ក្រសួងអប់រំ យុវជន និងកីឡា

វិទ្យាស្ថានបច្ចេកវិទ្យាកម្ពុជា

ដេប៉ាតឺម៉ង់គណិតអនុវត្ត និងស្ថិតិ

គម្រោងសញ្ញាបត្រវិស្វករ

ប្រធានបទ : ប្រព័ន្ធស្វែងរកអ្នកនិយាយតាមរយៈវីដេអូ

និស្សិត : ហុង គីមម៉េង

ឯកទេស : វិទ្យាសាស្ត្រទិន្នន័យ

សាស្ត្រាចារ្យទទួលបន្ទុក : បណ្ឌិត លិន មង្គលសិរី

ឆ្នាំសិក្សា : ២០២៤ - ២០២៥

MINISTERE DE L’EDUCATION,

DE LA JEUNESSE ET DES SPORTS

INSTITUTE DE TECHNOLOGIE DU CAMBODGE

DEPARTEMENT DE MATHÉMATIQUES APPLIQUÉES

ET STATISTIQUES

MEMOIRE DE FIN D’ETUDES INGENIEUR

Titre : **Détection de Locuteur Actif Basée sur la Vision**

Etudiante : **HONG Kimmeng**

Spécialité : **Science des données**

Tuteur de stage : **Dr. LIN Mongkolsery**

Année scolaire : **2024-2025**

ក្រសួងអប់រំ យុវជន និងកីឡា

វិទ្យាស្ថានបច្ចេកវិទ្យាកម្ពុជា

ដេប៉ាតឺម៉ង់គណិតវិទ្យាអនុវត្ត និងស្ថិតិ

គម្រោងសញ្ញាបត្រវិស្វករ

របស់និស្សិត: ហុង គីមម៉េង

កាលបរិច្ឆេទការពារនិក្ខេបបទ: ថ្ងៃទី០៨ ខែកញ្ញា ឆ្នាំ២០២៥

អនុញ្ញាតឲ្យការពារគម្រោង

នាយកវិទ្យាស្ថាន:________________

ថ្ងៃទី ខែ ឆ្នាំ ២០២៥

ប្រធានបទ: ប្រព័ន្ធស្វែងរកអ្នកនិយាយតាមរយៈវីដេអូ

សហគ្រាស: ក្រុមហ៊ុន អេអាយហ្វាម រូបូតតិក ហ្វេកធ័រី ឯ.ក

ប្រធានដេប៉ាតឺម៉ង់ : បណ្ឌិត លិន មង្គលសិរី ____________

សាស្ត្រាចារ្យដឹកនាំគម្រោង : បណ្ឌិត លិន មង្គលសិរី ____________

អ្នកទទួលខុសត្រូវក្នុងសហគ្រាស : បណ្ឌិត យុន គីមអាង ____________

MINISTERE DE L'EDUCATION,

DE LA JEUNESSE ET DES SPORTS

INSTITUT DE TECHNOLOGIE DU CAMBODGE

DEPARTEMENT DE MATHÉMATIQUES APPLIQUÉES ET STATISTIQUES

MEMOIRE DE FIN D'ETUDES INGENIEUR DE M. HONG Kimmeng

Date de soutenance : le 08 September 2025

« Autorise la soutenance du mémoire »

Directeur de l'Institut : ____________________

Phnom Penh, le 2025

Titre : Détection de Locuteur Actif Basée sur la Vision

Etablissement du stage : AI Farm Robotics Factory Co.Ltd

Doyen de la faculté : Dr. LIN Mongkolsery _________________________

Tuteur de stage : Dr. LIN Mongkolsery _________________________

Responsable de l'établissement : Dr. KHUN Kimang _________________________

INTERNSHIP REPORT ON

“Vision-Based Active Speaker Detection”

at

AI FARM ROBOTICS

AI Farm Robotics Factory Co.Ltd

Submitted By:

HONG KIMMENG

Under the supervision of:

Dr. KHUN KIMANG

Under the advisory of:

Dr. LIN MONGKOLSERY

Date of defense: September 08, 2025

In partial fulfillment of the requirements for the Internship Program

for the award of the degree of

Engineering in Applied Mathematics and Statistics

AMS

Department of Applied Mathematics and Statistics

Majoring in Data Science Department of Applied Mathematics and Statistics

Institute of Technology of Cambodia

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincere gratitude to all those who have supported me throughout the course of my studies and the completion of this thesis. Their guidance, encouragement, and assistance have been invaluable in helping me reach this important milestone.

First and foremost, I would like to express my deepest gratitude to **Dr. LIN Mongkolsery**, Head of the Department and my thesis advisor, for his invaluable guidance, encouragement, and support throughout my academic journey. His insightful advice, constructive feedback, and continuous encouragement have been crucial in shaping this thesis and in helping me overcome challenges along the way.

My heartfelt thanks go to my company supervisor, **Dr. KHUN Kimang**, whose mentorship and practical insights greatly contributed to the success of this work. His professional guidance and patience have been invaluable in bridging the gap between theory and practice.

I am also grateful to my reviewer committee member, **Dr. HAS Sothea**, for his time, effort, and thoughtful suggestions, which have helped me improve the quality of this research.

Finally, I would like to express my deepest appreciation to my family for their unconditional love and support, as well as to my team at the workplace for their encouragement and assistance during this journey. Without their support, this thesis would not have been possible.

ABSTRACT

This internship project aims to create a computer vision system that can find who is speaking in a scene and keep the camera focused on that person. The system works in real time and uses only visual information, without any microphone or audio input. This is useful for robots working in places where sound may not be clear or where audio devices are not available.

The project uses the AVA-ActiveSpeaker dataset to train and test the system. For face detection, YOLOv8 is used because it is fast and accurate. To keep track of each person's face between frames, the SORT tracking algorithm is applied. Speaker detection is done using a deep learning model that combines a Convolutional Neural Network (CNN) and a Long Short-Term Memory (LSTM) network. Pre-trained CNN models such as ResNet50 and EfficientNetB0 are used to get important features from faces, while the LSTM processes 25-frame sequences to understand movements related to speaking.

A camera control module, made with OpenCV, moves the robot's camera so that the detected speaker stays in the center of the frame. The final system was tested and showed good accuracy in finding speakers and keeping them in focus. This project can be useful for human–robot interaction, online meetings, and video recording, providing a vision-only solution that works in changing environments.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

1.1. Presentation of Internship

I completed my internship at **AI Farm Robotics Factory**, a company that works on robotics and AI. The internship took place at st.50m, Anlungkong Village, Dangkor Commune, Preysor District, Phnom Penh, Cambodia. I worked in the **VisionAI** lab as an AI Engineer Intern. My main project was "CenterStage: Vision-Based Speaker Detection for Robots." The goal was to build a system that finds who is speaking in a video and keeps the camera centered on that person in real time. The system uses only visual information. The internship duration is 3 months. I was responsible for designing, implementing, and testing the vision pipeline and preparing a working demo on the robot platform.

1.1.1. Objective of Internship

The objective of this internship was to provide me with practical, hands-on experience in applying computer vision and deep learning techniques to solve a real-world robotics problem. Specifically, my aim was to design and implement a system capable of detecting an active speaker in a scene using only visual information and to integrate this system into a robotic platform for real-time camera control.

From a learning perspective, the internship was intended to strengthen my technical skills in machine learning, computer vision, and deep learning. It also aimed to improve my ability to work with large datasets and to understand the complete process of data preparation, training, and evaluation. In addition, I wanted to gain practical experience with key techniques such as face detection, object tracking, and temporal sequence modeling, which are important in many computer vision applications.

Another important objective was to learn how to integrate an AI model into real-time robotic applications. This included not only developing and testing the model but also optimizing it for speed and stability so it could be deployed on an actual robot. Through this process, I sought to enhance my problem-solving, research, and teamwork skills by working on a project from concept to final deployment.

From the company's perspective, the goal of the internship was to create a vision-only active speaker detection system that could be integrated into its robotic products. Such a system would improve humanrobot interaction without the need for additional audio hardware, aligning with the company's mission to develop autonomous, intelligent, and adaptable robotic systems.

1.1.2. Duration of Internship

The internship lasted for a period of 3 months, starting on 19th May and ending on 19th August. It was conducted at AI Farm Robotics Factory. During this period, I worked under the guidance of my company supervisor, **Dr. KHUN Kimang**, who provided technical and practical direction for the project.

In addition, I was supported by my school advisor, **Dr. LIN Mongkolsery**, who offered academic guidance and ensured that the project met the requirements for my internship defense. The collaboration between my company supervisor and school advisor helped maintain a balance between industrial application and academic standards.

My working schedule followed the company's regular hours, typically from Monday to Friday, 8:00 AM to 5:00 PM, allowing me to be fully engaged in the project while also attending any required academic consultations.

1.2. Presentation of Organization

1.2.1. General Information of Company

AI Farm Robotics Factory, a subsidiary of **Khamsa Group of Businesses (KGB)**, is a technology-driven company based in Phnom Penh, the capital of the Kingdom of Cambodia. The company focuses on the design, development, and manufacturing of robotic systems and automation solutions. Its activities span across hardware engineering, software development, and artificial intelligence, particularly in computer vision, machine learning, and automation control.

I completed my internship in the **VisionAI** lab, which is part of **AI Farm**'s research and development division. The lab works on integrating advanced computer vision technologies into robotic systems, including object detection, tracking, facial recognition, and LLM. This environment provided me with access to state-of-the-art tools and the opportunity to work alongside skilled engineers and researchers in the robotics field.

(a) (b)

Figure 1.1. AI Farm and VisionAI Lab Logo

1.2.2. Vision & Mission

- **Vision**: To create a living code – Robot Makes Robot.
- **Mission**: To transform Cambodia into a robotic nation by fostering innovation, research, and manufacturing capabilities in robotics and AI.
- **Objective**: To become a leading robotics factory in Asia by 2050. Bring Cambodia into the global network of robotics nations, and make the country technologically independent and sovereign.

1.2.3. Address & Contact

- **Address**: THE FARM, FACTORY #3, RD. 39D, RING ROAD 2, DANGKOR, PHNOM PENH, KINGDOM OF CAMBODIA.
- **Website**: https://aifarm.dev
- **Contact**: info@aifarm.dev

II. PRESENTATION OF THE PROJECT

2.1. General Presentation of Project

The project focuses on developing a computer vision system that allows a robot to automatically detect the person who is speaking and keep them centered in the camera frame. The system works entirely from visual input, without the use of microphones or audio analysis.

The core idea is to capture video from the robot's camera, detect faces in real time, track each face across frames, and determine which person is speaking based on visual cues such as lip movements, jaw motion, cheek deformation, and subtle head movements that are typical during speech. Once the active speaker is identified, the robot's camera or head is adjusted to follow that person, ensuring they remain in the center of view.

This system is designed using several AI components. YOLOv8 is used for fast and accurate face detection, while SORT (Simple Online and Realtime Tracking) is used to assign a unique ID to each detected face and track it over time. Speaker detection is achieved using a deep learning model that combines a Convolutional Neural Network (CNN) with a Long Short-Term Memory (LSTM) network, enabling both spatial and temporal analysis of lip movements over sequences of 25 frames. The CNN backbones used include pre-trained models such as ResNet50 and EfficientNetB0 with early layers frozen to retain general visual features.

The project was developed and tested using the AVA-ActiveSpeaker dataset for training and evaluation, and OpenCV for real-time video processing and camera control. The final implementation is intended for integration into AI Farm's robotic systems to improve human–robot interaction in applications such as meetings, telepresence, and customer service.

2.2. Problematic

In human–robot interaction, keeping the camera focused on the person who is speaking greatly improves communication, engagement, and user experience. However, many existing solutions rely heavily on audio-based detection using microphone arrays. While effective in some environments, these solutions have several limitations:

- They require specialized audio hardware, which may not be available on all robots.
- They perform poorly in noisy environments or when multiple background sounds interfere.
- They are less effective in large or open spaces where sound can be distorted or delayed.

Some vision-based solutions exist, such as Apple's Center Stage feature, but they are typically designed for static or semi-static devices like tablets and webcams. They are not optimized for mobile robotic platforms that operate in dynamic environments with multiple moving subjects.

This creates a need for a vision-only, real-time speaker detection system that is reliable, adaptable to different environments, and suitable for integration into robots. By removing the dependency on audio input, such a system can operate effectively in noisy conditions, avoid hardware constraints, and open new possibilities for robotics in public spaces, events, and industrial settings.

2.3. Objective

The main objective of this project is to develop a real-time, vision-based active speaker detection system that can be integrated into a robotic platform. The system should accurately identify the person who is speaking using only video input and adjust the robot's camera to keep the speaker centered in the frame. In order to achieve this, the project has the following specific objectives:

- To prepare a labeled dataset of sequences of cropped faces using the AVA-ActiveSpeaker dataset.
- To implement a face detection module using YOLOv8 and a face tracking module using SORT to maintain consistent IDs for each face across frames.
- To design and train a speaker detection model using CNN and LSTM architectures, capable of recognizing speaking activity from 25-frame face sequences by analyzing visual cues such as lip motion, jaw movement, cheek deformation, and head movements.
- To integrate the detection and tracking system with a real-time camera control module using OpenCV.
- To evaluate the performance of the system in terms of detection accuracy, tracking stability, and real-time responsiveness on a robotic platform.

2.4. Scope of Project

This project focuses on the development of a vision-only active speaker detection system for robotic applications. The system uses only video input without relying on any audio signals or microphones. The scope of the project includes:

- Using the AVA-ActiveSpeaker dataset for training and evaluation.
- Implementing face detection and tracking to handle multiple people in the camera's field of view.
- Designing a deep learning model that combines CNN for spatial feature extraction and LSTM for temporal sequence analysis.
- Real-time deployment of the system on a robotic platform with camera control functionality.

However, the project does not cover:

- Audio-based speaker detection or audio-visual fusion approaches.
- Large-scale field testing in varied outdoor environments beyond the available testing setup.
- Hardware design or modification of the robotic platform.

The final outcome is a functional prototype capable of detecting an active speaker from video and controlling the robot's camera to maintain focus on the speaker in real time.

2.5. Planning of Project

The project was divided into five main phases, each with a specific focus and timeline, as illustrated in Figure 2.2..

The Research phase (Week 1–2) involved understanding the project requirements, reviewing related work, and preparing the development environment. This was followed by the Dataset Preparation phase (Week 3–4), where the AVA-ActiveSpeaker dataset was processed, face sequences were extracted, and tracking IDs were assigned using YOLOv8 and SORT.

The Model Development and Training phase (Week 5–8) focused on building and training the CNN + LSTM model, tuning hyperparameters, and evaluating its performance. Next, the Real-Time Implementation phase (Week 9–10) integrated the detection, tracking, and speaker recognition modules into a unified pipeline and developed the camera control system using OpenCV.

Finally, the Testing and Documentation phase (Week 11–13) involved performance evaluation in real scenarios, debugging, and fine-tuning. The last stage also included preparing the demonstration, finalizing documentation, and getting ready for the internship defense.

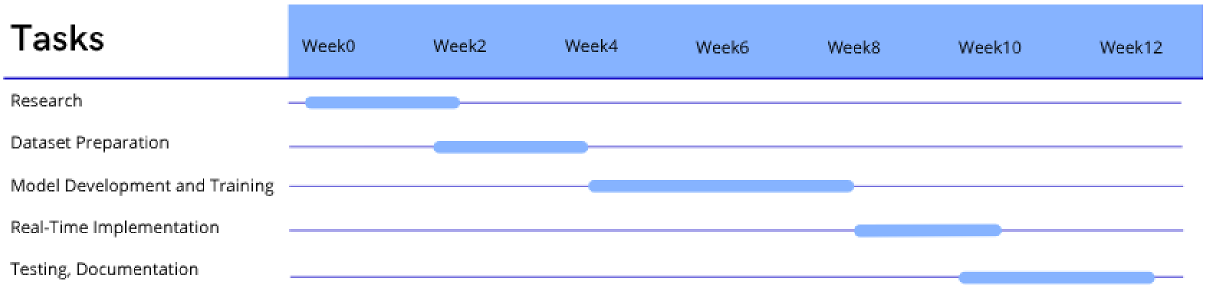


Figure 2.2. Project Planning Timeline

III.. LITERATURE REVIEW

3.1. Speaker Detection Approaches

3.1.1. Multimodal Audio–Visual Methods

Recent research demonstrates the strength of combining audio and visual modalities for active speaker detection. The Multi-modal Speaker Extraction-to-Detection framework (MuSED) leverages pre-training on audio–visual target speaker extraction, improving noise resilience and yielding 95.6% mAP on the AVA-ActiveSpeaker dataset (Tao et al. [15]).

3.1.2. Visual-Only and Landmark-Guided Approaches

In scenarios where audio may be unreliable or unavailable, visual-only methods become critical. The LASER model (Lip Landmark Assisted Speaker Detection for Robustness) incorporates lip landmark coordinates into the visual feature representation and introduces an auxiliary consistency loss to handle missing landmarks. It outperforms baseline models especially in cases where audio and visual streams are desynchronized (Nguyen, Yu, and Lee [11]).

3.1.3. Graph-Based and Assignation Models

Graph-based approaches offer another powerful strategy. The Multi-modal Assignation for Active Speaker Detection (MAAS) model builds a small graph structure to assign visual face representations to speech embeddings. On the AVA-ActiveSpeaker dataset, MAAS achieves a state-of-the-art mAP of 88.8% (Alcázar et al. [2]).

3.2. Face Detection and Tracking

3.2.1. Face Detection

Face detection is a critical component of vision-based speaker detection, as it defines the regions of interest where lip and facial motion can be analyzed. Early approaches relied on Haar cascades and Histogram of Oriented Gradients (HOG) features, but these methods struggled with robustness in real-world conditions. With the rise of deep learning, CNN-based detectors such as Faster R-CNN, SSD, and the YOLO family have become the state of the art.

YOLO (You Only Look Once) models, in particular, are widely adopted for their real-time speed and accuracy. The recent YOLOv8 framework introduces anchor-free detection and decoupled head architectures, making it both lightweight and robust for face detection tasks in unconstrained environments (Jocher et al. [9]). This makes YOLOv8 a suitable choice for robotics and video-based systems, where low-latency performance is essential.

3.2.2. Face Tracking

Once faces are detected, consistent identity assignment across frames is necessary to build temporal sequences. Traditional tracking-by-detection approaches often rely on algorithms like Kalman filtering and Hungarian matching for object association.

SORT (Simple Online and Real-time Tracking) is one of the most popular approaches in this category. It uses bounding box detections as input and applies a Kalman filter for state estimation and the Hungarian algorithm for data association (Bewley et al. [4]). SORT is lightweight and efficient, achieving real-time tracking speeds, though it does not incorporate appearance features. Extensions such as DeepSORT integrate deep feature embeddings for more reliable long-term identity tracking in crowded scenes.

For the purposes of active speaker detection, SORT provides an ideal balance between computational efficiency and tracking stability, making it a strong choice for maintaining consistent speaker identity in multi-face videos.

3.3. Deep Learning for Visual Speaker Detection

3.3.1. CNNs for Spatial Feature Extraction

Convolutional Neural Networks (CNNs) have been the backbone of modern computer vision tasks, including visual speech recognition and active speaker detection. Architectures such as ResNet (He et al. [7]), VGG16 (Simonyan and Zisserman [13]), and EfficientNet (Tan and Le [14]) are widely used for extracting spatial features from face images.

- ResNet introduced residual connections, enabling very deep models without vanishing gradients, making it suitable for capturing fine-grained facial movements.
- VGG16, while older, provides a strong baseline for image classification tasks with its uniform 3×3 convolution design.
- EfficientNet uses compound scaling to balance depth, width, and resolution, achieving state-of-the-art accuracy with fewer parameters, making it attractive for real-time applications.

These architectures form the CNN component of hybrid CNN–RNN pipelines used in lip reading and visual speaker detection tasks.

3.3.2. Temporal Modeling with RNNs and LSTMs

Speaker detection from video requires modeling temporal dynamics of facial motion across frames. While CNNs capture spatial features per frame, Recurrent Neural Networks (RNNs) and particularly Long Short-Term Memory (LSTM) units (Hochreiter and Schmidhuber [8]) are used to model sequential dependencies such as lip movement patterns. LSTMs mitigate vanishing gradient issues in standard RNNs, making them effective for sequence learning in visual tasks.

Recent works have combined CNNs with LSTMs for lip reading and active speaker detection. For example, Chung and Zisserman [5] demonstrated the effectiveness of CNN–LSTM models for lip reading on the LRW dataset, where sequences of mouth crops were classified into spoken words. Such models serve as a direct foundation for vision-only active speaker detection, since the classification relies purely on temporal facial cues.

3.3.3. Datasets for Visual Speech and Speaker Detection

Several datasets have been curated for training and evaluating visual-only models:

- LRW (Lip Reading in the Wild) provides large-scale word-level lip reading data (Chung and Zisserman [5]).
- LRS2/LRS3 (Lip Reading Sentences) extend to sentence-level lip reading (Afouras et al. [1]).
- AVA-ActiveSpeaker (Roth et al. [12]) is one of the most widely used benchmarks for ASD, with dense annotations of speaking activity aligned with video frames.

The AVA-ActiveSpeaker dataset is particularly relevant to this thesis, as it allows supervised training of models on cropped face sequences labeled as speaking or not speaking.

3.3.4. Summary

Deep learning models combining CNNs for spatial feature extraction with LSTMs for temporal modeling have become the dominant paradigm for visual active speaker detection. While earlier work focused on audio-visual fusion, the growing interest in visual-only methods and the availability of large-scale datasets such as AVA-ActiveSpeaker highlight the feasibility of purely vision-based speaker detection in real-world applications.

3.4. Existing Applications

Active speaker detection has already been applied in both commercial and research domains, particularly for video conferencing, human–robot interaction, and assistive technologies.

3.4.1. Commercial Applications

One of the most widely known commercial systems is Apple's Center Stage, introduced in the iPad Pro. It employs computer vision to automatically pan and zoom the camera, keeping the active speaker framed in the view even if they move around. While this works well for video conferencing, it is mainly designed for static or semi-static environments and does not address the dynamic requirements of robotic applications (Apple Inc. [3]). Similarly, platforms such as Zoom and Microsoft Teams use audio-based speaker detection to highlight the active speaker, but these approaches are highly sensitive to background noise and fail when microphones are unavailable.

3.4.2. Research Prototypes in Robotics

In robotics, active speaker detection plays an important role in improving human–robot communication. For instance, telepresence robots integrate face tracking and camera steering to enhance remote presence during conversations (L. Chen and Zhang [10]). Other research has explored multimodal systems that combine microphone arrays with visual analysis of lip movement, yielding more robust speaker localization in noisy environments (Z. Zhang and Lincoln [16]). However, these methods are often heavily dependent on audio signals, making them less effective for vision-only contexts where microphones are absent or unreliable.

3.4.3. Research Gaps

Despite these advances, most commercial systems such as Center Stage (Apple Inc. [3]) and conferencing platforms rely on either vision-only framing without explicit active speaker classification, or audio-only detection. Research systems in robotics often assume both audio and visual inputs (Z. Zhang and Lincoln [16]), which limits deployment in environments where audio cannot be captured. Therefore, there is a clear gap for vision-only, real-time speaker detection frameworks tailored to robotic applications.

3.5. Summary

The literature on active speaker detection shows a clear evolution from traditional audio-only methods to multimodal audio–visual systems and, more recently, to vision-only approaches. Audio-based techniques, while effective in controlled environments, are often unreliable in noisy conditions or when microphone arrays are unavailable. Multimodal systems that combine audio and visual features, such as MuSED, achieve state-of-the-art performance on benchmarks like AVA-ActiveSpeaker but remain dependent on audio streams, limiting their applicability in settings where sound cannot be captured reliably.

Vision-only approaches, including landmark-guided methods like LASER and motion-based CNN models, demonstrate that visual cues such as lip movements, jaw motion, and facial dynamics can provide robust indicators of speech activity. These methods are particularly relevant to robotic systems, where hardware or environmental constraints may preclude the use of microphones.

Complementary techniques in face detection and tracking, such as YOLOv8 and SORT, provide the necessary tools to identify and maintain consistent identities across frames, enabling temporal modeling of speaker behavior. Moreover, CNN–LSTM architectures have been widely applied for tasks like lip reading and visual speech recognition, showing strong potential for vision-only active speaker detection.

Existing applications, such as Apple's Center Stage and video conferencing platforms, focus mainly on audio-driven or vision-only framing without explicit classification of the active speaker. Research prototypes in robotics demonstrate the importance of active speaker detection for human–robot interaction but often rely on multimodal inputs that are not always practical.

In summary, while significant progress has been made, there is still a research gap in developing a vision-only, real-time active speaker detection system tailored to robotic platforms. This thesis aims to address that gap by proposing a CNN–LSTM-based framework that detects the active speaker from cropped face sequences and integrates seamlessly with a robotic camera control module.

IV. METHODOLOGY

4.1. Tools and Frameworks

The implementation of this project required a combination of deep learning frameworks, computer vision libraries, and development tools. The main tools and frameworks used are described below:

Python was the primary programming language for this project. It was chosen due to its strong ecosystem of machine learning and computer vision libraries, ease of prototyping, and wide adoption in the research community.

Figure 4.3. Python Logo

PyTorch was used as the main deep learning framework. It provides flexibility, GPU acceleration, and a dynamic computation graph, which makes it well suited for building and training deep learning models such as CNNs and LSTMs. Pre-trained models such as ResNet50 and EfficientNetB0 were accessed through PyTorch's model zoo and fine-tuned for the speaker detection task.

Figure 4.4. Pytorch Logo

YOLOv8 (Ultralytics) was employed for face detection. It is the latest generation of the YOLO family of object detectors and is known for its high speed and accuracy. Its anchor-free design and strong generalization ability make it particularly effective for detecting faces in real-time video streams.

SORT (Simple Online and Realtime Tracking) was used for tracking multiple faces across video frames. It assigns unique IDs to detected faces and maintains their trajectories over time using a Kalman filter and Hungarian algorithm. This ensured consistency in speaker detection when multiple people were present in the scene.

OpenCV was used for real-time video processing and camera control. It handled tasks such as frame extraction, image preprocessing, drawing bounding boxes, and controlling the robot's camera orientation to keep the detected speaker centered.

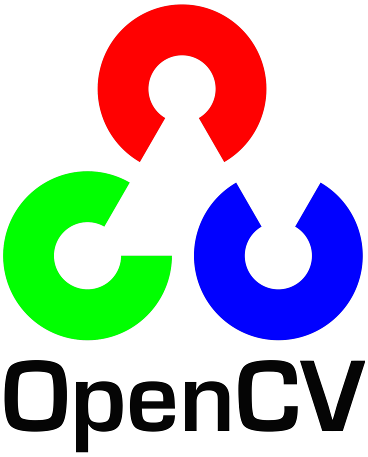


Figure 4.5. OpenCV Logo

Weights & Biases (W&B) was used for experiment tracking, visualization, and model management. It allowed me to log training metrics such as accuracy, loss, precision, and recall in real time, making it easier to monitor model performance during experiments.

Figure 4.6. Weight & Bias Logo

Google Colab with NVIDIA GPU: Model training and experiments were conducted on Google Colab using NVIDIA A100 GPUs with 40GB VRAM. This provided sufficient computational resources for training deep CNN–LSTM models on large-scale video data.

Figure 4.7. Google Colab Logo

Supporting Tools: Other supporting tools included NumPy and Pandas for data handling, Matplotlib for visualization, and Jupyter Notebook for prototyping and interactive experiments. GitHub was used for version control and collaboration.

Figure 4.8. NumPy Logo

Figure 4.9. Pandas Logo

Figure 4.10. Matplotlib Logo

Figure 4.11. GitHub Logo

4.2. System Overview and Architecture

The proposed vision-based active speaker detection system is designed as a modular pipeline that processes video input, detects faces, tracks individuals, identifies the active speaker, and finally controls the camera to focus on the speaker. The architecture can be divided into two workflows: training and inference, as illustrated in Figures 4.12. and 4.13..

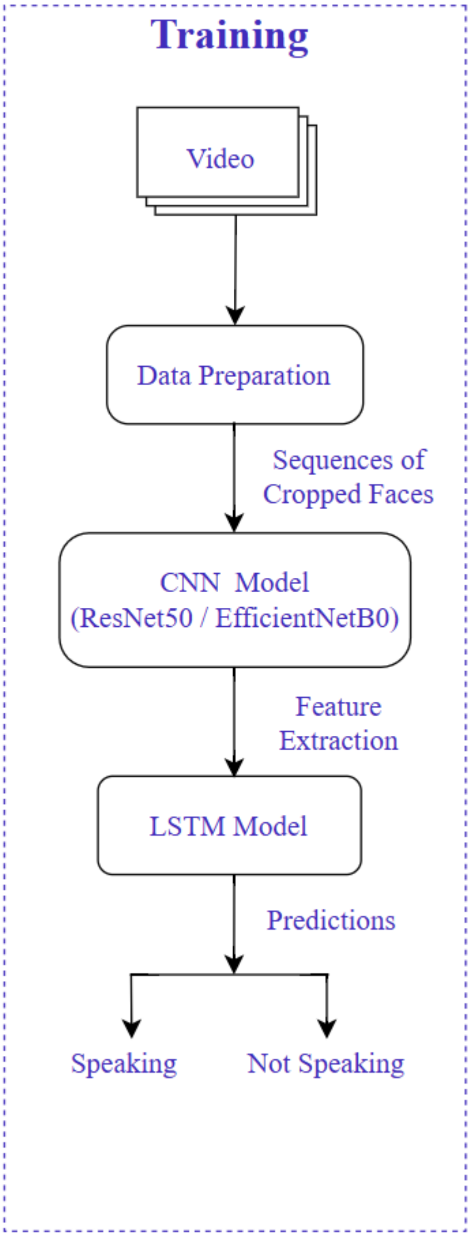


Figure 4.12. Training Workflow of the Proposed System

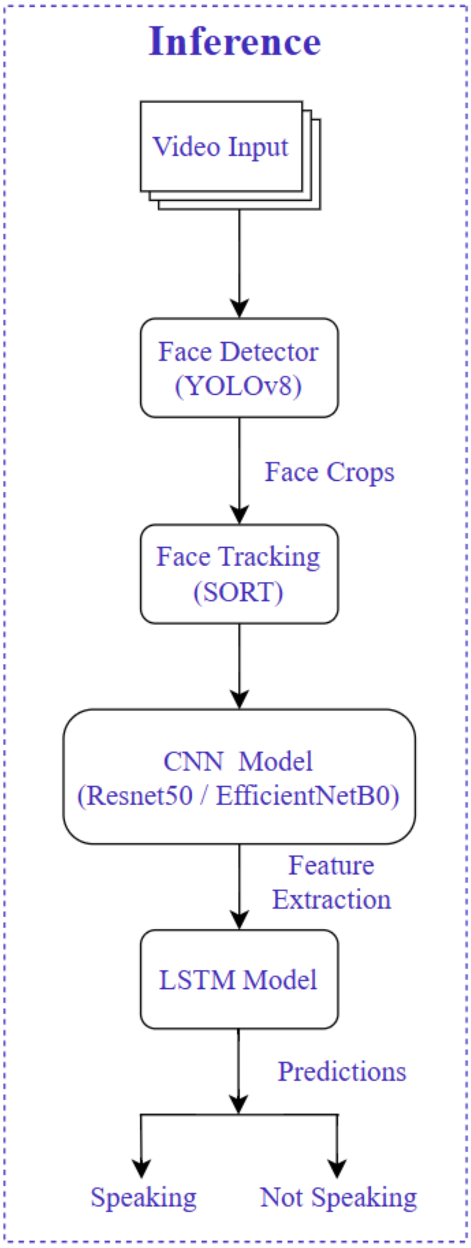


Figure 4.13. Inference Workflow of the Proposed System

4.2.1. Training Workflow

The training workflow focuses on preparing the dataset and building the speaker detection model (Figure 4.12.). In this project, the AVA-ActiveSpeaker dataset was used. Instead of performing face detection manually, the dataset already provides bounding box annotations and timeframes indicating which person is speaking in each video segment.

During data preparation, the raw videos are segmented based on the given annotations.

Face crops are extracted using the provided bounding boxes, and sequences of 25 consecutive frames are grouped together to form input samples. Each sequence corresponds to a single individual over time, with ground-truth labels indicating whether the person is Speaking or Not Speaking.

The cropped face sequences are then processed by a Convolutional Neural Network (CNN), where models such as ResNet50 and EfficientNetB0 are employed for spatial feature extraction. The CNN outputs high-level visual features that capture relevant facial cues such as lip and jaw motion, cheek movement, and head dynamics.

These features are passed to a Long Short-Term Memory (LSTM) network, which learns the temporal dependencies across the sequence. The LSTM produces a prediction for each sequence, classifying it as Speaking or Not Speaking. The predictions are compared with the provided labels from the dataset to compute the training loss, which is minimized using supervised learning.

4.2.2. Inference Workflow

The inference workflow represents the real-time operation of the system in a robotic environment (Figure 4.13.). In this stage, the system receives a video stream from the robot's camera. Each frame is processed by a YOLOv8 face detector, which identifies all faces in the scene.

The detected faces are then passed to the SORT tracking algorithm, which assigns unique IDs to each individual and ensures consistent tracking across frames. This step is essential for maintaining temporal consistency in multi-person environments.

For each tracked face, a sequence of frames is collected and passed through the CNN + LSTM model trained during the previous phase. The CNN extracts spatial features while the LSTM analyzes temporal dynamics to classify whether the tracked individual is speaking.

The final prediction is used by the camera control module, implemented with OpenCV, to adjust the robot's camera or head orientation. The goal is to keep the active speaker centered in the frame, thereby enabling natural and responsive human–robot interaction.

4.3. Dataset Description (AVA-ActiveSpeaker)

The dataset used in this project is the AVA-ActiveSpeaker dataset, which is one of the most widely used benchmarks for active speaker detection. It was introduced by Chung et al.

(2019) and is specifically designed for tasks that require identifying whether a visible person in a video is speaking.

The dataset is built on top of the AVA video dataset, which consists of movie clips from YouTube. For each video, face bounding boxes are annotated frame by frame, and each bounding box is labeled as either Speaking or Not Speaking. These labels are based solely on visual information, although the dataset also provides synchronized audio for multimodal research.

The AVA-ActiveSpeaker dataset contains:

- 260 movies (approximately 5,000 hours of video).
- 3.65 million labeled face tracks.
- Frame-level annotations at 25 frames per second (fps).
- Each face track is linked with bounding box coordinates and an activity label.

For training in this project, the dataset was processed into sequences of 25 consecutive frames, each sequence corresponding to one person's face. This made it suitable for the CNN + LSTM model, where the CNN extracts features from individual frames, and the LSTM learns temporal dynamics across the sequence.

This dataset was chosen for several reasons:

- It is large-scale and diverse, covering different speakers, backgrounds, and conditions.
- It provides dense annotations (bounding boxes + labels) that make supervised training possible.
- It includes challenging scenarios such as multiple speakers, occlusions, and background noise, which improve model robustness.
- It is widely used in research, enabling comparison with existing works.

In this project, only the visual labels and bounding boxes were used, since the focus is on vision-based active speaker detection without audio input. The bounding boxes provided by the dataset were used to crop face regions, which were then organized into sequences for training. Audio features were not used in this project to maintain compatibility with robotic platforms that may lack microphones.

4.4. Data Preparation

The AVA-ActiveSpeaker dataset provides annotated face bounding boxes and labels at the frame level. To transform this into usable input for the CNN + LSTM model, several preprocessing steps were performed.

- **Face Cropping**: Each video frame comes with bounding box coordinates for every detected face. Cropped face images were extracted from the original frames using these annotations, ensuring the model focused only on the relevant region.
- **Sequence Construction**: Cropped face images were grouped into sequences of 25 consecutive frames, representing approximately one second of video. This design allowed the LSTM to capture temporal dynamics such as lip motion, jaw movement, and head movements. Figure 4.14. illustrates an example: in the top row, the frames show a person actively speaking, while in the bottom row, the frames show another person who is not speaking.

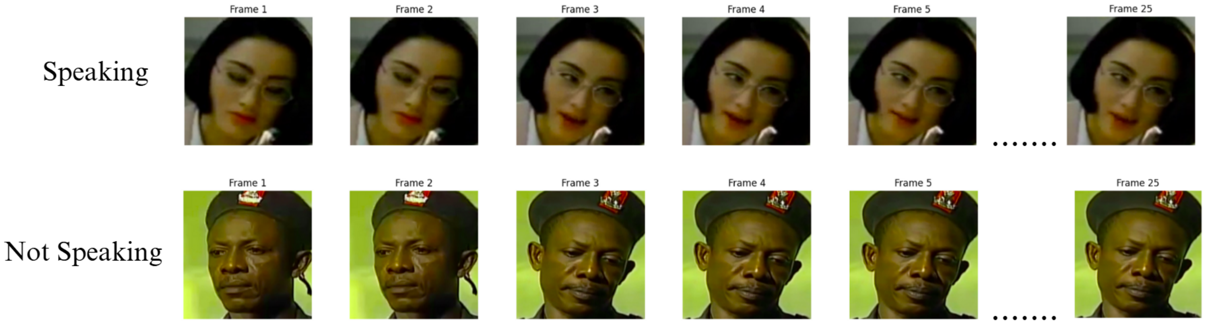


Figure 4.14. Example of 25-frame sequences labeled as “Speaking” (top row) and “Not Speaking” (bottom row).

- **Sequence Filtering and Cleaning**: To improve data quality and ensure consistency, additional filtering steps were applied:
  - **Filtering Short Sequences**: Tracks containing fewer than 25 frames were removed, since they could not form a complete sequence.
  - **Uniform Frame Sampling**: For tracks longer than 25 frames, frames were uniformly sampled to create fixed-length sequences of 25 frames. This ensured consistent input size and avoided bias from longer tracks.
  - **Resolution Filtering**: Sequences containing face crops below 128 pixels in height or width were discarded, since low-resolution images did not provide enough detail for reliable feature extraction.
- **Label Assignment**: Labels from the dataset were aligned with the constructed sequences. Each 25-frame sequence was assigned a single label: Speaking if the majority of frames in the sequence were labeled as speaking, and Not Speaking otherwise.

- **Data Splitting**: The dataset was split into two subsets:
  - Training set (80%): Used for model optimization.
  - Validation set (20%): Used for model evaluation.
- **Data Preprocessing**: Before feeding sequences into the model, standard preprocessing steps were applied:
  - **Resizing**: All cropped faces were resized to 224×224 pixels, matching the input size of pre-trained CNN backbones.
  - **Data Augmentation**: To improve model robustness and reduce overfitting, data augmentation techniques were applied during training. In particular, **random color jittering** was used to introduce variations in brightness, contrast, saturation, and hue, making the model more resilient to changes in lighting conditions and color distribution across different environments. In addition, **horizontal flipping** was applied with a probability of 0.5 to simulate mirrored viewpoints, thereby increasing variability in speaker orientation. To preserve the natural temporal flow of the sequences, augmentations were applied once per sequence rather than individually on each frame, ensuring that all frames in a sequence were transformed consistently. An example is shown in Figure 4.15., where the original sample is transformed into the augmented version.
- **Batch Formation**: Prepared sequences were grouped into mini-batches for efficient GPU training. Each batch contained multiple 25-frame sequences, allowing parallel processing and faster training.

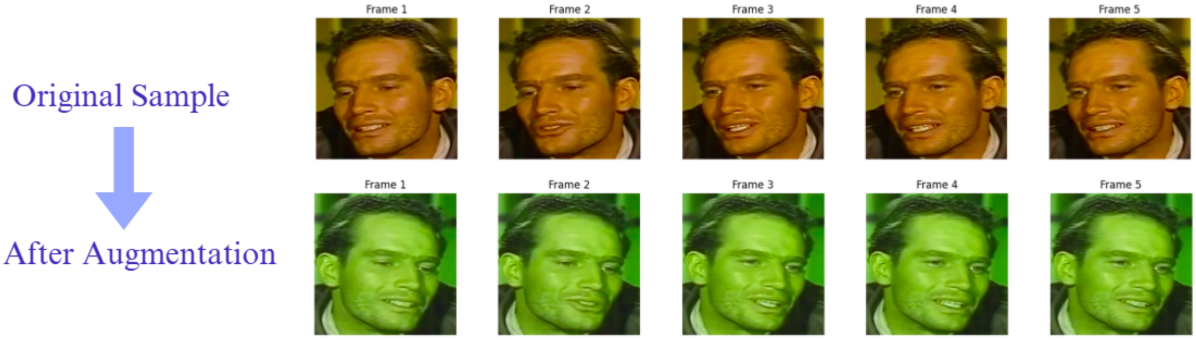


Figure 4.15. Example of data augmentations using random color jittering and horizontal flipping

4.5. Proposed Speaker Detection Model

The overall architecture of the proposed model follows the same workflow illustrated earlier in Figure 4.12. and Figure 4.13., integrating convolutional and recurrent neural networks to identify whether a given face sequence corresponds to a speaking or non-speaking individual. In this section, we describe the model components in greater detail, focusing on the CNN backbone for spatial feature extraction and the LSTM for temporal modeling.

4.5.1. CNN Backbone for Spatial Feature Extraction

The first stage of the model is a Convolutional Neural Network (CNN), which processes each frame independently to extract spatial features. In this project, multiple pre-trained CNN architectures were tested, including:

- ResNet50
- EfficientNetB0

These models were chosen due to their proven effectiveness in computer vision tasks and availability of pre-trained weights on the ImageNet dataset. To take advantage of transfer learning, the early layers were frozen, preserving general-purpose visual features (edges, textures, shapes), while the deeper layers were fine-tuned to adapt to the active speaker detection task.

The CNN backbone outputs a feature vector for each frame, representing spatial cues such as lip movement, jaw motion, cheek deformation, and subtle head or facial expressions.

4.5.2. LSTM for Temporal Modeling

The extracted features from 25 consecutive frames are passed into a Long Short-Term Memory (LSTM) network. LSTMs are a type of recurrent neural network designed to capture long-term dependencies and temporal relationships. In this system, the LSTM models how facial features evolve across time, enabling the detection of patterns specific to speaking behavior, such as repetitive lip movements and coordinated facial dynamics.

By combining the sequential features, the LSTM produces a temporally-aware representation of the face sequence. This representation allows the system to distinguish between a speaking and non-speaking person, even when individual frames alone may not provide sufficient evidence.

4.5.3. Classification Layer

The output of the LSTM is fed into a fully connected classification head with a sigmoid activation function. This final layer outputs a probability score indicating whether the sequence belongs to the “Speaking” or “Not Speaking” class.

4.5.4. Summary of Model Workflow

- Input: A sequence of 25 cropped face images.
- CNN Backbone: Spatial features are extracted frame by frame.
- LSTM Layer: Temporal dynamics across frames are modeled.
- Classification Head: A binary probability is output (Speaking / Not Speaking).

This combination of CNN (spatial) and LSTM (temporal) ensures that both short-term facial features and long-term speaking cues are considered, making the model effective for active speaker detection.

4.6. Model Training and Evaluation

The training and evaluation procedure was carefully designed to ensure that the proposed model could effectively learn the distinction between speaking and non-speaking face sequences. This section describes the main aspects of the training setup, including the loss function, optimizer, hyperparameters, evaluation metrics, and monitoring strategy.

4.6.1. Loss Function

Since the task is a binary classification problem (Speaking vs. Not Speaking), the model was trained using the **Binary Cross-Entropy with Logits Loss (BCEWithLogitsLoss)**. This formulation is more numerically stable than applying a sigmoid activation followed by binary cross-entropy separately, as it combines both steps into one function. The loss is defined as:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \Big[y_i \cdot \log \sigma(z_i) + (1 - y_i) \cdot \log(1 - \sigma(z_i)) \Big]$$

where z_i is the raw logit (pre-activation output of the model), $\sigma(z_i) = \frac{1}{1+e^{-z_i}}$ is the sigmoid function, $y_i \in \{0, 1\}$ is the ground truth label, and N is the batch size.

Using logits directly improves training stability, particularly when predicted probabilities are very close to 0 or 1, reducing numerical underflow or overflow issues.

4.6.2. Optimizer and Learning Rate Scheduling

The model was trained using the **Adam optimizer**, chosen for its efficiency and adaptive learning rates. The initial learning rate was set to 1×10^{-4}. To improve convergence, a **ReduceLROnPlateau** scheduler was employed. Whenever the validation loss plateaued, the learning rate was reduced by a factor of 0.5. This adaptive adjustment helped the model escape shallow minima and improved stability across epochs.

4.6.3. Training Hyperparameters and Regularization

The main hyperparameters were:

- Batch size: 32 sequences per batch
- Sequence length: 25 consecutive frames
- Epochs: 20 (with early stopping)
- Dropout: 0.1 applied to the LSTM input and 0.3 applied after the LSTM layer

Dropout regularization was specifically applied within the LSTM to prevent overfitting and improve generalization.

4.6.4. Training Strategy

Training was conducted with mini-batches of sequences. Early stopping was applied to halt training when validation performance stopped improving, preventing overfitting. Dropout regularization ensured that the LSTM model generalized well to unseen data.

4.6.5. Evaluation Metrics

Model performance was assessed using four key metrics: Accuracy, Precision, Recall, and F1-score. These metrics are widely used in classification problems and provide a balanced view of performance.

- **Accuracy**: Measures the proportion of correctly classified sequences.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

  where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

- **Precision**: Indicates how many of the predicted “Speaking” sequences were actually correct.

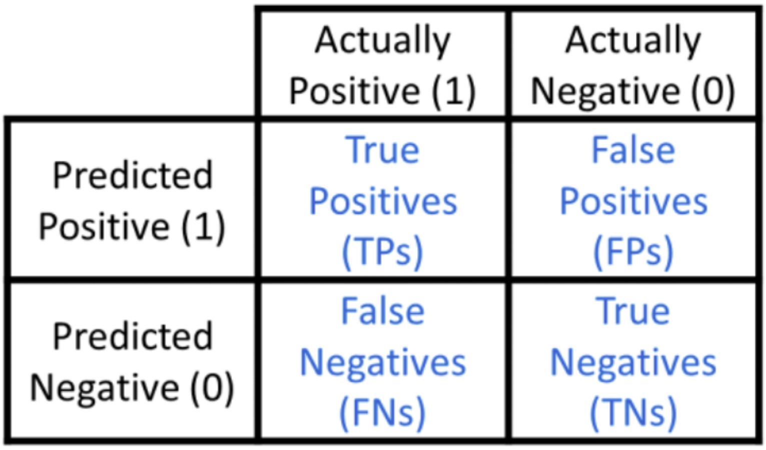
$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall**: Measures the ability of the model to correctly identify actual “Speaking” sequences.

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F1-score**: Represents the harmonic mean of Precision and Recall, providing a balanced measure even with class imbalance.

$$\text{F1-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$



	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Figure 4.16. Confusion matrix illustrating True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) (Draelos [6])

These metrics were computed on test set to evaluate the generalization ability of the proposed model. They were derived from the values of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN), which are summarized in the confusion matrix (Figure 4.16.).

4.6.6. Training Monitoring with Weights & Biases

The training progress was monitored using the **Weights & Biases (W&B)** platform. W&B enabled real-time visualization of loss and accuracy curves, tracking of hyperparameter effects, and systematic comparison across different CNN backbones (ResNet50 and EfficientNetB0). This facilitated experiment management and selection of the most effective configuration.

4.7. Camera Control Module using OpenCV

For the real-time deployment of the speaker detection system, a camera control module was developed using the OpenCV library. OpenCV provides efficient tools for capturing, processing, and managing live video streams, making it suitable for integrating real-time computer vision applications.

The camera module was responsible for initializing the webcam, capturing frames in real time, and passing them into the face detection and tracking pipeline. Video frames were captured using the cv2.VideoCapture() function, which allows flexible control over the camera source (built-in webcam, external USB camera, or IP camera). Frame resolution and frame rate parameters were configured to balance between computational efficiency and detection accuracy, ensuring that the system could operate smoothly on available hardware resources.

To handle continuous video streams, the camera module was designed as a loop that retrieves one frame at a time. Each frame was preprocessed (resized and converted to the required color format) before being passed to the YOLOv8 face detector. This modular design allowed the camera system to remain independent of the detection pipeline, meaning that the same interface could be reused for other real-time computer vision applications in the future.

Additionally, the OpenCV module provided features for user interaction, such as pressing a specific key (e.g., q) to terminate the process, or saving selected frames for debugging purposes. This flexibility simplified both the development and testing phases of the system.

In summary, the OpenCV-based camera control module enabled a seamless connection between real-world video input and the deep learning models, thereby making the speaker detection system functional in a live environment.

V. RESULTS AND DISCUSSIONS

This chapter presents the results obtained from the experiments carried out on the proposed vision-based speaker detection system. The purpose is to evaluate how well the model performs in detecting active speakers using visual cues from face sequences, and to analyze the effectiveness of different CNN backbones when combined with the LSTM architecture.

The results are organized into two main parts: quantitative performance based on evaluation metrics such as accuracy, precision, recall, and F1-score, and qualitative assessment based on sample visual outputs. In addition, the system's performance in real-time deployment is discussed to highlight its practical applicability. The discussion section interprets the findings and compare the results.

5.1. Dataset Split and Experimental Setup

For the experiments, the AVA-ActiveSpeaker dataset was divided into two subsets: a training set and a validation set. The training set was used to optimize the model parameters, while the validation set was used both for hyperparameter tuning and for reporting final performance.

Each sample consisted of 25 consecutive frames of cropped face sequences, providing temporal information to the CNN–LSTM architecture. Frames were resized to a consistent resolution, and sequences not meeting the minimum resolution requirement (128 pixels) were excluded.

All experiments were conducted on an NVIDIA L4 GPU (22 GB VRAM) using Google Colab Pro. Models were implemented in PyTorch, with training runs monitored and logged through Weights & Biases. The batch size was fixed at 32, the learning rate initialized at 0.0001, and optimization performed using Adam. A ReduceLROnPlateau scheduler was applied to halve the learning rate when validation loss plateaued. Dropout layers (0.1 on the LSTM input and 0.3 after the LSTM) were used for regularization. Binary Cross-Entropy with Logits Loss served as the objective function.

This setup ensured that the trained models could be fairly compared across different CNN backbones (ResNet50 and EfficientNetB0) under consistent experimental conditions.

5.2. Model Performance

5.2.1. Evaluation Metrics

The trained models were evaluated using standard classification metrics, namely accuracy, precision, recall, and F1-score, computed on the validation/test set. Table 5.1. summarizes the results obtained for each backbone.

The results indicate that EfficientNetB0 achieves the best overall balance, yielding the highest accuracy (72.81%), precision (71.00%), and F1-score (73.06%). ResNet50, on the other hand, attains a higher recall (79.31%), meaning it is more effective at detecting true speakers, though at the cost of lower precision.

This trade-off has important implications for deployment in robotic systems. If the primary objective is to minimize missed detections, ensuring that the robot almost never fails to recognize an active speaker, ResNet50 may be the more suitable backbone due to its higher recall. However, if a more balanced performance with fewer false positives is preferred, EfficientNetB0 is the better candidate.

Table 5.1. Performance comparison of different CNN backbones combined with LSTM on the speaker detection task.

Backbone	Accuracy	Precision	Recall	F1-score
ResNet50	71.23%	67.65%	79.31%	73.02%
EfficientNetB0	72.81%	71%	75.24%	73.06%

5.2.2. Confusion Matrix

To further evaluate the performance of the ResNet50 backbone combined with the LSTM model, a confusion matrix was generated on the test set (see Figure 5.17.). The confusion matrix provides detailed insights into the distribution of predictions across the two classes: *Speaking* and *Not Speaking*.

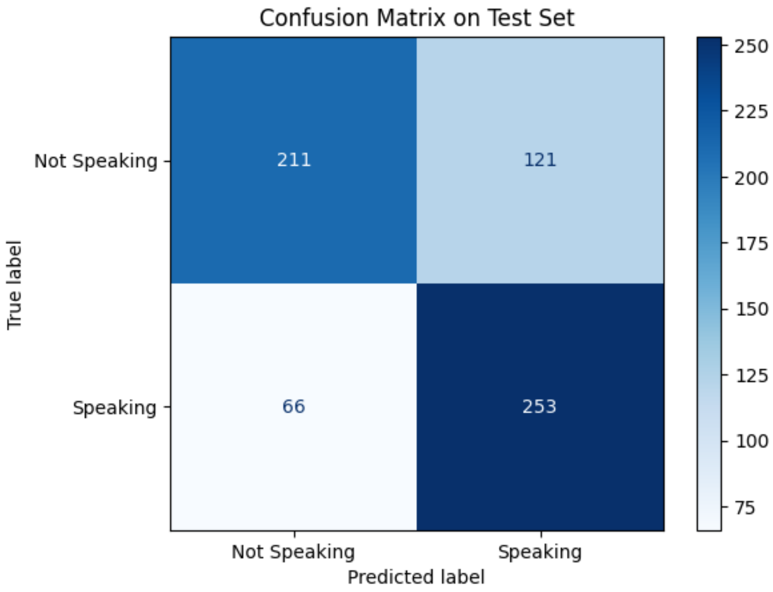


Figure 5.17. Confusion matrix of ResNet50+LSTM on the test set.

As shown in Figure 5.17., the model correctly classified a total of **253 speaking frames** and **211 non-speaking frames**. However, it also produced **121 false positives**, where non-speaking samples were misclassified as speaking, and **66 false negatives**, where speaking samples were misclassified as not speaking.

These results align with the evaluation metrics reported earlier. In particular, the relatively higher recall score reflects the model's ability to detect most speaking samples (low false negatives), while the presence of a moderate number of false positives explains the slightly lower precision. This trade-off suggests that the ResNet50 backbone is more effective when the priority is to avoid missing true speaking segments, which is desirable in scenarios such as speaker tracking or video conferencing assistance.

5.3. Qualitative Results

While numerical metrics provide a global assessment of performance, qualitative examples help illustrate how the model behaves in real-world scenarios. Figure 5.18. shows sample frames where the system correctly identified the active speaker. In these cases, the model successfully detected subtle lip and jaw movements, even when the background was cluttered or other non-speaking faces were present in the scene.

In contrast, Figure 5.19. highlights some failure cases. False negatives were observed when the speaker's face was partially occluded (e.g., by a microphone or hand), when lip movements were minimal, or when the subject was facing away from the camera. False positives occasionally occurred in situations where non-speaking individuals exhibited facial motions resembling speech, such as smiling or chewing.

Despite these limitations, the overall qualitative performance indicates that the proposed CNN–LSTM model generalizes well to diverse speaking conditions. The bounding boxes produced by YOLOv8 combined with SORT tracking ensured stable identification across consecutive frames, reducing the likelihood of switching speaker labels between adjacent time steps.

These qualitative results highlight both the strengths and weaknesses of the system, providing useful insights for future improvements.

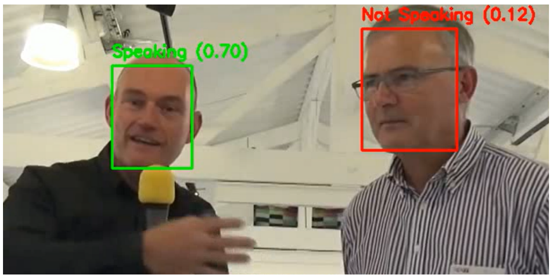


Figure 5.18. Corrected Prediction Frame on Real Video

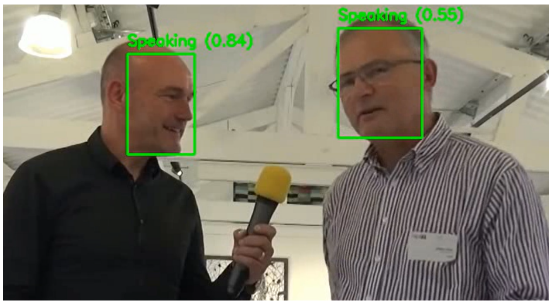


Figure 5.19. False Prediction Frame on Real Video

5.3.1. Discussion

The results highlight the strengths and trade-offs of the different CNN backbones used for feature extraction in the speaker detection pipeline. ResNet50, while achieving slightly lower overall accuracy compared to EfficientNetB0, demonstrated higher recall. This means that the model was better at capturing instances where a person is speaking, which is particularly valuable in real-world applications where missing an active speaker is more critical than occasionally producing false positives.

On the other hand, EfficientNetB0 provided a more balanced performance, achieving higher precision and slightly better overall accuracy. This suggests that it is more effective in reducing false alarms, which can be beneficial in scenarios where frequent misclassification of non-speaking individuals could disrupt downstream processes.

These findings indicate that the choice of backbone should depend on the application context. If the priority is ensuring that active speakers are rarely missed (e.g., in robotic camera control or human–robot interaction), ResNet50 may be more suitable. Conversely, if minimizing false detections is more critical (e.g., in broadcast or meeting transcription systems), Efficient-

NetB0 provides a better trade-off.

VI. CONCLUSION, LIMITATIONS AND FUTURE WORKS

6.1. Conclusion

This internship project focused on the development of a vision-based speaker detection system designed for real-time deployment in robotic platforms. The main objective was to enable a robot to identify the active speaker using only visual cues and adjust its camera view accordingly, without relying on audio inputs.

To achieve this goal, the AVA-ActiveSpeaker dataset was used for training and evaluation. Preprocessing steps such as face cropping, sequence construction, frame filtering, and data augmentation ensured the dataset was prepared in a consistent and robust manner. A CNN–LSTM architecture was designed to capture both spatial and temporal features from face sequences, with different CNN backbones (ResNet50 and EfficientNetB0) tested for performance comparison.

The results demonstrated that EfficientNetB0 achieved better overall performance in terms of precision and F1-score, while ResNet50 showed stronger recall capabilities. The integration of the model into a real-time pipeline using OpenCV, YOLOv8 for face detection, and SORT for face tracking proved that the system can operate at real-time speeds, maintaining a stable and reliable prediction flow. Additionally, the camera control module successfully centered the active speaker within the video frame, illustrating the system's applicability in robotic scenarios and human–robot interaction.

Overall, the project successfully delivered a functional prototype that combines advanced deep learning methods with practical deployment tools. The system demonstrated not only technical feasibility but also real-world potential for enhancing communication in robotics, teleconferencing, and other interactive systems.

6.2. Limitations

Although the proposed system achieved promising results, several limitations were identified during development and testing. These limitations highlight the challenges of relying solely on visual cues for active speaker detection and point toward areas requiring further improvement.

First, the system is sensitive to occlusion and subtle lip movements. When a speaker's mouth is partially blocked by objects such as a microphone, a hand, or even head rotation, the model's accuracy decreases. Similarly, when lip motions are small or speech is slow, the visual cues may be insufficient for reliable detection.

Another limitation lies in the generalization to different environments. The model was trained and tested primarily on the AVA-ActiveSpeaker dataset. While this dataset contains diverse scenes, its coverage of lighting variations, cultural differences in facial expressions, and extreme camera angles is still limited. As a result, performance in completely unseen real-world settings may not fully match experimental results.

From a computational perspective, although the system achieved real-time inference on an NVIDIA L4 GPU, deployment on resource-constrained devices such as embedded boards or mobile robots remains challenging. The CNN–LSTM architecture requires significant memory and processing power, which may not be practical without further optimization.

Finally, the evaluation was conducted using only visual information. While this demonstrates that vision alone can provide valuable cues, the absence of audio input limits the robustness of the system. Multimodal approaches combining both vision and audio could further improve accuracy, especially in noisy or complex environments.

6.3. Future Works

Building on the findings and limitations of this project, several directions for future work can be considered to further improve the proposed system:

- **Improved Robustness to Occlusion and Subtle Movements**: Future work should focus on enhancing robustness when the speaker's face is partially occluded or when lip movements are minimal. This might include using attention-based neural networks or incorporating 3D face modeling to better capture motion cues under challenging conditions.
- **Expanding Dataset Diversity**: To improve generalization, the model should be trained and validated on more diverse datasets that include different lighting conditions, cultural variations in expression, and camera perspectives. Collecting a custom dataset aligned with the robot's operating environment could also strengthen deployment results.
- **Lightweight Deployment on Embedded Devices**: For integration into mobile robots or IoT platforms, model optimization techniques such as pruning, quantization, or knowledge distillation should be explored. These approaches can reduce memory and computation requirements while preserving accuracy.

References

[1] T. Afouras et al. “Deep Audio-Visual Speech Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. 2018.

[2] J. L. Alcázar et al. “MAAS: Multi-Modal Assignation for Active Speaker Detection”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021.

[3] Apple Inc. *Center Stage on iPad Pro*. `https://www.apple.com/ipad-pro/`. 2021.

[4] A. Bewley et al. “Simple Online and Realtime Tracking”. In: *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. 2016, pp. 3464–3468.

[5] J. S. Chung and A. Zisserman. “Lip Reading in the Wild”. In: *Proceedings of the Asian Conference on Computer Vision (ACCV)*. 2016.

[6] A. R. Draelos. *Measuring performance: The confusion matrix*. `https://glassboxmedicine.com/2019/02/17/measuring-performance-the-confusion-matrix/`. 2019.

[7] K. He et al. “Deep Residual Learning for Image Recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.

[8] S. Hochreiter and J. Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[9] G. Jocher et al. *YOLO by Ultralytics*. `https://github.com/ultralytics/ultralytics`. 2023.

[10] G. Fan L. Chen and Y. Zhang. “Speaker Localization and Camera Steering for Telepresence Robots”. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 2258–2264.

[11] L. T. P. Nguyen, Z. Yu, and Y. J. Lee. “LASER: Lip Landmark Assisted Speaker Detection for Robustness”. In: *arXiv preprint arXiv:2501.11899* (2025).

[12] J. Roth et al. “AVA-ActiveSpeaker: An Audio-Visual Dataset for Active Speaker Detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[13] K. Simonyan and A. Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *International Conference on Learning Representations (ICLR)*. 2015.

[14] M. Tan and Q. Le. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”. In: *Proceedings of the International Conference on Machine Learning (ICML)*. 2019, pp. 6105–6114.

[15] R. Tao et al. “Enhancing Real-World Active Speaker Detection with Multi-Modal Extraction Pre-Training”. In: *arXiv preprint arXiv:2404.00861* (2024).

[16] J. Barker Z. Zhang and M. Lincoln. “Robust Multimodal Speaker Detection for Human–Robot Interaction”. In: *IEEE Transactions on Multimedia* 21.10 (2019), pp. 2540–2552.