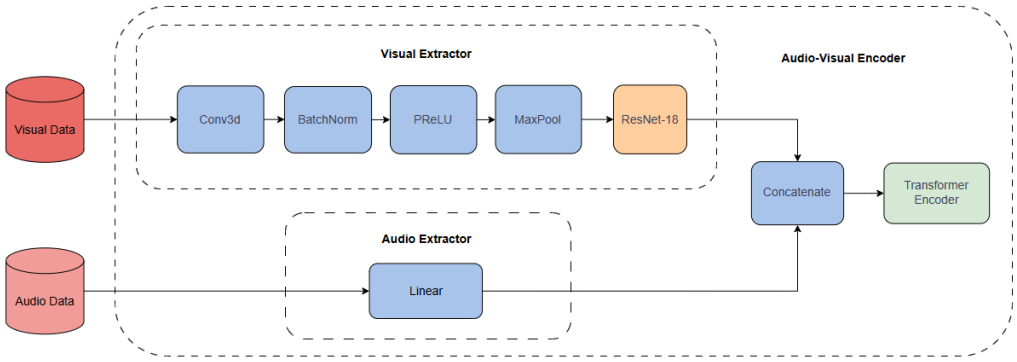


Figure 8. **Architecture of Audio-Visual Encoder.**

4.2.1.1 Audio Extractor

Once the audio signal has been transformed into this time-frequency representation, it is passed into a component known as a simple audio extractor. The audio extractor is designed as a linear projection layer, which serves as a basic, yet crucial, part of the model. As illustrated in Figure 8, the linear projection layer maps the input features into a lower-dimensional space. The primary reason for this choice is to prevent the model from becoming overly reliant on the audio stream alone [38].

In our approach, each time-step of the audio signal is represented by a vector with a dimension of 1,024. This dimension is chosen to be consistent with the dimensions used in the visual modality. The reason for using this particular dimension size is twofold: it is large enough to capture the important information necessary for understanding speech, but not so large that it causes excessive memory usage or computational demands. Similarly, in the visual branch of our model, this dimension size ensures that we capture the primary information while keeping resource consumption manageable.

4.2.1.2 Visual Extractor

The sequence of frames, after being converted to grayscale, is processed through a visual extractor as illustrated in Figure 8. To effectively extract visual features, we employ a Convolution 3D layer [52]. Equation (1) illustrates how the Convolution 3D layer utilizes the continuity from a sequence of frames to extract features from a sequence of frames:

$$O(x,y,z) = \sum_{i=0}^{D_f-1} \sum_{j=0}^{H_f-1} \sum_{k=0}^{W_f-1} X(x+i, y+j, z+k) \cdot F(i,j,k) \tag{1}$$

where $O(x, y, z)$, X and F represent the output feature at position (x, y, z), the input sequence of frames, and the 3D convolutional kernel of size (i, j, k) respectively. Unlike the traditional Convolution 2D layer [53], which scans over the two dimensions of a single image independently, the Convolution 3D layer processes three dimensions: height, width, and temporal depth. This allows the layer to capture the temporal relationships between consecutive frames, which is critical for understanding the context and nuances of spoken words in a video sequence.

The feature map, produced after the Convolution 3D, undergoes a series of common processing steps: BatchNorm3d [54], PReLU [55], and MaxPool3d [56]. BatchNorm3d normalizes the feature map to improve training stability and convergence, PReLU introduces non-linearity to the network, and MaxPool3d reduces the spatial dimensions while retaining important features.

After extracting the relationships between frames, these features are fed into a pre-trained classical Convolutional Neural Network(CNN), ResNet-18 [57]. ResNet-18 is a CNN model consisting of 18 layers, which strikes a balance between depth and complexity, making it deep enough to learn intricate features without being excessively deep. Compared to models like VGG-16 and VGG-19 [58], which are deeper and achieve good performance, ResNet-18 has significantly fewer parameters and requires less computational resources. Despite this, ResNet-18 achieves similar or better performance due to its efficient architecture and the use of residual connections, making it a more practical choice for tasks requiring robust feature extraction with lower computational cost. ResNet-18 also utilizes residual connections, also known as skip connections. The mathematical formula of this operation is expressed in Equation (2):

$$h' = f(h, W) + h \tag{2}$$

where h' is the output produced by the current residual layer, f represents the forward operations of the layer, W the weights of the layer, and h is the input or the previous output layer fed into the current layer. The skip connections in ResNet-18 not only facilitate easier training but also enhance the model's ability to generalize better on unseen data. These connections help overcome the vanishing gradient problem by preserving the gradient flow throughout the network. This makes it easier to train very deep networks and allows the extraction of relevant features through multiple layers.

The output of this feature extractor is a sequence of vectors, where each time-step corresponds to the relevant features extracted from a single frame. In our configuration, the dimension of each vector is set to 1,024. This dimensional number is chosen to be sufficiently large to capture the essential information from each frame while also being efficient in terms of memory and computational resources. This balance ensures that the primary information is preserved without imposing excessive computational overhead.

4.2.1.3 Transformer Encoder

The combined visual and acoustic features are processed using an encoder module based on the Transformer architecture [59]. This module is designed with a stack of 24 identical Transformer blocks. Each of these blocks consists of two main sub-layers, which work together to process and encode the input features.

To effectively encode the combined features from the visual and audio inputs, the Transformer architecture utilizes a mechanism known as Multi-Head Attention (MHA) [59]. The

MHA mechanism is particularly effective for handling sequences of continuous frames, which is common in video and speech processing. It operates by stacking multiple parallel attention mechanisms, each referred to as an Attention Head. The number of these attention heads is denoted by h, as shown in Figure 9.

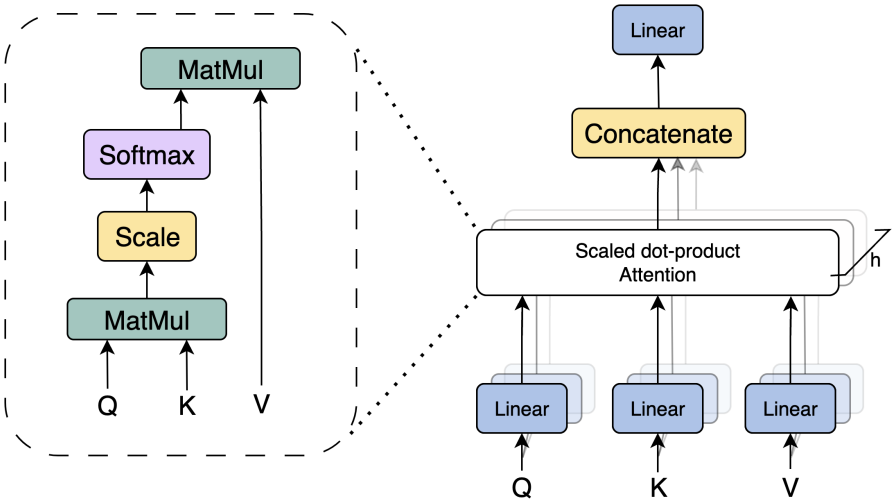


Figure 9. **Structure of Multi-Head Attention mechanism**.

Each head in MHA operates with its own distinct set of learned parameters, allowing it to apply different mappings to the same queries, keys, and values. As a result, each head processes the input data in its unique way, resulting in different perspectives or features being extracted. Mathematically, the output of each attention head is represented as follows:

$$h_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{3}$$

where:

- Q, K, and V are the matrices representing the queries, keys, and values, respectively.
- W_i^Q, W_i^K, and W_i^V are the projection matrices specific to the i^{th} attention head, used to transform the queries, keys, and values for that head.

The core operation within each Attention Head is the scaled dot-product attention mechanism, which is mathematically expressed as follows:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{4}$$

where Q, K, V and d are query matrix, key matrix, value matrix and dimension of the key vectors respectively. Equation (4) involves computing the attention scores by taking the dot product of the query and key matrices, scaling it by the square root of the key dimension, applying the $softmax$ function to obtain the attention weights, and then using these weights to generate the final output by multiplying with the value matrix. Since this is a form of self-attention, the query, key, and value matrices all come from the same input tensor, allowing the model to assess and encode the context of each time-step relative to all other time-steps in the sequence.

The outputs of all Attention Heads are then concatenated together to form a combined representation:

$$MHA(Q, K, V) = concatenate(h_1, h_2, ..., h_h)W^O \tag{5}$$

In Equation 5, $h_1, h_2, ..., h_h$ represents the concatenated outputs from all h attention headsm while W^O is a final linear projection matrix that transforms the concatenated outputs into the desired dimension of the feature vectors.

The MHA has several advantages when working with sequences of frames. It allows the model to simultaneously focus on different parts of the input sequence. By doing so, it can capture various aspects of the temporal (time-based) and spatial (space-based) relationships within the data, leading to a richer and more nuanced representation of the data. It also enhances the model's ability to capture long-range dependencies (relationships between distant time steps) and subtle contextual information.

After the MHA mechanism produces its output, this data is fed into the second sub-layer of the Transformer block. This sub-layer consists of a sequence of operations:

1. **Linear Layer** A linear layer that applies a linear transformation to the input data.
2. **Activation Function** A non-linear activation function that introduces non-linearity into the model, enhancing its ability to learn complex patterns. In our model, that function is ReLU [60].
3. **Linear Layer** Another linear layer is utilized to learn the mapping from audio-visual to the LLM's representation.

To facilitate the training process and improve the stability of the learning, each of these sub-layers is surrounded by a residual connection and followed by layer normalization. The residual connection helps in mitigating issues related to vanishing gradients by allowing gradients to flow through the network more easily. Layer normalization standardizes the inputs to each sub-layer, which helps in stabilizing and accelerating training.

This process is repeated across 24 Transformer blocks, each refining the features through these layers. By the end of this series of blocks, the model generates a final feature representation that encapsulates relevant information from both visual and audio modalities. This comprehensive feature representation is crucial for effectively understanding and integrating the information from the two different types of data.

4.2.2 Deduplication

In many cases, the length of the text data is much shorter compared to the length of the corresponding video. This situation is analogous to what happens in ASR systems, where the duration of spoken input (speech) is typically longer than the length of the transcribed output text. In ASR, spoken language is continuous and lengthy, whereas the text representation is much shorter because it condenses the spoken content into written form. Similarly, when mapping audio-visual speech representations to a text space using a model like AV-HuBERT, the integration output corresponds to the length of the video frames used as input. If this integration output is fed directly into a decoder model, such as a LLM, there would be significant computational challenges. Decoders, especially large ones, require substantial computational

resources when processing long sequences. Thus, directly using the integration output from the video frames without modification would result in a heavy computational burden, making the process inefficient and resource-intensive.

To address this issue, we can exploit the temporal smoothness of the video. In a video, consecutive frames typically contain overlapping or similar information because the video content changes gradually over time. This means that frames close to each other in time are often redundant in terms of the information they provide. For purpose make the processing more efficient, we propose reducing the length of the integration representation before feeding it into the LLM. This can be achieved by aggregating or summarizing the information from consecutive frames to eliminate redundancy and reduce the overall sequence length. By doing so, we can create a more compact representation that captures the essential information while minimizing the computational load on the decoder. This approach allows for a more efficient handling of the data and helps in managing the computational demands associated with large language models.

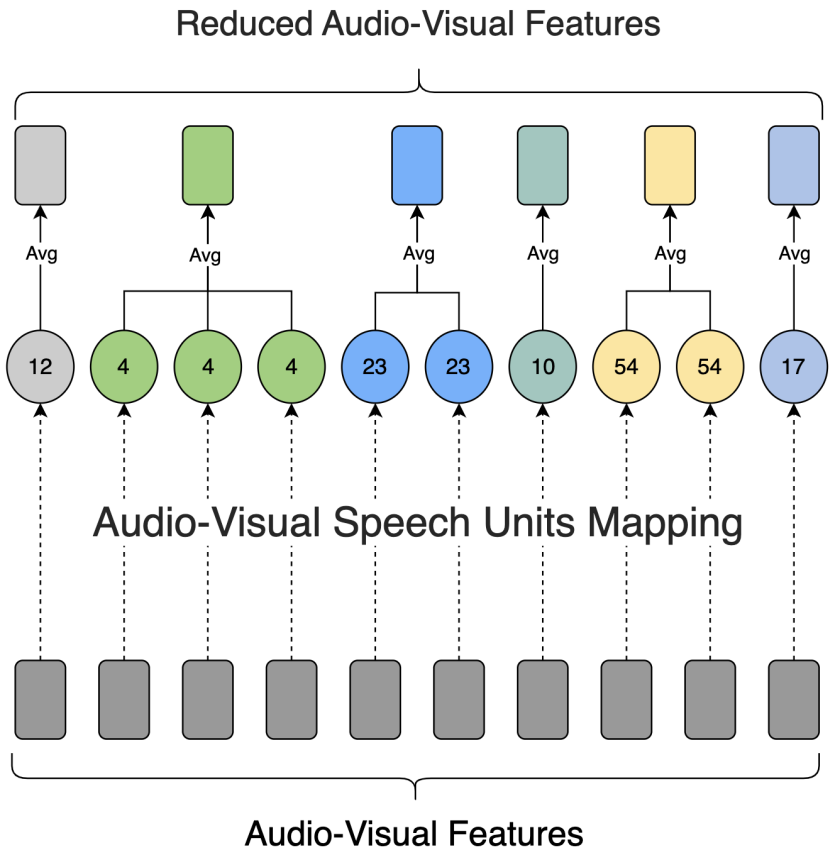


Figure 10. **Overview of Deduplication Process.** The term "Avg" is short for Averaging

The overview of this deduplication process is described as in Figure 10. Extending previous researching in VSP-LLM [40], we use K-Means, an unsupervised model, to group the similar audio-visual features representing contiguous and overlapping video frames into clusters. K-Means is a widely used method for partitioning data into clusters based on feature similarity. It is well-suited for clustering unlabeled features, which means we do not need predefined labels for our data. The K-Means algorithm is chosen because it is both lightweight and efficient, making it suitable for handling large sets of features.

In the initial phase of training, the K-Means algorithm is employed on the training data

to effectively cluster the extracted features. These features are sourced from the 12^{th} layer of a Transformer encoder block. The decision to use the 12^{th} layer, as opposed to deeper layers, is strategic. While deeper layers tend to capture more complex and abstract patterns, they may become too specialized, potentially losing the general, overlapping information critical for clustering contiguous frames. By leveraging the 12^{th} layer, we strike a balance between capturing meaningful representations and preserving the similarity between adjacent frames, which is essential for accurate and coherent clustering.

Once the features are extracted, the K-Means algorithm is employed to cluster those with similar or overlapping information. Each resulting cluster is designated as an Audio-Visual Speech Unit. Representative features for each cluster are then generated by averaging the contiguous features within that cluster. For instance, as illustrated in Figure 10, if the resulting Audio-Visual Speech Units from 10 features are [12, 4, 4, 4, 23, 23, 10, 54, 54, 17], the features at positions 2, 3, and 4 are averaged, as are those at positions 5 and 6, and 8 and 9. This process results in a condensed set of features—in this example, reducing the sequence from 10 to 6 features. This reduction in sequence length is achieved without compromising the integrity of the essential information, thereby optimizing the efficiency and effectiveness of the model.

4.2.3 Decoder (VinaLLaMA)

After the deduplication step, we obtain a streamlined set of audio-visual features, referred to as Reduced Audio-Visual Features. These optimized features are then prepared for subsequent processing by concatenating them with an embedding instruction, creating a unified input that is fed into a decoder built on a Large Language Model (LLM). Building on the approach by Yeo et al. [40], our objective is to achieve comparable or superior results for Vietnamese speech recognition by utilizing VinaLLaMA [44] as the decoder. VinaLLaMA, a state-of-the-art foundation language model, has been specifically adapted to understand and process the nuanced cultural and semantic intricacies of the Vietnamese language, making it exceptionally well-suited for our speech-to-text application.

The core of our model's decoder architecture is composed of a stack of 32 identical Transformer decoder blocks. These blocks share a structural resemblance to Transformer encoder blocks but are distinguished by a few critical differences. The most notable distinction is the inclusion of a Masked Multi-Head Attention (MMHA) layer at the beginning of each decoder block, as depicted in Figure 11. The MMHA layer performs a similar mathematical operation to the standard attention mechanism, as described in Equation 4, but with an added masking step.

$$MaskAttention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}} + mask\right)V \tag{6}$$

Here's a breakdown of this formula:

- **Attention Calculation**: The base calculation for attention is the scaled dot-product, where Q (query), K (key), and V (value) are matrices derived from the input. This base calculation is represented as: $\frac{QK^T}{\sqrt{d}}$
- **Mask**: The addition of the "mask" term modifies the attention scores by ensuring that the model cannot access future time steps. This is achieved by adding a large negative value (effectively $-\infty$) to the attention scores for positions that should be masked out.

This prevents the model from attending to future time steps, ensuring that predictions at any given time-step are based only on past and current information.

- **Softmax Activation**: The modified attention scores are then passed through a softmax activation function, which normalizes these scores so that they sum to one, producing the final attention weights.
- **Weighted Sum**: The final attention weights are used to compute a weighted sum of the values V, generating the output of the attention mechanism.

The mask ensures that when predicting or generating text at a particular time step, the model only considers information from the current or past time steps, not from future ones. This is crucial for tasks like auto-regressive generation, where predictions are made sequentially and must rely only on previously generated content.

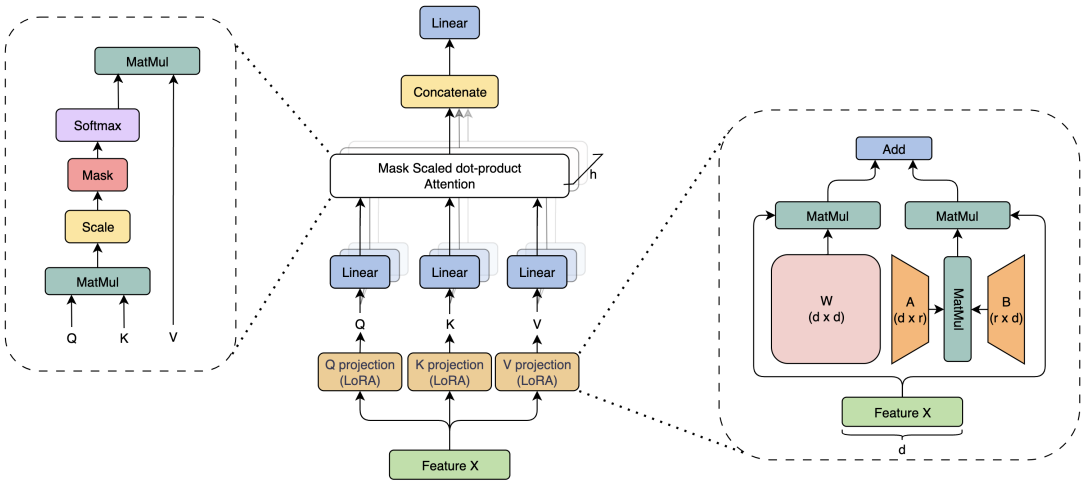


Figure 11. **Mask Multi-Head Attention mechanism is applied LoRA technique.** In Mask Scaled dot-product Attention operation, 'mask' step is added before computing softmax. Q and K projection also be applied the same LoRA technique as shown in V projection.

Although using VinaLLaMA helps us understand and incorporate knowledge and cultural nuances of Vietnamese, training this model can be highly resource-intensive. Specifically, it demands significant memory and computational power. To address these resource constraints, we apply the Low-Rank Adaptation (LoRA) technique [61] to the MMHA layer of the VinaLLaMA model as illustrated in 11. LoRA is a technique designed to reduce the computational and memory requirements associated with fine-tuning LLMs. Instead of updating all the parameters in the model during the fine-tuning process, LoRA focuses on a smaller subset of parameters. To achieve this, LoRA introduces two low-rank matrices, A and B. These matrices have dimensions where A is (d, r) and B is (r, d), with r being significantly smaller than d. This setup is why the method is called "low-rank".

During fine-tuning, instead of updating the entire weight matrix W of the model, LoRA focuses on updating only two low-rank matrices, A and B. This approach reduces the total number of parameters that need to be adjusted to $d * r + r * d$, which is considerably less than updating the full matrix. This reduction in the number of parameters leads to lower computational and memory requirements. The forward computation with LoRA involves calculating the hidden representation h using the formula:

$$h = (W + AB)X \tag{7}$$

In this equation, W represents the original weight matrix, A and B are the low-rank matrices, and X is the input embedding. The product AB forms an adaptation matrix that approximates the changes needed to modify the original weight matrix W to fit new data. Since A and B are low-rank, their product AB is also low-rank, which makes its computation efficient.

Also in Equation 7, the use of addition rather than multiplication or more complex operations in combining W and AB simplifies the process. This addition operation effectively merges the adapted weights with the original weights without requiring additional parameters. The simplicity of addition also aids in the efficiency of back-propagation by focusing on adjusting only the low-rank matrices A and B, thereby simplifying gradient computation. If necessary, the original weights W can still be updated as they would be in traditional fine-tuning.

4.2.4 Loss Function

In this problem, we employ the common categorical loss function known as Cross Entropy. The Cross Entropy loss function is used to measure the discrepancy between the predicted probabilities and the actual ground truth values. It is expressed by the following equation:

$$Loss = -\sum_{l=1}^{L} \log p(y^l | X, I, y^{<l}) \tag{8}$$

where X is the input video, I is the instruction used, y^l is the l-th text token of the ground truth sentence, $y^{<l}$ is the previous predictions, and L is the length of ground truth.

4.3 Metrics

For evaluation, this work adopts two of the most commonly used metrics in speech recognition: Word Error Rate (WER) and Character Error Rate (CER) [62, 63]. Lower values of these metrics indicate better performance of the speech recognition models. In this study, the WER metric is utilized throughout the training, validation, and testing phases, providing a comprehensive measure of overall model performance. Meanwhile, the CER metric is employed exclusively during the testing phase to offer a more granular assessment of the model's accuracy at the character level.

WER is advantageous for providing a quick and comprehensive assessment of the model's overall performance, as it measures the rate of errors at the word level. However, WER has its limitations, as it cannot provide detailed insights into the nature of the errors, such as specific mispronunciations or character-level mistakes.

On the other hand, CER offers a more granular view by evaluating errors at the character level. This can highlight common pronunciation errors and other specific issues that WER might miss. However, calculating CER is computationally intensive because it requires character-by-character comparison, making it less suitable for frequent use during training and validation.

By combining these two metrics, this study can leverage the strengths of each. WER provides a broad overview of model performance during the iterative stages of development, while CER offers deeper insights during the final evaluation phase, ensuring a thorough understanding of the model's accuracy and areas for improvement.

4.4 Experiments

In this section, the details of the experiments conducted during this study will be discussed. All experiments were conducted on an NVIDIA RTX A4000 16GB GPU, and the models were implemented using the PyTorch framework [64]. Moverover, models used in experiments utilizes the same checkpoints for backbones. For the AV-HuBERT backbone, the Large model's checkpoint pre-trained on LRS3 [65] and VoxCeleb2 [66] was utilized as referred in [40]. For the VinaLLaMA-2.7B backbone, a checkpoint fine-tuned on data synthesized by the model's research team [44] was employed.

Hyperparameters	**Base Model**	**Final Model**
Number of clusters in deduplication	200	200
Amount of data in deduplication (hours)	10	20
Number of freezing-parameters steps	13,000	10,000
Amount of training data (hours)	100	1,000
Number of training steps	30,000	300,000
Number of warm-up steps	10,000	100,000
Batch size	1	1
Learning rate	0.005	0.005
Optimizer	Adam	Adam
Betas	(0.9, 0.98)	(0.9, 0.98)
Epsilon	1×10^{-8}	1×10^{-8}
Weight decay	0.0	0.0
Scheduler	Tri-stage	Tri-stage

Table 7. **Configuration of hyperparameters.**

Initially, a base model was trained to serve as a benchmark for evaluating the results of subsequent experiments. The base model was trained using both audio and video data from the 100-hour training subset, as described in Section 4.1.4, over the course of one day. The remaining hyperparameters followed the configurations outlined in the reference paper [40], as detailed in Table 7.

Following this, a series of experiments was conducted to explore the impact of aspects of the model including:

1. **Modality.** To assess the impact of each modality on the model's performance. Results are discussed in Section 5.2.

2. **Parameters in Deduplication** To evaluate the impact of the amount of data and the number of clusters inputted into the K-Means model during the deduplication process on model performance. Results are discussed in Section 5.3.

3. **Number of parameter-freezing steps** To determine the effect of the initial number

of steps in which the encoder parameters are frozen on model performance. Results are discussed in Section 5.4.

4. **Amount of training data** To assess the impact of the number of hours of training data on model performance. Results are discussed in Section 5.5.

In experiments 1 to 3, all hyperparameters remained consistent with those of the base model depicted in Table 7, except for the specific hyperparameter being assessed. In experiment 4, when training the model on the 200-hour training subset, the number of training steps and the number of warm-up steps were adjusted to 60,000 and 20,000, respectively, while all other hyperparameters were kept the same as those of the base model.

The insights gained from these experiments were subsequently applied to train the final model. The final model was trained on both audio and visual data from the full training set (1,000 hours) over a course of 10 days, utilizing the hyperparameters detailed in Table 7. The rationale for selecting these specific hyperparameters is comprehensively discussed in the Results Section (Section 5).

By starting with a solid base model and systematically exploring these critical aspects, the aim was to refine and optimize the final model, ensuring its robustness and efficiency. This thorough experimental approach not only validates the effectiveness of the chosen methodologies but also provides a comprehensive understanding of the factors influencing model performance.

Chapter 5

RESULTS

The Results chapter is pivotal in presenting the tangible outcomes of the project, derived from the real-world training and evaluation processes of the ViAVSP-LLM model described in the preceding chapter (Chapter 4). This chapter will detail the experimental findings, encompassing both quantitative and qualitative results, in a clear and comprehensive manner. These results will provide a thorough overview of the project's discoveries, offering valuable insights and significant implications. These insights will be further explored and contextualized in the subsequent Discussion chapter (Chapter 6).

5.1 Main Results

Model	**Validation WER (%)**	**Test WER (%)**	**Test CER (%)**
ViAVSP-LLM (base)	14.49	17.28	10.56
ViAVSP-LLM (final)	10.04	12.03	7.2

Table 8. **Main Results of ViAVSP-LLM models on VASR dataset.** The red data represents the best result.

Table 8 presents the results from both the base model and the final model, as detailed in Section 4.4, evaluated on the validation and test sets of the VASR dataset. By leveraging insights gained from prior experiments and utilizing a larger training dataset, the final model achieves a Word Error Rate (WER) of 12.03% and a Character Error Rate (CER) of 7.2% on the test set. These results demonstrate a significant improvement over the base model, highlighting the effectiveness of the experimental refinements and additional training data in enhancing the model's performance.

For a comparative analysis, the ViAVSP-LLM base and final models were evaluated against two highly accurate audio-based models for Vietnamese available on HuggingFace, using audio-only data from the test set. The results, depicted in Table 9, highlight that the ViAVSP-LLM base model, with just 100 hours of fine-tuning data, achieved competitive results of 19.66% WER and 12.29% CER, compared to Nguyen's work [68], which reported 20.92% WER and 10.44% CER. Remarkably, the ViAVSP-LLM final model, trained on 1,000 hours of data, surpassed all three methods with a WER of 13.26% and a CER of 8.09%.

These results underscore the effectiveness of incorporating both audio and visual data in the training process, enhancing the model's ability to understand speech and improving its

Modality	Method	Data (hours)	WER (%)	CER (%)
Audio	Nguyen [67]	250	17.35	9.21
	Nguyen [68]	250	20.92	10.44
	ViAVSP-LLM (base)	100	19.66	12.29
	ViAVSP-LLM (final)	1,000	13.26	8.09
Audio + Visual	ViAVSP-LLM (base)	100	18.34	11.32
	ViAVSP-LLM (final)	1,000	12.03	7.2

Table 9. **Comparison with audio-based methods on the VASR test set.** The "Modality" column refers to the modality of data used in the evaluation process. The "Data" column indicates the amount of fine-tuning data. The red data corresponds to the best-performing method in each modality.

performance on unfamiliar data. Table 9 also demonstrates that the integration of visual data, when available, significantly boosts model accuracy compared to relying solely on audio data.

5.2 Impact of Modality

Modality	Validation WER (%)	Test WER (%)	Test CER (%)
Visual	43.58	68.71	51.62
Audio	15.01	18.34	11.32
Audio + Visual	14.49	17.28	10.56

Table 10. **Results of Experiments on Modality.** The red data represents the best result.

This section discusses the experiments on modality outlined in Section 4.4. Table 5.2 illustrates the significant performance improvement achieved by utilizing audio data alone compared to relying solely on visual data or lip reading. This outcome underscores that the majority of speech information is inherently contained within the audio modality, whereas visual data capture only a fraction of this information. Furthermore, Table 5.2 demonstrates that combining speech information from both audio and visual modalities enhances model accuracy more effectively than using a single modality. This synergistic approach leverages the complementary strengths of both modalities, leading to superior performance in speech recognition tasks.

5.3 Impact of Parameters in Deduplication

In this section, the two aspects of the deduplication process examined are the amount of data and the number of clusters inputted to the K-Means model. Detail of the experiments in this section is mentioned in 4.4. As shown in Table 11, increasing either of these parameters does not guarantee improved model performance. Initially, as the parameters increase, the Validation WER decreases up to a certain point, beyond which it starts to rise again, sometimes exceeding the initial results. This behavior is illustrated in Figure 12.

If the number of clusters inputted to the model is fewer than the actual number of clusters in the data, the K-Means algorithm tends to group distinct data points into a single cluster, leading to the scenario depicted in Figure 12(a). In the context of the ViAVSP-LLM model,

Number of hours			Number of clusters			Validation WER (%)
10	20	30	200	1,000	2,000	
x			x			14.49
x				x		14.43
x					x	16.57
	x		x			13.98
		x	x			18.49

Table 11. **Results of Experiments on Parameters in Deduplication.** The red data represents the best result.

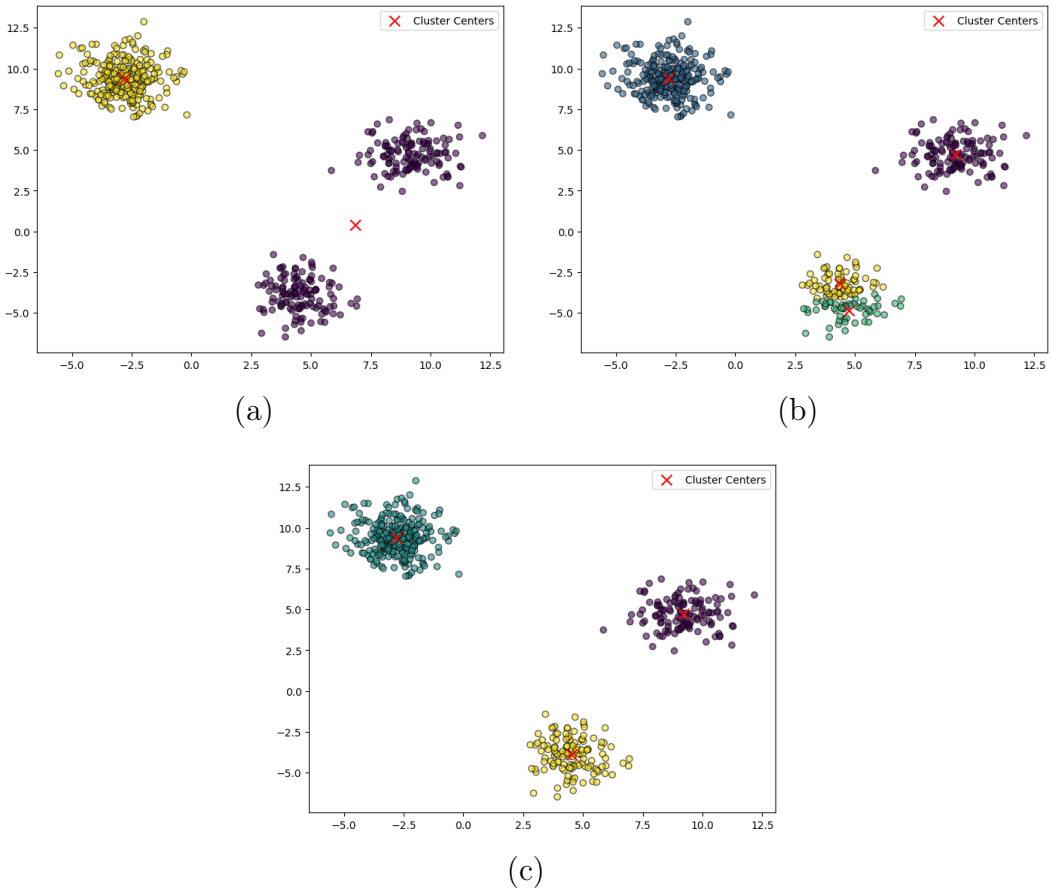


Figure 12. **Example of K-Means.**

this causes the deduplication process to assign the same label to different features, resulting in the removal of useful features. Conversely, if the number of clusters inputted to the model exceeds the actual number, K-Means splits an actual cluster into multiple groups, as shown in Figure 12(b). This leads to the deduplication process splitting similar features into different groups, resulting in replicated features in the model inputs.

To avoid these issues, a balance between the amount of data and the number of clusters is necessary. While achieving an ideal scenario like Figure 12(c) is challenging, experiments detailed in Table 11 suggest that using 2 hours of data and 200 clusters provides an acceptably optimal balance for training the final model. This choice is also supported by [40], which notes that increasing the number of clusters raises the Floating Point Operations (FLOPs), thereby slowing down the training process. Therefore, the number of clusters is kept small to meet the

study's time and resource constraints.

5.4 Impact of Freezing Parameters

Number of parameters-freezing steps	Validation WER (%)
40,000	21.26
13,000	14.49
10,000	13.08

Table 12. **Results of Experiments on Number of Freezing-parameters Steps.** The red data represents the best result.

In this section, the impact of freezing parameters, particularly the number of parameter-freezing steps, is examined. Detail of the experiments in this section is available in Section 4.4. The parameter-freezing steps refer to the initial N steps during which the encoder's parameters remain frozen. Following the methodology outlined in [40], this freezing technique allows the LLM decoder to adapt to the inputs from the encoder. Subsequently, the encoder is unfrozen to learn representations from the training data. As shown in Table 5.4, maintaining an appropriate number of parameter-freezing steps enhances model performance. Specifically, freezing the parameters for 10,000 steps yielded the best results, achieving a WER of 13.08%. Consequently, this configuration was selected for the final model.

5.5 Impact of Amount of Training Data

Amount of training data (hours)	Validation WER (%)
100	14.49
200	11.84

Table 13. **Results of Experiments on Amount of Training Data.** The red data represents the best result.

This section discusses the outputs of experiments on the amount of training data as outlined in Section 4.4. The results presented in Table 13 demonstrate that increasing the amount of training data significantly enhances the model's performance. This trend aligns with recent observations in the literature [40, 38, 35, 9, 69, 70].

5.6 Impact of Rich Context

Integrating the LLM into the ViAVSP-LLM model has significantly improved its ability to understand and generate contextually accurate transcripts. As shown in Table 14, the model's best results on the test set closely match their ground truths. These instances typically involve longer speech segments, providing rich contextual information for accurate inference. In contrast, the worst cases, depicted in Table 15, exhibit high WER and CER. These shorter utterances lack sufficient context, leading the model to produce inaccurate transcriptions. The data suggests a clear correlation: longer speech segments enhance model performance. This is further supported by Figure 13, which illustrates the inverse relationship between text length and error rates (WER and CER). This trend aligns with findings by Yeo et al. [40].

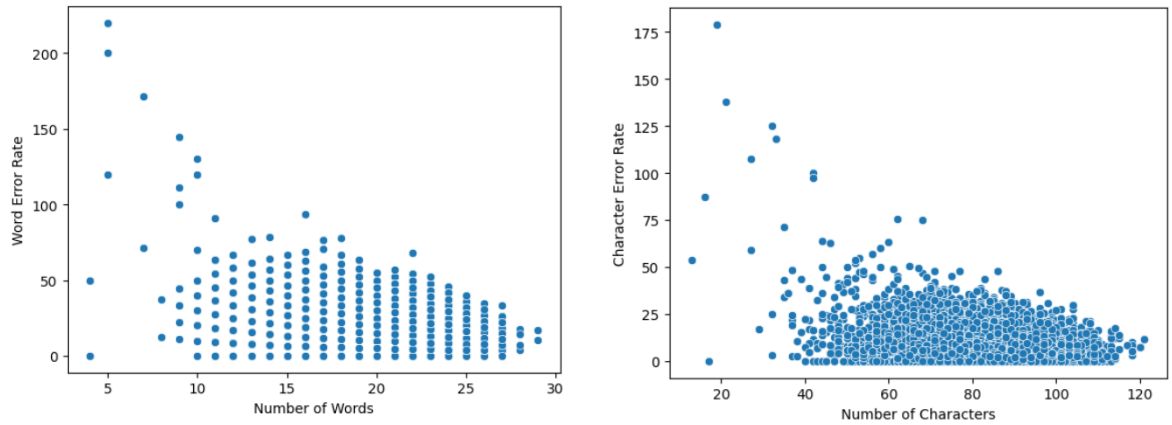
Hypothesis	**Reference**	**WER (%)**	**CER (%)**
tụi mình cứ như thế bên cạnh nhau trải qua một vài tháng hạnh phúc bên cạnh nhau	tụi mình cứ như thế bên cạnh nhau trải qua một vài tháng hạnh phúc bên cạnh nhau	0.0	0.0
khi không một ngày không nhân dịp gì cả đi ăn với nhau cái tự nhiên mình kêu là	khi không một ngày không nhân dịp gì cả đi ăn với nhau cái tự nhiên mình kêu là	0.0	0.0
tiếc lúc đó thì tụi mình chỉ nghĩ lại lấy cái cuộn phim ra không đúng kỹ thuật á	tiếc lúc đó thì tụi mình chỉ nghĩ lại lấy cái cuộn phim ra không đúng kỹ thuật á	0.0	0.0
xảy ra với mình luôn thì câu chuyện này xảy ra vào vài năm trước thì lúc đó mình có nhắn tin trên	xảy ra với mình luôn thì câu chuyện này xảy ra vào vài năm trước thì lúc đó mình có nhắn tin trên	0.0	0.0
chưa bao giờ thấy hai người này thể hiện là một cặp đôi yêu nhau thật sự thậm chí	chưa bao giờ thấy hai người này thể hiện là một cặp đôi yêu nhau thật sự thậm chí	0.0	0.0

Table 14. **Best Inference Cases in Test Set.**

Hypothesis	**Reference**	**WER (%)**	**CER (%)**
đi theo nữa làm sao xuất kim đóng này nó đi nho	lép nằm con nay nho	220.00	178.95
cái đợt ô ét nó ghê lắm luôn kiểm sát tao dự	tôi kiểm soát giao dự	200.00	138.10
đối xử với họ bằng cái mức độ mà họ đối xử với bạn hồng hơn không kém	bằng với bạn không hơn không kém	171.43	125.00
vậy này luôn có ao nét á thì cũng ao nét ba bốn tám là thôi một cụ phi	có nét thì cũng a nét tấm mà thôi	144.44	118.18
i síp cho đàn ông đồng vòng cờ lai trim lên tới một triệu hai trăm mười hai ngàn	và cộng đồng vòng gờ lai rim lên tới triệu	130.00	100.00

Table 15. **Worst Inference Cases in Test Set.**

Additionally, many of the worst cases are nonsensical, highlighting a limitation of the VASR dataset. The presence of noisy or irrelevant data contributes to the model's hallucinations in these instances, underscoring the need for cleaner, more curated datasets to further improve performance.



(a) (b)

Figure 13. **Correlation between Text Length and Metrics.** The figure (a) depicts the correlation between text length and WER, while the figure (b) illustrates the correlation between text length and CER.

Chapter 6

DISCUSSIONS

The Discussion chapter serves as a platform for in-depth analysis, interpretation, and exploration of the implications arising from the results presented in Chapter 5. This chapter delves into the nuances of the findings, uncovering both insights and limitations inherent in the research process. Additionally, it explores potential avenues for future research and practical applications, fostering a comprehensive understanding of the subject matter and enriching the broader discourse within the field. By presenting clear and detailed analyses, this project significantly contributes to the ongoing discussion surrounding audio-visual speech recognition, promoting deeper understanding and providing valuable insights for future endeavors in this domain.

6.1 Interpretation and Implications

The ViAVSP-LLM model and the experimental results presented in this thesis have convincingly demonstrated that integrating auditory and visual data in speech recognition significantly enhances the accuracy of produced transcripts. Furthermore, compared to other audio-based methods, the ViAVSP-LLM models trained on both audio and visual data have exhibited superior performance on audio-only evaluation datasets. Notably, Nguyen's method [68], which uses 250 hours of fine-tuning data, is outperformed by the ViAVSP-LLM base model trained on just 100 hours of data. This finding underscores the substantial impact of incorporating LLM, revealing that integrating LLM into speech recognition models significantly boosts their contextual understanding capabilities, even with limited training data.

6.2 Limitations and Future Works

While the ViAVSP-LLM model has shown promising results, several limitations need to be addressed in future research. Firstly, the produced transcript may not always be entirely accurate due to the noise present in the data, as mentioned in Section 5.6. Improving data quality is crucial for enhancing the model's ability to interpret speeches correctly. Future work should focus on refining data collection and processing techniques to mitigate this issue.

Regarding the dataset, there is a lack of Vietnamese multimodal datasets such as VASR. This shortage is primarily due to the labor-intensive nature of producing accurate annotations. The studies in [71] and [72] have demonstrated promising results using synthetic data to train AVSR models, achieving competitive performance compared to real data. This opens up

opportunities for developing more accurate models with less reliance on labor-intensive data collection processes.

Moreover, although integrating LLM into the model has enhanced its capability to comprehend context, it also increases the model's complexity, leading to heavier constraints on resources such as time and hardware. Future work should focus on optimizing the model to reduce these constraints while maintaining or improving performance.

Addressing these limitations and exploring these future research directions will significantly contribute to the continued advancement of multimodal speech recognition technology.

Chapter 7

CONCLUSIONS

In summary, this thesis has explored and evaluated the approach of combining audio and visual data in automatic speech recognition (ASR) systems for the Vietnamese language. Through experimentation and analysis of this method on the VASR dataset, valuable insights have been gained regarding the strengths and limitations of this approach. The results demonstrate that integrating audio and visual data proves highly effective. Based on these findings, the project has developed a demo with the ViAVSP-LLM model.

The research results highlight the potential of audio-visual models in enhancing the accuracy and performance of speech recognition from multimedia data. Despite the promising performance of these models, some challenges and limitations have been identified, including the complexity of the model, and the need for rich and diverse training data. Future research efforts may focus on addressing these challenges by refining model architectures, expanding training data and enhancing preprocessing methods.

Overall, this thesis contributes to the advancement of automatic speech recognition technology and emphasizes the importance of integrating audio and visual data, marking the first study of this approach for the Vietnamese language. By continuing to innovate and refine these methods, the field can move toward a future where speech recognition technology becomes more accurate and widespread, offering practical benefits for daily life and various other domains.

REFERENCES

[1] J. S. Edu, J. M. Such, and G. Suarez-Tangil, "Smart home personal assistants: A security and privacy review," *ACM Computing Surveys*, vol. 53, p. 1–36, Dec. 2020.

[2] G. Chollet, H. Sansen, Y. Tevissen, J. Boudy, M. Hariz, C. Lohr, and F. Yassa, "Privacy preserving personal assistant with on-device diarization and spoken dialogue system for home and beyond," 2024.

[3] S. Esposito, D. Sgandurra, and G. Bella, "Alexa versus alexa: Controlling smart speakers by self-issuing voice commands," 2022.

[4] H.-Y. Shum, X. He, and D. Li, "From eliza to xiaoice: Challenges and opportunities with social chatbots," 2018.

[5] D. Kong, "Science driven innovations powering mobile product: Cloud ai vs. device ai solutions on smart device," 2017.

[6] J. Li, B. Wang, Y. Zhi, Z. Li, L. Li, Q. Hong, and D. Wang, "Oriental language recognition (olr) 2020: Summary and analysis," 2021.

[7] R. Mitev, A. Pazii, M. Miettinen, W. Enck, and A.-R. Sadeghi, "Leakypick: Iot audio spy detector," in *Annual Computer Security Applications Conference*, ACSAC '20, ACM, Dec. 2020.

[8] F. M. Ramirez, L. Chkhetiani, A. Ehrenberg, R. McHardy, R. Botros, Y. Khare, A. Vanzo, T. Peyash, G. Oexle, M. Liang, I. Sklyar, E. Fakhan, A. Etefy, D. McCrystal, S. Flamini, D. Donato, and T. Yoshioka, "Anatomy of industrial scale multilingual asr," 2024.

[9] P. Ma, S. Petridis, and M. Pantic, "Visual Speech Recognition for Multiple Languages in the Wild," *Nature Machine Intelligence*, vol. 4, pp. 930–939, Oct. 2022. arXiv:2202.13084 [cs, eess].

[10] A. B. A. Hassanat, *Visual Speech Recognition*. InTech, June 2011.

[11] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, "Large Language Models: A Survey," Feb. 2024. arXiv:2402.06196 [cs].

[12] B. Essaid, H. Kheddar, N. Batel, M. E. H. Chowdhury, and A. Lakas, "Artificial intelligence for cochlear implants: Review of strategies, challenges, and perspectives," *IEEE Access*, p. 1–1, 2024.

[13] M. H. M. Noor and A. O. Ige, "A survey on deep learning and state-of-the-art applications," 2024.

[14] L. D. Khai, "Unsupervised pre-training for vietnamese automatic speech recognition in the hykist project," *FH Aachen University of Applied Sciences*, 2023.

[15] A. Härmä, B. den Brinker, U. Grossekathofer, O. Ouweltjes, S. Nallanthighal, S. Abrol, and V. Sharma, "Survey on biomarkers in human vocalizations," 2024.

[16] N. Markl and S. J. McNulty, "Language technology practitioners as language managers: arbitrating data bias and predictive bias in asr," 2022.

[17] R. Aloufi, H. Haddadi, and D. Boyle, "Paralinguistic privacy protection at the edge," 2022.

[18] J. Liao, S. E. Eskimez, L. Lu, Y. Shi, M. Gong, L. Shou, H. Qu, and M. Zeng, "Improving readability for automatic speech recognition transcription," 2020.

[19] S. Zhi, R. P. Levy, and S. C. Meylan, "Multimodal input aids a bayesian model of phonetic learning," 2024.

[20] K. A. Noriy, X. Yang, M. Budka, and J. J. Zhang, "Clara: Multilingual contrastive learning for audio representation acquisition," 2023.

[21] J. Kunze, L. Kirsch, I. Kurenkov, A. Krug, J. Johannsmeier, and S. Stober, "Transfer learning for speech recognition on a budget," 2017.

[22] C.-F. Chen, Q. Fan, N. Mallinar, T. Sercu, and R. Feris, "Big-little net: An efficient multi-scale feature representation for visual and speech recognition," 2019.

[23] K. Hoover, S. Chaudhuri, C. Pantofaru, M. Slaney, and I. Sturdy, "Putting a face to the voice: Fusing audio and visual signals across a video to determine speakers," 2017.

[24] C. Cangea, P. Velickovic, and P. Lio, "Xflow: Cross-modal deep neural networks for audiovisual classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, p. 3711–3720, Sept. 2020.

[25] S. Petridis, Z. Li, and M. Pantic, "End-to-end visual speech recognition with lstms," 2017.

[26] M. Zimmermann, M. M. Ghazi, H. K. Ekenel, and J.-P. Thiran, "Combining multiple views for visual speech recognition," 2018.

[27] E. Egorov, V. Kostyumov, M. Konyk, and S. Kolesnikov, "Lrwr: Large-scale benchmark for lip reading in russian language," 2021.

[28] T. Afouras, J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Deep Audio-Visual Speech Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 8717–8727, Dec. 2022. arXiv:1809.02108 [cs].

[29] T. Afouras, J. S. Chung, and A. Zisserman, "Lrs3-ted: a large-scale dataset for visual speech recognition," 2018.

[30] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Lip Reading Sentences in the Wild," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3444–3453, July 2017. arXiv:1611.05358 [cs].

[31] W. Dai, S. Cahyawijaya, T. Yu, E. J. Barezi, P. Xu, C. T. S. Yiu, R. Frieske, H. Lovenia, G. I. Winata, Q. Chen, X. Ma, B. E. Shi, and P. Fung, "Ci-avsr: A cantonese audio-visual speech dataset for in-car command recognition," 2022.

[32] M. Luo, S. Yang, X. Chen, Z. Liu, and S. Shan, "Synchronous bidirectional learning for multilingual lip reading," 2020.

[33] G. Schwiebert, C. Weber, L. Qu, H. Siqueira, and S. Wermter, "Glips - german lipreading dataset," 2022.

[34] Y. Dai, H. Chen, J. Du, X. Ding, N. Ding, F. Jiang, and C.-H. Lee, "Improving audio-visual speech recognition by lip-subword correlation based visual pre-training and cross-modal fusion encoder," 2024.

[35] P. Ma, A. Haliassos, A. Fernandez-Lopez, H. Chen, S. Petridis, and M. Pantic, "Auto-AVSR: Audio-Visual Speech Recognition with Automatic Labels," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, June 2023. arXiv:2303.14307 [cs, eess].

[36] A. Haliassos, P. Ma, R. Mira, S. Petridis, and M. Pantic, "Jointly learning visual and auditory speech representations from raw data," 2023.

[37] F. Yu, H. Wang, X. Shi, and S. Zhang, "Lcb-net: Long-context biasing for audio-visual speech recognition," 2024.

[38] B. Shi, W.-N. Hsu, K. Lakhotia, and A. Mohamed, "Learning Audio-Visual Speech Representation by Masked Multimodal Cluster Prediction," Mar. 2022. arXiv:2201.02184 [cs, eess].

[39] D. M. Chan, S. Ghosh, D. Chakrabarty, and B. Hoffmeister, "Multi-modal pre-training for automated speech recognition," 2022.

[40] J. H. Yeo, S. Han, M. Kim, and Y. M. Ro, "Where Visual Speech Meets Language: VSP-LLM Framework for Efficient and Context-Aware Visual Speech Processing," May 2024. arXiv:2402.15151 [cs, eess].

[41] P. K. Rubenstein, C. Asawaroengchai, D. D. Nguyen, A. Bapna, Z. Borsos, F. de Chaumont Quitry, P. Chen, D. E. Badawy, W. Han, E. Kharitonov, H. Muckenhirn, D. Padfield, J. Qin, D. Rozenberg, T. Sainath, J. Schalkwyk, M. Sharifi, M. T. Ramanovich, M. Tagliasacchi, A. Tudor, M. Velimirović, D. Vincent, J. Yu, Y. Wang, V. Zayats, N. Zeghidour, Y. Zhang, Z. Zhang, L. Zilka, and C. Frank, "Audiopalm: A large language model that can speak and listen," 2023.

[42] Y. Fathullah, C. Wu, E. Lakomkin, J. Jia, Y. Shangguan, K. Li, J. Guo, W. Xiong, J. Mahadeokar, O. Kalinli, C. Fuegen, and M. Seltzer, "Prompting large language models with speech recognition abilities," 2023.

[43] J. Wu, Y. Gaur, Z. Chen, L. Zhou, Y. Zhu, T. Wang, J. Li, S. Liu, B. Ren, L. Liu, and Y. Wu, "On decoder-only architecture for speech-to-text and large language model integration," 2023.

[44] Q. Nguyen, H. Pham, and D. Dao, "VinaLLaMA: LLaMA-based Vietnamese Foundation Model," Dec. 2023. arXiv:2312.11011 [cs].

[45] A. Ng, "Train / Dev / Test sets - Practical Aspects of Deep Learning." `https://www.coursera.org/lecture/deep-neural-network/train-dev-test-sets-cxG1s`. Accessed: 2 August 2024.

[46] J. Park, J.-W. Hwang, K. Choi, S.-H. Lee, J. H. Ahn, R.-H. Park, and H.-M. Park, "OLKAVS: An Open Large-Scale Korean Audio-Visual Speech Dataset," Jan. 2023. arXiv:2301.06375 [cs].

[47] N. Sethiya and C. K. Maurya, "End-to-end speech-to-text translation: A survey," 2024.

[48] R. Prabhavalkar, T. Hori, T. N. Sainath, R. Schlüter, and S. Watanabe, "End-to-end speech recognition: A survey," 2023.

[49] Y. M. Assael, B. Shillingford, S. Whiteson, and N. de Freitas, "Lipnet: End-to-end sentence-level lipreading," 2016.

[50] A. Pasad, C.-M. Chien, S. Settle, and K. Livescu, "What Do Self-Supervised Speech Models Know About Words?," Jan. 2024. arXiv:2307.00162 [cs, eess].

[51] J. Heo, C.-y. Lim, J.-h. Kim, H.-s. Shin, and H.-J. Yu, "One-Step Knowledge Distillation and Fine-Tuning in Using Large Pre-Trained Self-Supervised Learning Models for Speaker Verification," June 2023. arXiv:2305.17394 [cs, eess].

[52] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks," Oct. 2015. arXiv:1412.0767 [cs].

[53] D. Gramlich, P. Pauli, C. W. Scherer, F. Allgöwer, and C. Ebenbauer, "Convolutional Neural Networks as 2-D systems," Apr. 2023. arXiv:2303.03042 [cs, eess, math].

[54] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," Mar. 2015. arXiv:1502.03167 [cs].

[55] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," Feb. 2015. arXiv:1502.01852 [cs].

[56] H. Gholamalinezhad and H. Khosravi, "Pooling Methods in Deep Neural Networks, a Review," Sept. 2020. arXiv:2009.07485 [cs].

[57] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," Dec. 2015. arXiv:1511.08458 [cs].

[58] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Apr. 2015. arXiv:1409.1556 [cs].

[59] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," Aug. 2023. arXiv:1706.03762 [cs].

[60] A. F. Agarap, "Deep Learning using Rectified Linear Units (ReLU)," Feb. 2019. arXiv:1803.08375 [cs, stat].

[61] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," Oct. 2021. arXiv:2106.09685 [cs].

[62] C. Sheng, G. Kuang, L. Bai, C. Hou, Y. Guo, X. Xu, M. Pietikäinen, and L. Liu, "Deep Learning for Visual Speech Analysis: A Survey," May 2022. arXiv:2205.10839 [cs].

[63] E. Ristad and P. Yianilos, "Learning string-edit distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 522–532, May 1998. Conference Name:

IEEE Transactions on Pattern Analysis and Machine Intelligence.

[64] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," Oct. 2017.

[65] T. Afouras, J. S. Chung, and A. Zisserman, "LRS3-TED: a large-scale dataset for visual speech recognition," Oct. 2018. arXiv:1809.00496 [cs].

[66] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep Speaker Recognition," in *Interspeech 2018*, pp. 1086–1090, ISCA, Sept. 2018.

[67] T. B. Nguyen, "nguyenvulebinh/vietnamese-wav2vec2," June 2024. original-date: 2022-11-04T13:22:34Z.

[68] T. B. Nguyen, "Vietnamese end-to-end speech recognition using wav2vec 2.0," 09 2021.

[69] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units," June 2021. arXiv:2106.07447 [cs, eess].

[70] P. Ma, S. Petridis, and M. Pantic, "End-to-end Audio-visual Speech Recognition with Conformers," Feb. 2021. arXiv:2102.06657 [cs, eess].

[71] R. Liu, J. Wei, F. Liu, C. Si, Y. Zhang, J. Rao, S. Zheng, D. Peng, D. Yang, D. Zhou, and A. M. Dai, "Best Practices and Lessons Learned on Synthetic Data for Language Models," Apr. 2024. arXiv:2404.07503 [cs].

[72] X. Liu, E. Lakomkin, K. Vougioukas, P. Ma, H. Chen, R. Xie, M. Doulaty, N. Moritz, J. Kolar, S. Petridis, M. Pantic, and C. Fuegen, "SynthVSR: Scaling Up Visual Speech Recognition With Synthetic Supervision," pp. 18806–18815, 2023.

Appendix A

DEMO

We have implemented a straightforward demonstration using Gradio, an intuitive web-based interface library, and deployed it on the HuggingFace platform. Gradio provides user-friendly, interactive demos that enable users to interact with our machine learning models directly in their web browsers without needing to install any additional software. By hosting this demo on HuggingFace, a widely recognized platform for sharing and collaborating on machine learning models, we are able to reach a broader audience, providing them with a hands-on experience of our model's capabilities. This setup not only makes it easier for others to understand and engage with our work but also facilitates feedback and collaboration, as users can easily test the model's performance and share their insights. The integration with HuggingFace also ensures that our demo is accessible, scalable, and can be easily updated as our model evolves.

A.1 UI Structure

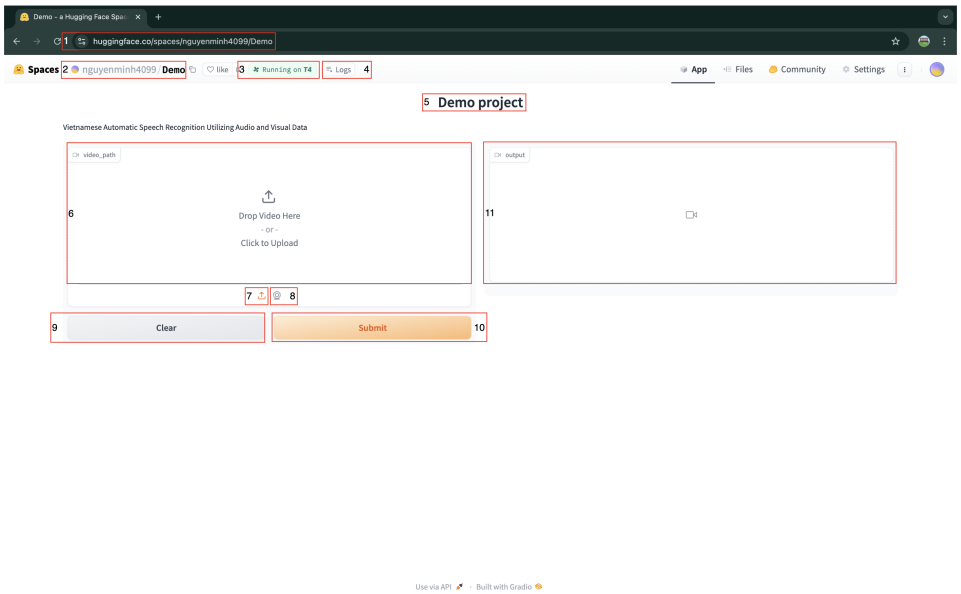


Figure 14. **Overview of Demo Interface.**

Information Available on the Main Site:

1. **Demo Page URL**: The link to the page where the demo can be accessed.

2. **Space ID**: The unique identifier for the demo space.
3. **Space Status and Active Hardware**: Displays the current status of the space and details about the active hardware being used.
4. **Inference Logs**: A log of events and outputs generated during the inference process.
5. **Title of The Application**: The title or name of the application.
6. **Input Video Area**: A designated area where users can input their videos.
7. **Upload Video**: An option to upload a video from a local computer.
8. **Record Video**: An option to record a video directly through the site.
9. **Clear Input and Output**: An option to clear both the input and output videos.
10. **Start Inference**: A button to submit the input video into the pipeline and begin the inference process.
11. **Output Video Area**: The area where the output video is displayed, showing the original video with embedded predicted transcripts.

A.2 Pipeline

Before the transcription prediction process begins, several preprocessing steps are applied to the original video to ensure the model can accurately predict and visualize the output. Below is a detailed explanation of the steps from submitting the input video to the display of the final output (the video with embedded predicted transcripts) in the output area:

1. **Validation of Input Video**: The system first checks if the input video contains both visual (not necessarily including the mouth) and audio components. If the video lacks the required modalities, the pipeline will crash.
2. **Normalization of Video and Audio**: The video and audio components are normalized to improve quality and support the output display. Given the variety of video formats and extensions, this normalization step is crucial. It reformats the video to a unified H.264 codec and avi format, as Gradio only supports displaying video data with the H.264 codec.
3. **Segmentation of the Video**: The video is split into segments, each lasting 3 seconds, to reduce computational intensity. During this process, timestamps (start time, end time) based on the duration of the original video are recorded. These timestamps are essential for embedding the predicted transcripts back into the input video. The splitting algorithm uses a fixed 3-second interval, which was also used during pre-training and fine-tuning. If the input video is shorter than 3 seconds, the algorithm cannot produce any segments, causing the pipeline to crash.
4. **Cropping the Mouth Region**: For each segment, the mouth region of the speaker is cropped. Segments that do not contain a visible mouth image for the full or half duration are ignored. If no segments with the mouth region are found, the pipeline will crash.
5. **Preparation for Model Prediction**: Once the mouth regions are cropped, the dataset

is ready to be loaded into the model for transcript prediction. The model processes each segment individually, predicting the transcript one segment at a time.

6. **Embedding Transcripts and Displaying the Output**: After all the transcripts for each segment have been predicted, they are embedded into the video at the respective timestamps recorded during the segmentation process. The final video, with embedded transcripts, is then displayed in the output area.

Note: When the pipeline is crashed, the input video will be displayed the output area.

The time required for inference depends largely on the duration of the input video and the stability of the internet connection. We welcome any feedback or comments regarding the app's user interface, pipeline, and results. usersr input is invaluable to us, and we are committed to reading and implementing improvements as quickly as possible to enhance performance. Our goal is to provide users with a practical and efficient application that serves both educational and professional needs effectively.

A.3 Use Cases

Our solution addresses the speech-to-text problem by leveraging both visual and audio information from the input video, ensuring a more accurate and robust transcription process. Given this dual-modal approach, the use cases of our application are specifically designed to handle videos that contain both visual and audio components.

A.3.1 Getting a Random Sample from Local Computer

The following are detailed steps to predict a video from a local computer.

1. **Visit the Application Site** To explore our application, please visit our page[1] on HuggingFace, which is displayed similarly to Figure 14. For optimal performance and user experience, we recommend using Google Chrome as usersr browser. Please be aware that Safari on macOS is currently not supported, which may impact functionality.

2. **Upload a Video** Click on the input area, as shown in Figure 15, to select and upload a video file in one of the supported formats: mp4, webm, wav (audio only), avi, etc. After uploading, the video will appear in the input area, as illustrated in Figure 16. Please note that video formats can vary based on the recording and downloading settings of different devices. If the video does not display correctly, it may be due to Gradio's limitation in only supporting video files encoded with the H.264 codec. Videos with a different codec will not be displayed, as shown in Figure 17.

[1] See Section B for URL.

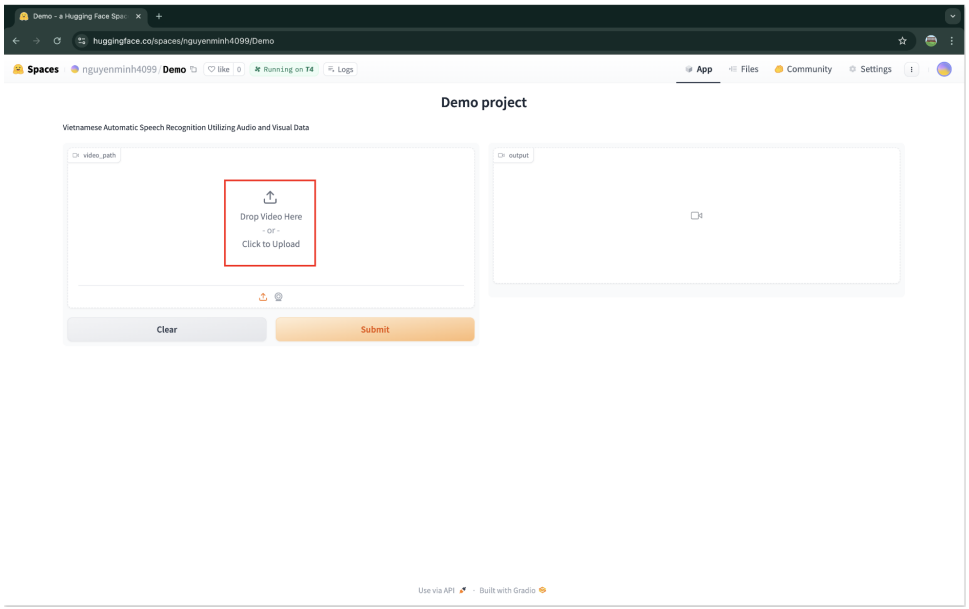


Figure 15. **Uploading a Video.**

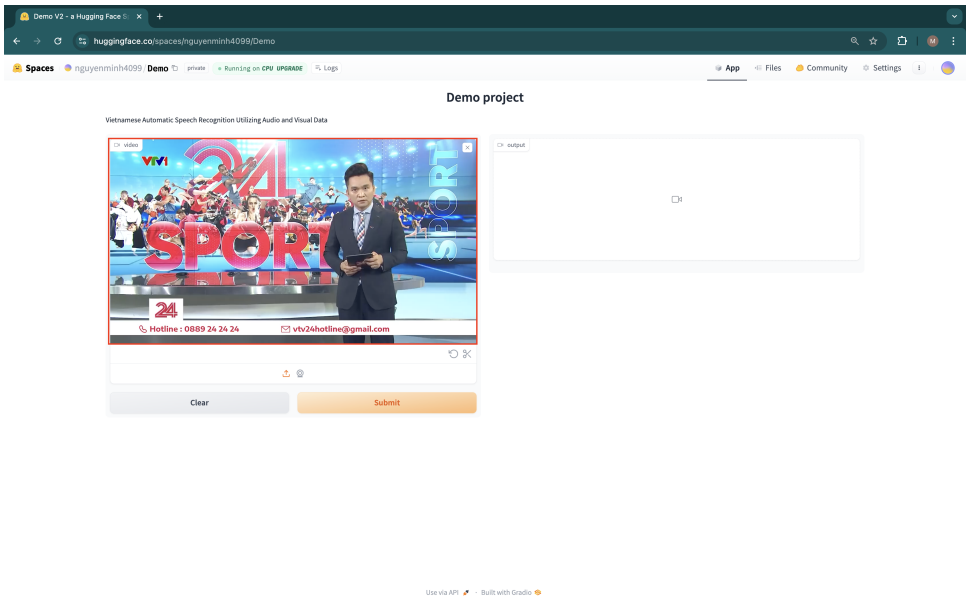


Figure 16. **Displaying the Video.**

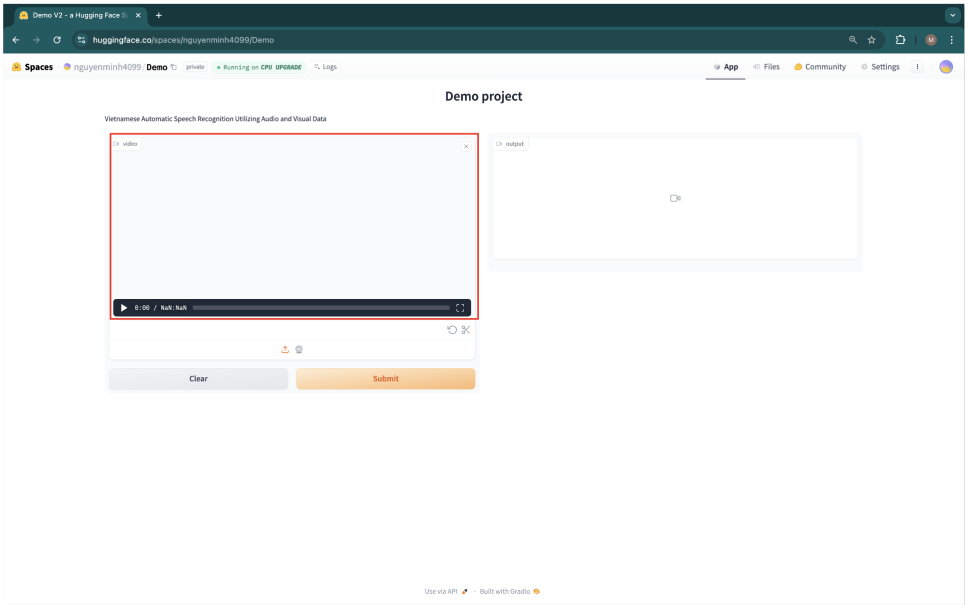


Figure 17. **Error when Displaying the Video.**

3. **Start Inference** To start the inference process, simply click the *Submit* button, as shown in Figure 18. Users can actively monitor the system's memory usage—including disk, RAM, CPU, and GPU—by accessing the Hardware section, depicted in Figure 19. Additionally, the *Logs* section offers real-time updates throughout the process, enabling users to track each step of the inference in detail, as illustrated in Figure 20.

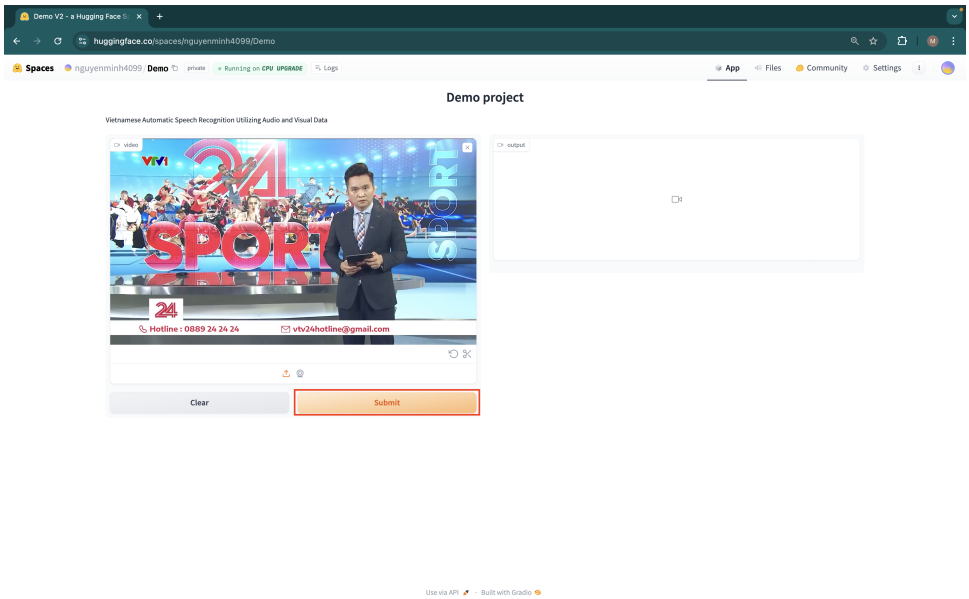


Figure 18. **Starting inference.**

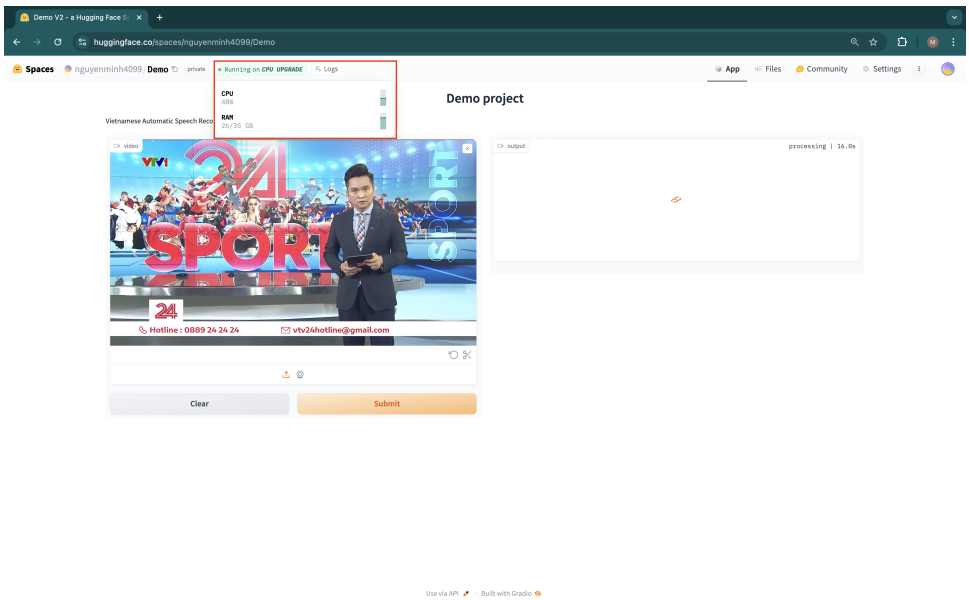


Figure 19. **Checking the Hardware.**

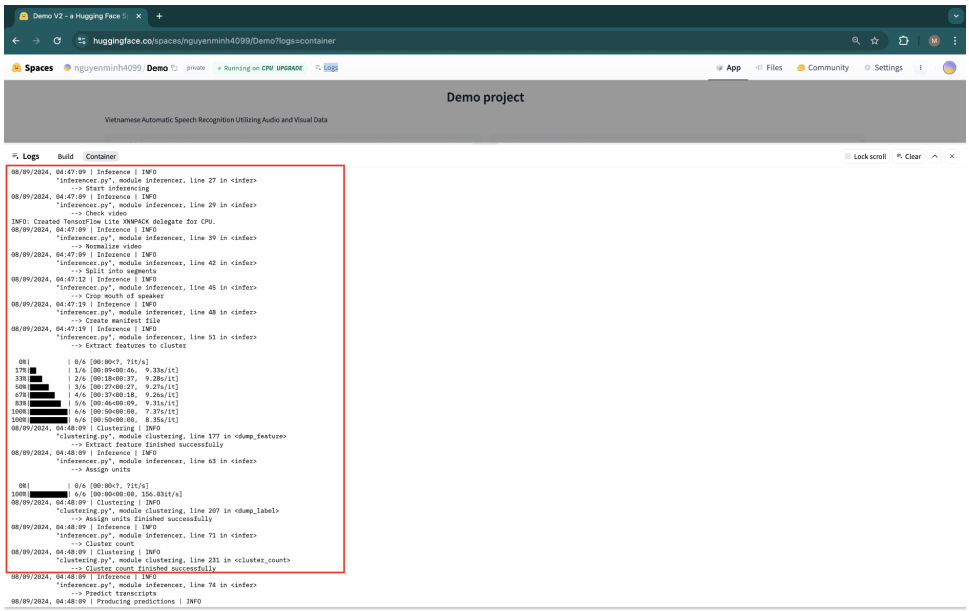


Figure 20. **Viewing the Logs.**

4. **View and Download the Output** Upon completion of the inference process, the processed video will be automatically displayed in the output area, as shown in Figure 21. At this point, the video is readily available for download, ensuring a seamless transition from processing to retrieval.

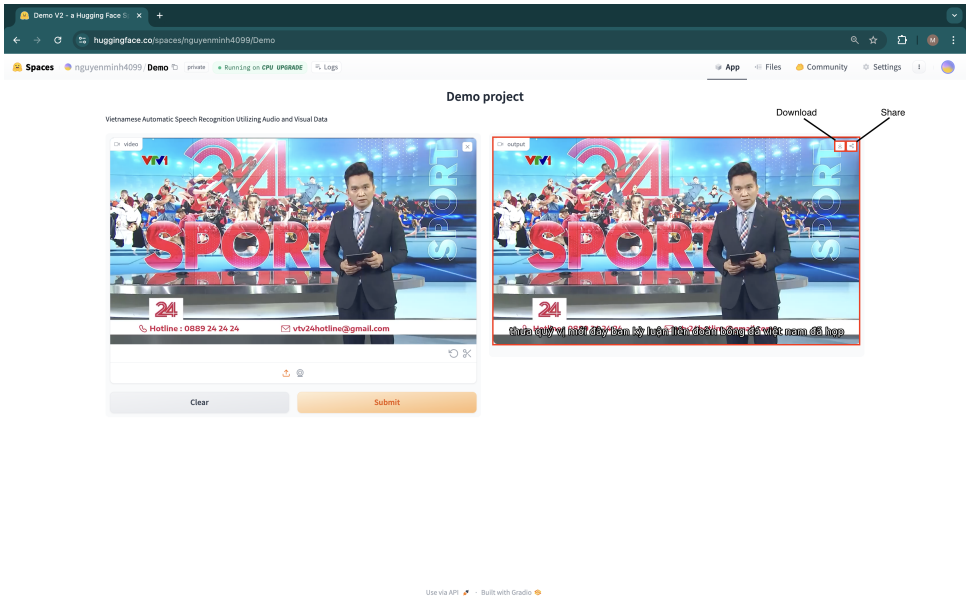


Figure 21. **Receiving the Inference Result.**

5. **Clear the Workspace** Before transcribing another video, click the *Clear* button as in Figure 22 to remove the current results and return the site to the state as in Figure 15.

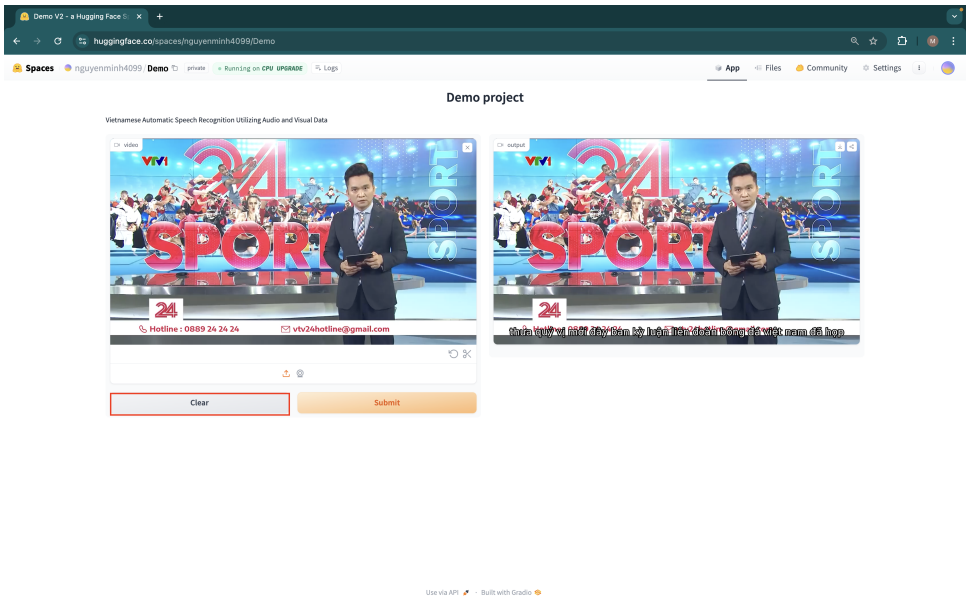


Figure 22. **Clearing the Workspace.**

A.3.2 Recored Directly from Webcam

The following are detailed steps to predict a video recored directly from webcam.

1. **Switch to Webcam Recording** Click on the button with camera symbol (Figure 23) to switch the input method to recording video via the webcam. This will enable the webcam interface for video capture directly within the application.

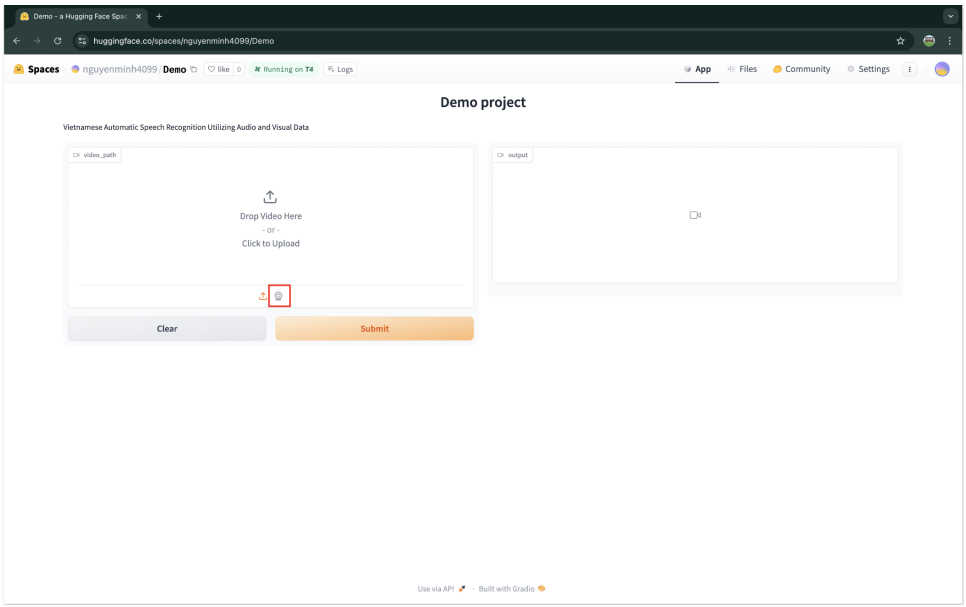


Figure 23. **Switching to Use the Webcam.**

2. **Turn on Webcam** Click center on the input area (red area in Figure 24) to turn on webcam of device.

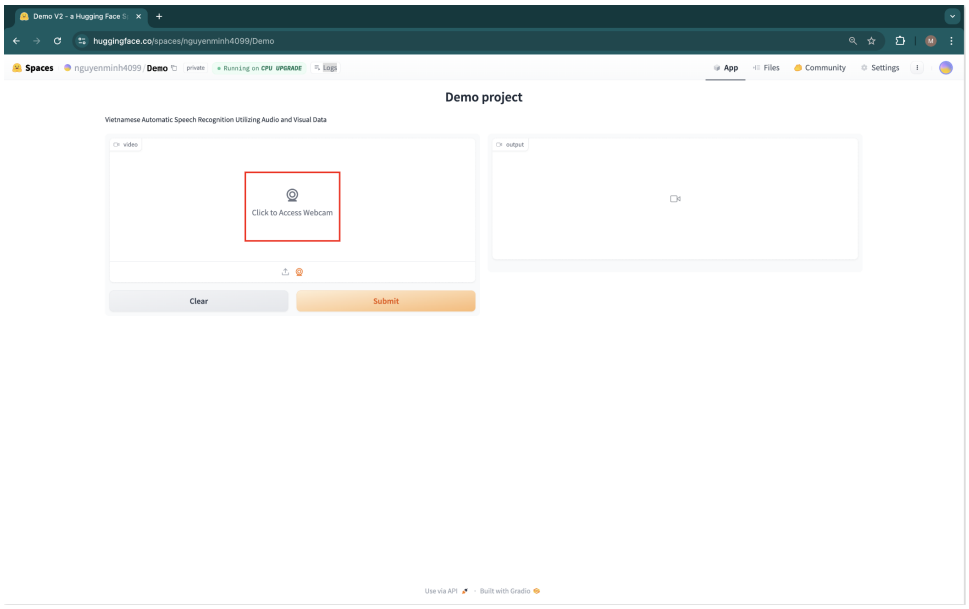


Figure 24. **Turning on the Webcam.**

3. **Start Recording** When starting to see live in input area, which means webcam is turned on, click the red circle button (Figure 25) to start recording the video. The webcam will begin capturing the footage.

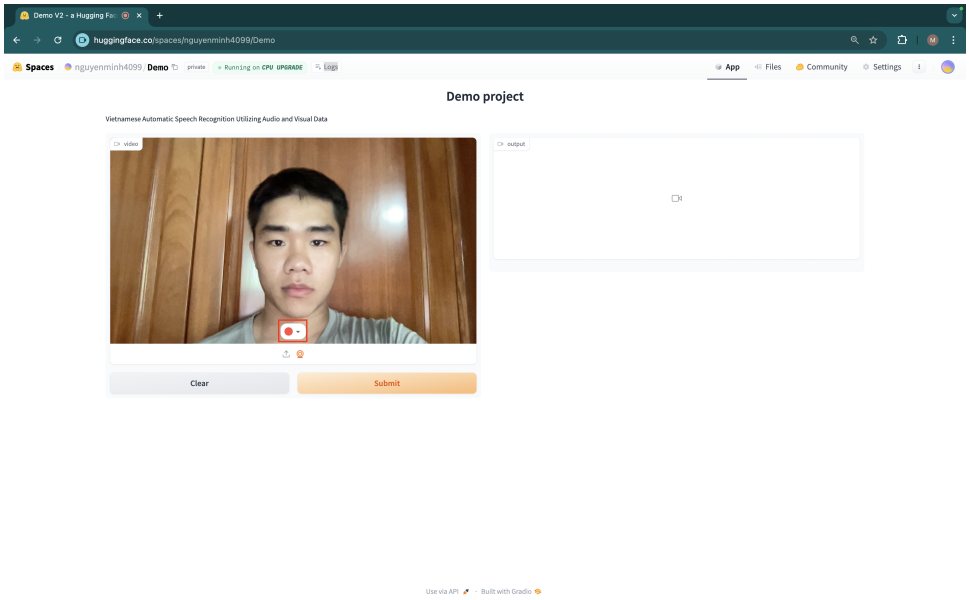


Figure 25. **Starting Recording.**

4. **Stop Recording** Click the red square button as in Figure 26 to stop recording.

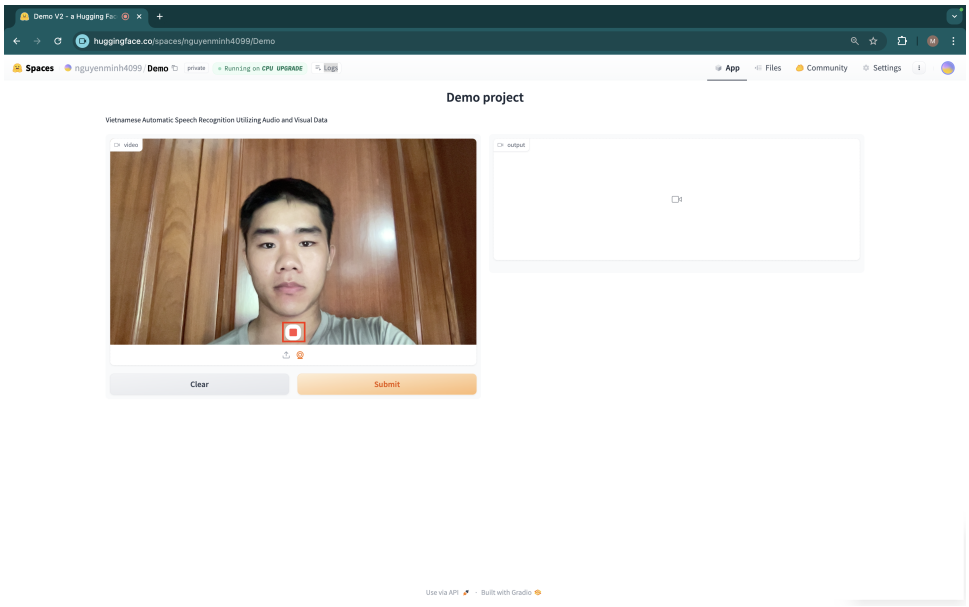


Figure 26. **Stopping Recording.**

5. **Proceed with Inference** Now the recorded video will be displayed in the input area as in Figure 27 and ready for further processing. From this point, follow the same steps starting from Step 3 in Section A.3.1 as when uploading a video from user local device. This includes submitting the video for inference and viewing the results.

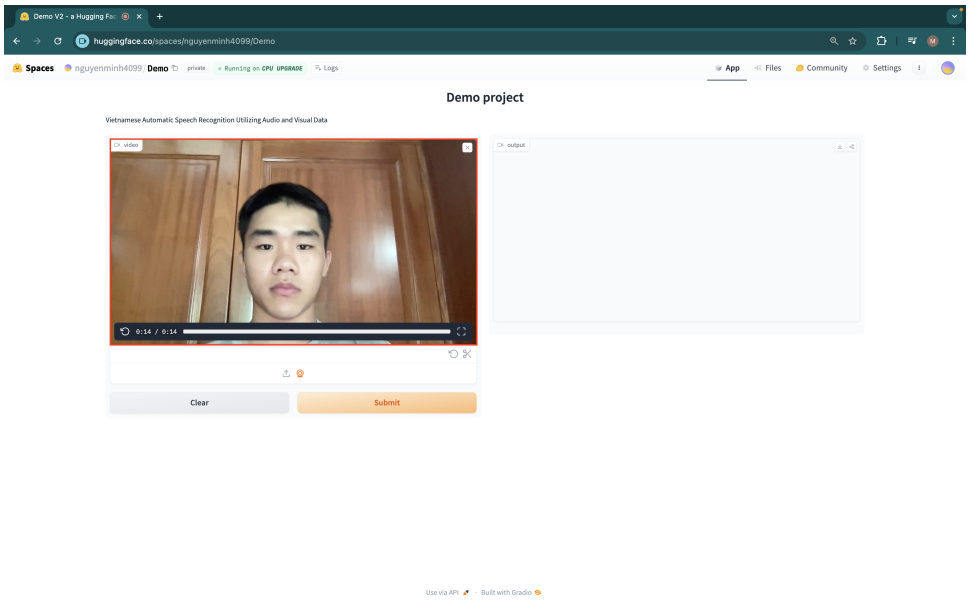


Figure 27. **Proceeding with Inference.**

Appendix B

SUPPLEMENTS

In this chapter, links to the source code, datasets, and model training results are provided in Table 16. To protect speaker anonymity, prospective users must provide their contact information and agree to the stipulated terms and conditions before accessing the dataset. This procedure is crucial for safeguarding individual privacy and ensuring the ethical use of the data.

Item	**Link**	**Description**
VASR dataset	`https://huggingface.co/datasets/tanthinhdt/vasr`	This dataset is the primary resource utilized in this thesis.
Source code	`https://github.com/tanthinhdt/vietnamese-av-asr`	All the code used in this study is stored in main branch in this GitHub repository.
Base model	`https://huggingface.co/GSU24AI03-SU24AI21/ViAVSP_LLM_v1_0`	The checkpoints of the base model.
Final model	`https://huggingface.co/GSU24AI03-SU24AI21/ViAVSP_LLM_v2_0`	The checkpoints of the final model.
Demo	`https://huggingface.co/spaces/nguyenminh4099/Demo`	The demo site on HuggingFace.

Table 16. **Source code and data.**