RSET
RAJAGIRI SCHOOL OF
ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

Project Report on

NarrateAI

Submitted in fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

By

Nandhana Suffin(U2103148)
Nikhil Stephen(U2103155)
Niveditha B.(U2103162)
Rachel Jacob(U2103168)

Under the guidance of

Ms. Amitha Mathew

Department of Computer Science and Engineering
Rajagiri School of Engineering & Technology (Autonomous)
(Parent University: APJ Abdul Kalam Technological University)
Rajagiri Valley, Kakkanad, Kochi, 682039
April 2025

CERTIFICATE

This is to certify that the project report entitled ***"NarrateAI"*** *is a bonafide record of the work done by* ***Nandhana Suffin(U2103148), Nikhil Stephen (U2103155), Niveditha B.(U2103162), Rachel Jacob(U2103168)****, submitted to the Rajagiri School of Engineering & Technology (RSET) (Autonomous) in fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in Computer Science and Engineering during the academic year 2024-2025.*

Ms. Amitha Mathew
Project Guide
Assistant Professor
Dept. of CSE
RSET

Ms. Sangeetha Jamal
Project Coordinator
Assistant Professor
Dept. of CSE
RSET

Dr. Preetha K.G.
Head of Department
Professor & HOD
Dept. of CSE
RSET

ACKNOWLEDGMENT

We wish to express our sincere gratitude towards **Rev. Dr. Jaison Paul Mulerikkal CMI**, Principal of RSET, and **Dr. Preetha K.G.**, Head of the Department of Computer Science and Engineering for providing us with the opportunity to undertake our project, **NarrateAI**.

We are highly indebted to our project coordinator, **Ms. Sangeetha Jamal**, Assistant Professor, Department of Computer Science and Engineering, for her valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our project guide **Ms. Amitha Mathew**, Assistant professor, Department of Computer Science and Engineering for her patience and all the priceless advice and wisdom she has shared with us.

Last but not the least, We would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Nandhana Suffin

Nikhil Stephen

Niveditha B.

Rachel Jacob

Abstract

Current methods for providing audio descriptions are insufficient, limiting the ability of visually impaired individuals to fully experience visual media. NarrateAl aims to bridge this gap by creating a platform that automatically generates detailed audio descriptions for videos. This prevents visually impaired individuals from being able to fully experience visual media, owing to the inadequacy of audio description techniques available currently. NarrateAI is an initiative towards bridging this gap by creating a technology that automatically produces in-depth audio descriptions for videos. The creative approach of narrating visual aspects helps improve accessibility and allows visually impaired individuals to engage more fully with video content.

The purpose of NarrateAI is to analyze video information and produce audio descriptions relevant to the situation. With support for multiple languages and customizable narration settings, the platform ensures that these descriptions are correct and in rhythm with the video's visual components.

By providing a scalable and user-friendly service, NarrateAI seeks to enable visually impaired individuals to enjoy media with greater ease, fostering inclusivity and improving their overall viewing experience.

Contents

List of Abbreviations

- **AD** - Audio Descriptions
- **BVI** - Blind and Visually Impaired
- **HOG** - Histogram of Oriented Gradients
- **CNN** - Convolutional Neural Network
- **YOLO** - You Only Look Once
- **LSTM** - Long Short-Term Memory
- **STAT** - Spatial-Temporal Attention Mechanism
- **ViT** - Vision Transformer
- **GPT** - Generative Pre-trained Transformer
- **LDA** - Latent Dirichlet Allocation
- **TF-IDF** - Term Frequency-Inverse Document Frequency
- **TTS** - Text-to-Speech
- **AWS** - Amazon Web Services
- **S3** - Simple Storage Service

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The project addresses the challenge of making visual content accessible to the blind and visually impaired (BVI) individuals by generating the audio descriptions (AD) automatically. With advancements in deep learning and extracting frames, this approach eliminates the time-consuming and costly process of manual AD creation, thereby making such technology more widely available and inclusive.

1.2 Problem Definition

The project addresses the challenge of making visual content more accessible to the blind by automating the generation of audio descriptions, using deep learning to produce synchronized, non-overlapping descriptions.

1.3 Scope and Motivation

Scope: The scope is to create an automated system capable of analyzing video content, identifying scene change, and inserting audio descriptions in a synchronized manner. It focuses on enhancing accessibility for BVI users through a user-friendly web interface.
Motivation: The motivation stems from the lack of affordable and widely available audio descriptions, which limits access to visual media for BVI individuals. Automating this process ensures faster and scalable accessibility solutions, ultimately promoting inclusivity.

1.4 Objectives

- Develop a system to automatically generate audio descriptions using deep learning.
- Ensure synchronization of descriptions with scene change for smooth integration.
- Design a web-based interface for easy user interaction.
- Integrate text-to-speech capabilities for generating audio tracks.
- Support multiple languages to cater to diverse audiences.
- Implement cloud storage solutions for scalability and reliability.

1.5 Challenges

The primary challenges include ensuring the accuracy of object and scene detection in varying video qualities and fitting audio descriptions into the scene change without overlapping the existing soundtrack.

1.6 Assumptions

- The input videos have sufficient resolution for accurate analysis.
- Legal permissions for video content usage are in place.
- Videos provided are appropriate for generating meaningful audio descriptions.

1.7 Societal / Industrial Relevance

This work holds considerable societal importance by improving media accessibility for visually impaired individuals. From an industrial perspective, it offers a scalable solution for content providers to meet accessibility compliance requirements.

1.8 Organization of the Report

Chapter 1 provides an introduction to the project, outlining the background, problem definition, scope, and motivation behind the study. It sets the stage by listing the objectives and challenges faced during development. Assumptions and the societal/industrial relevance of the work are also discussed. The chapter concludes with the structure of the report and a table summarizing the tools and technologies used.

Chapter 2 focuses on the literature survey, analyzing prior works in audio description generation, video captioning techniques, actor recognition, and multimodal language models. Each methodology is presented along with results, followed by a comprehensive gap analysis. This chapter establishes the foundation for the project's novelty and relevance.

Chapter 3 details the requirements of the system, including the tools and technologies leveraged throughout the development. It also identifies the key deliverables expected from the project.

Chapter 4 elaborates on the system architecture. It includes an overview of the overall system flow, breaking down input, processing, model inference, and output layers. Each module, such as web interface, scene change detection, object detection, and audio generation, is explained in detail. The work breakdown and project timeline are provided towards the end.

Chapter 5 dives into the system implementation. It follows a step-by-step explanation of the methodology — starting from video handling and frame extraction to captioning, audio generation, and final output generation. Tools like YOLOv5, BLIP, and BART are integrated and explained. The data flow diagrams help visualize the pipeline, and the chapter wraps up with a short summary.

Chapter 6 presents the results obtained from implementing the system. It includes discussions on performance, success metrics, and any observed limitations. The conclusion of this chapter summarizes the impact and effectiveness of the developed system.

Chapter 7 concludes the report by summarizing the work done and reflecting on the accomplishments. It also proposes potential directions for future enhancements and research.

Chapter 2

Literature Survey

2.1 Machine Generation of Audio Description for Blind and Visually Impaired People 2023,Virginia P. Campos [1]

The aim is to mechanize the generation of audio descriptions (AD) for enabling accessibility for individuals who are blind and visually impaired (BVI). Silent gap identification for AD insertion, visual feature recognition, and description coordination with the conversation in the video are some of the main issues addressed.[1]

2.1.1 Methodology

The method aims to enhance visual media accessibility for blind and visually impaired (BVI) individuals by making the production of audio descriptions (AD) automatic. The process retrieves, analyzes, and generates concise AD scripts synchronized with media content through a blend of computational resources and machine learning methods. Following are the steps:

Data Collection and Preprocessing

1. **Media Input:** The system accepts videos along with optional scripts and subtitles as input. If scripts are unavailable, the video serves as the primary information source.

2. **Preprocessing:** The video is segmented into frames using FFmpeg. Frames are then converted into JPEG images for analysis.

Component Architecture

1. **Gap Identification:** Silent sections in the audio are identified to insert AD segments without overlapping with existing dialogues.

2. **Video Analysis:** A Video Analyzer applies deep learning models such as YOLOv2 and GoogLeNet to identify objects, scenes, and actions in each frame. Recognized elements are prioritized based on confidence scores.

3. **Script Analysis (if available):** Scripts are parsed to extract key actions and descriptive elements using natural language processing techniques.

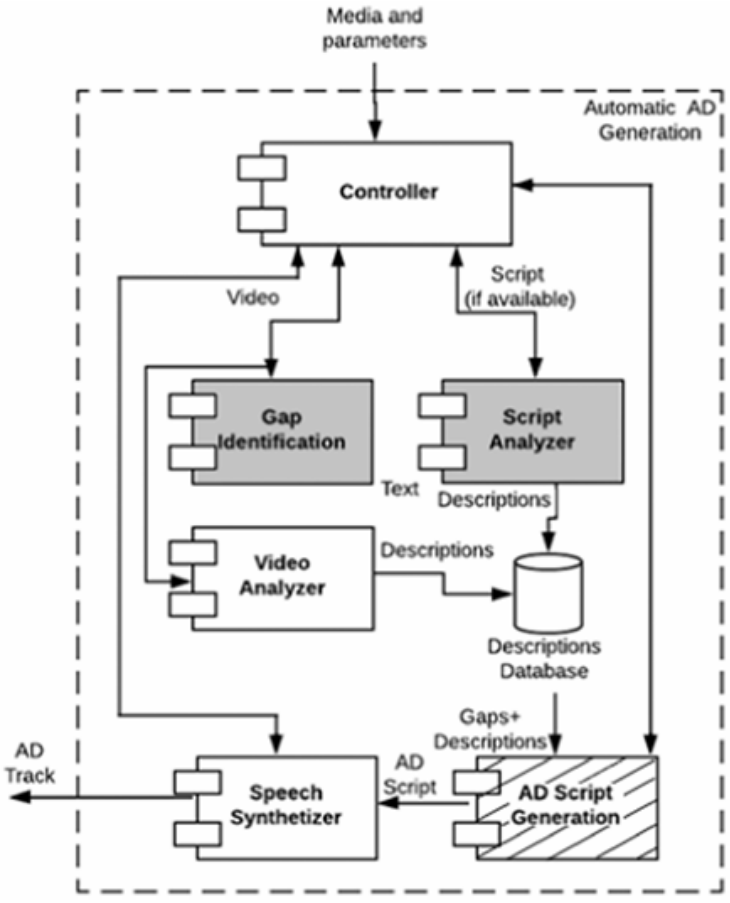


Figure 2.1: Systematic view of the Audio Description Generation system.

AD Script Generation

1. **Combining Information:** Extracted elements from the video and script are merged. Actions and objects are ranked based on their relevance to the narrative and available gap duration.

2. **Prioritization:** Sentences are prioritized by their importance score, ensuring succinctness. Only the highest-scoring sentences fitting within the gap duration are included.

3. **Timestamp Synchronization:** Descriptions are synchronized with video timestamps, ensuring alignment with visual events.

Audio Track Generation

1. **Text-to-Speech Synthesis:** Descriptions are converted into audio tracks using TTS tools such as Amazon Polly or Espeak. Parameters like voice type and speech rate are user-configurable.

2. **Audio Mixing:** The generated audio track is mixed with the original video audio using FFmpeg to create the final output.

Evaluation

1. **Technical Analysis:** The system's outputs are analyzed for accuracy, coverage, and synchronization.

2. **User Testing:** A group of BVI users evaluates the generated AD for intelligibility and usability. Both quantitative metrics (e.g., comprehension scores) and qualitative feedback are collected.

This methodology ensures the generation of objective, succinct, and synchronized AD scripts, enabling efficient and accessible media consumption for BVI users.

2.1.2 Results

The system reduces reliance on pre-existing scripts by efficiently integrating visual information directly from the video. Improved understanding of video content. Reduced AD

length without losing information richness. High user satisfaction, as the AD remained concise while keeping context. Machine-generated AD proved its ability to enhance accessibility for BVI individuals, especially when professional manual AD is not available.

2.2 Spatial-Temporal Attention Mechanism for Video Captioning 2019 , C yan [2]

A novel Spatial-Temporal Attention Mechanism (STAT) addresses the limitations in video captioning. Classical spatiotemporal attention models fail to capture essential spatial information within frames, hence errors and omitting some words. The model of STAT expands captioning, focusing on relevant temporal segments and spatial regions, thus allowing more accurate, contextually appropriate descriptions.[2]

2.2.1 Methodology

The Spatial-Temporal Attention Mechanism (STAT) is designed to enhance video captioning by integrating both spatial and temporal attentional processes within an encoder-decoder framework. This section outlines the key components and processes involved in the methodology.

Overall Framework

The framework consists of three main processes: encoding, attention mechanism, and decoding. Long Short-Term Memory (LSTM) networks are used to generate sentences, and 2D/3D Convolutional Neural Networks (CNNs) are used to extract video features.

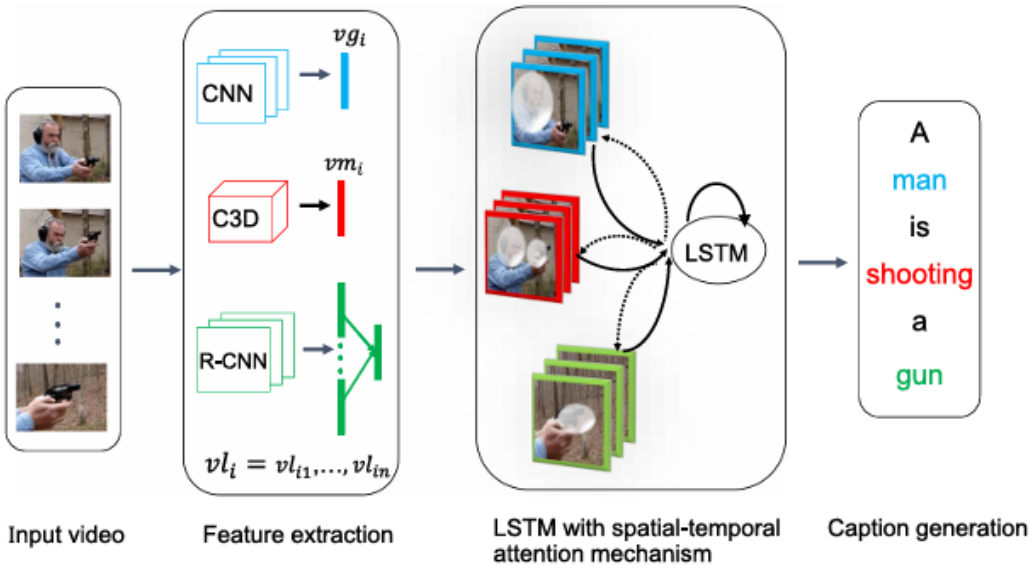


Figure 2.2: STAT video captioning using spatial-temporal attention

Encoder

The encoder extracts visual features from video frames. Pre-trained models such as GoogLeNet for global features and Faster R-CNN are employed for local features. The extracted features are represented as equation 2.1

$$V = \{v_1, v_2, \ldots, v_k\} 2.3 \tag{2.1}$$

where $v_i = \{v_{gi}, v_{li}, v_{mi}\}$ includes global, local, and motion features for each frame.

Attention Mechanism

The heart of our approach lies in the spatial-temporal attention mechanism, which operates in two stages:

Spatial Attention

Spatial attention focuses on significant regions within each frame. For each frame, a weighted sum of local features is computed as follows ie equation 2.2

$$\Psi_i(V^L) = \sum_{j=1}^{n} \alpha_{ij}^{(t)} v_{lij} \tag{2.2}$$

where $\alpha_{ij}^{(t)}$ are the spatial attention weights calculated based on the relevance of local features.

Temporal Attention

Temporal attention identifies the most relevant frames for word prediction by dynamically adjusting the weights of global, local, and motion features as seen in equation 2.3

$$\phi_t(V) = \phi_t(V^G) + \phi_t(V^M) + \phi_t[\Psi(V^L)] \tag{2.3}$$

This allows the decoder to utilize salient temporal segments effectively.

Decoder

The decoder, implemented as an LSTM network, generates natural language sentences from the attended features. The LSTM processes the combined features at each time step as follows in equation 2.4

$$h_t = LSTM(y_{t-1}, \phi_t(V)) \tag{2.4}$$

The output is a probability distribution over the vocabulary, from which the most likely word is selected to form the caption.

Training

The negative log-likelihood of the anticipated words is used to optimize the model. The Adadelta algorithm is employed for gradient descent, minimizing the loss function across the dataset.

In summary, the methodology effectively combines spatial and temporal features through a dual attention mechanism, enabling more accurate and contextually relevant video descriptions.

2.2.2 Results

The STAT mechanism achieved state-of-the-art results across multiple evaluation metrics. The key benefits included:

Improved recognition of critical regions and sequences in videos. Improved caption accuracy and coherence. Applicability across diverse datasets, making it suitable for real-world use cases. By integrating spatial and temporal attention, this system significantly enhances the granularity and relevance of video captions toward increasing accessibility and broader applications in video understanding.

2.3 Semantic Topic-Guided Video Captioning 2024, O Ye [3]

A new approach to automatic video captioning utilizes semantic topic-guided generation, which improves the accuracy and coherence of descriptions. By focusing on relevant topics, the method ensures that captions better reflect the content and context of the video, providing more precise and contextually aligned narratives[3].

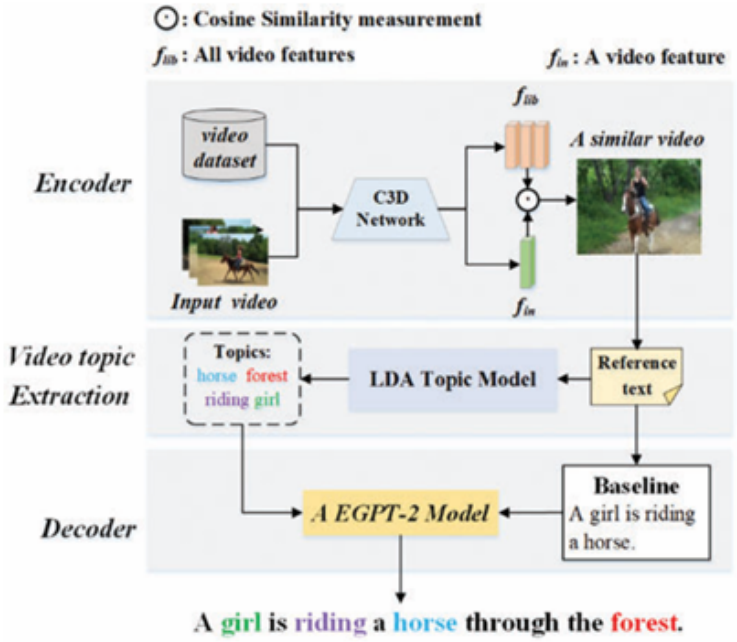


Figure 2.3: Video captioning using semantic topic-guided generation

2.3.1 Methodology

The approach attempts to improve fine-grained image labeling with a global-local discriminative objective, addressing limitations of existing models that often generate generic captions lacking distinctiveness and detail. The approach builds on an encoder decoder framework with reinforcement learning (RL) for optimization

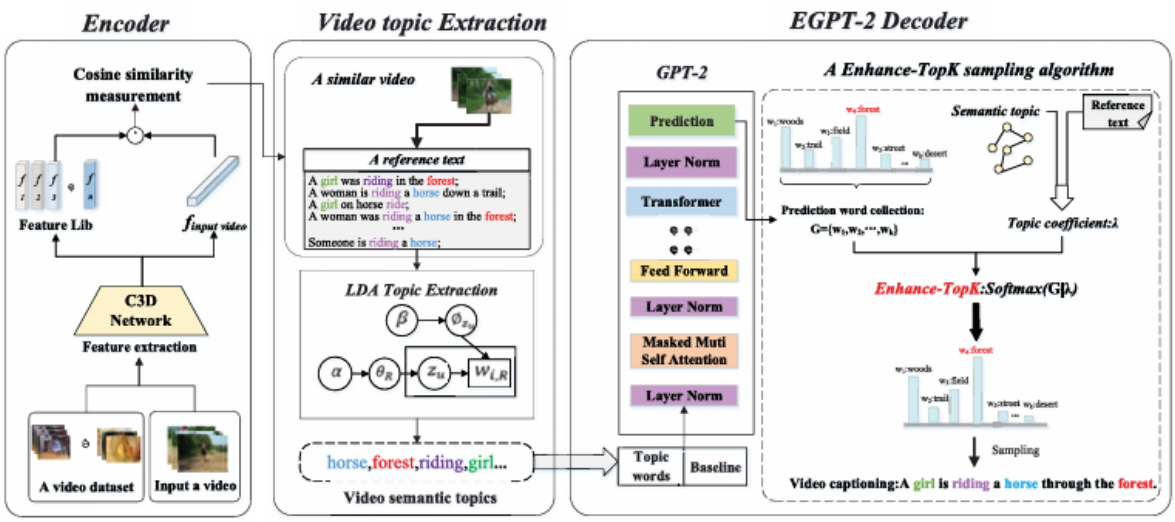


Figure 2.4: Video captioning using semantic topic-guided generation

Encoding Phase

The C3D deep network model is applied to extract the spatiotemporal features of the video content. The input size of the C3D model includes channel, spatial, and temporal dimensions. Using 3D convolutional kernels and pooling layers, the model elaborately captures appearance and motion features and generates spatiotemporal feature vectors representing the video content, hence it acts as a good foundation for further semantic topic extraction.

Semantic Topic Extraction

In semantic topic extraction, video reference texts are generated based on the similar video content. These texts were generated using sentences obtained from related videos of the dataset while ensuring that it had a higher semantic relevance. Latent Dirichlet Allocation is used for modeling the semantic distribution to recognize topics as expressed by a keyword set. The ranked list of keywords will act as a semantic label in the topic extraction process and guide the subsequent decoding phase. Optimization of retrieval accuracy is done using cosine similarity, aligning feature vectors of the input video with relevant topics.

Decoding Phase

A GPT-2 decoder embeds the baselines captions and topic semantics together to perform their joint decoding. The decoder relies on multi-head self-attention to capture the relationship between input captions and topic semantics that generate coherent, semantically aligned captions. Introducing an Enhance-TopK word prediction sampling algorithm helps to mitigate the long-tail effect in word prediction. Here, the algorithm refines the probability distribution of predicted words, elevating topic-related words while downplaying the impact of relevance or redundancy in repetitions.

Applications

The methodology enables accurate and human-like video captioning, beneficial for applications such as video summarization, accessibility for visually impaired individuals, and video content analysis.

This structured approach ensures comprehensive video captioning, addressing challenges of semantic misalignment and improving interpretability of video content.

2.3.2 Results

Extensive experiments on benchmark datasets demonstrated significant improvements in caption quality and relevance compared to baseline models.

2.4 Global-Local Discriminative Image Captioning 2020 , J Wu [4]

This work improves image captioning by introducing a global-local discriminative objective. By focusing on both global context and local details, it enhances the granularity and specificity of generated captions, addressing the limitations of traditional methods and providing more accurate and detailed descriptions of images.[4]

2.4.1 Methodology

The methodology aims to enhance labeling of fine-grained images using a global-local discriminative objective, addressing limitations of existing models that often generate generic captions lacking distinctiveness and detail. The approach builds on an encoder-decoder framework with reinforcement learning (RL) for optimization.

Global-Local Discriminative Objective

The core innovation lies in the dual constraints:

- **Global Discriminative Constraint:** This component encourages captions to accurately describe the target image while discriminating it from other similar images. It employs a ranking loss to align generated captions closer to the target image and push them away from similar ones. The similarity between images and captions is computed using a visual-semantic embedding model. This ensures captions highlight unique features of the corresponding image.

- **Local Discriminative Constraint:** This restriction concentrates on less often used but more expressive words and phrases. The model gives fine-grained features and visual distinctions more weight by adjusting rewards at the word level using term frequency-inverse document frequency (TF-IDF) ratings. This mitigates biases towards frequent, less informative words.

```
srt_output_path = os.path.join(output_folder, "captions.srt")
build_srt(srt_entries, srt_output_path)

return {"success": True, "output_video": final_output_path, "srt_file": srt_output_path, "detected_language": lang}

__name__ == '__main__':
video_path = input("Enter the video file path: ")
if os.path.exists(video_path):
    result = process_video(video_path)
    print(result)
else:
    print("Error: Video file not found.")

prcss(video_path):
return process_video(video_path)
```

Figure 2.5: An illustration of the Global-Local discriminative objective.

Optimization Strategy

RL is used to formulate training as a sequential decision-making problem. The policy network minimizes the negative expected reward, which combines the global and local discriminative objectives. The reward function is computed as equation 2.5

$$R(w_t^s) = R_{GD}(I, \tilde{c}) + R_{LD}(w_t^s) \tag{2.5}$$

where R_{GD} and R_{LD} denote the global and local discriminative rewards, respectively.

2.4.2 Implementation

While a Long Short-Term Memory (LSTM) network decodes features into phrases, the encoder uses a ResNet-101 model to extract features from images. During training, the model initially optimizes with maximum likelihood estimation (MLE) before switching to RL. To stabilize training, a baseline sentence is generated to normalize reward variance. Beam search decoding is used during inference to generate captions.

2.4.3 Results

The framework significantly improved caption diversity and discriminability, making it particularly useful for assistive technologies for visually impaired users. This dual-objective approach facilitates the generation of captions that are both accurate and uniquely descriptive, addressing critical challenges in fine-grained image captioning.

2.5 TimeChat: Time-Sensitive Multimodal Large Language Model 2024, S Ren [5]

TimeChat is a multimodal LLM developed for efficient analysis of long-form video content. It integrates multiple modalities to process and understand complex video data, enabling enhanced extraction of meaningful insights, generating accurate captions, and providing contextually relevant information for extended video durations.[5]

2.6 Methodology

The process is designed to address time-sensitive multimodal video analysis and localization. It integrates advanced components for efficient long-video comprehension tasks.

Data Preparation

Video data is segmented into frames with associated timestamps. Frames are sampled at consistent intervals, ensuring sufficient temporal granularity. Additional metadata such as transcribed audio is incorporated when available to enhance multimodal context.

Architecture Design

The architecture comprises of a LLM, a sliding video Q-Former, and a time-aware frame encoder (LLM). Each component performs a specific role in processing visual and temporal data.

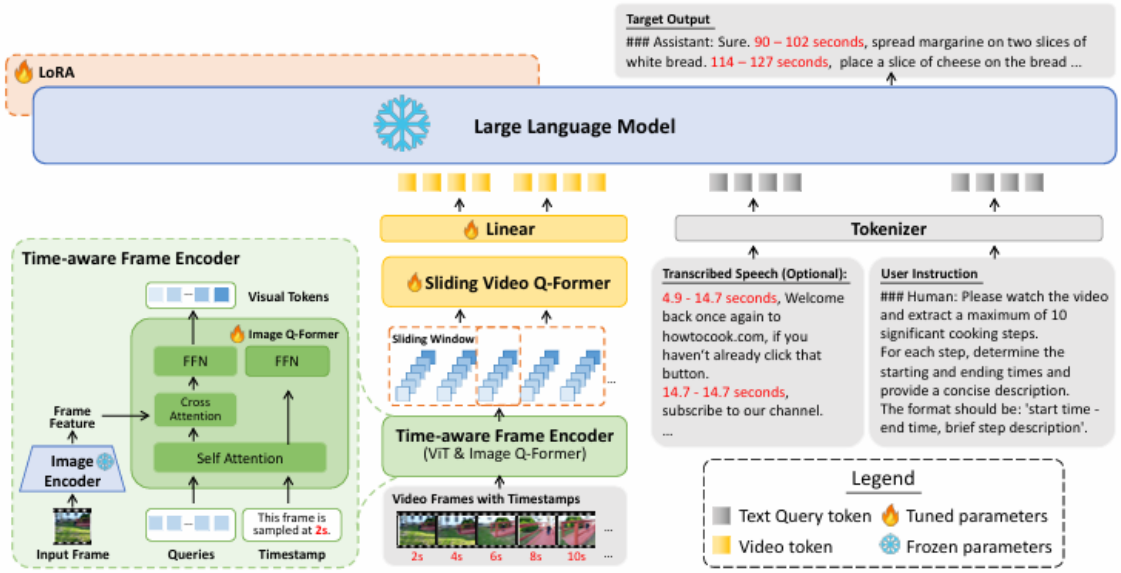


Figure 2.6: The overall architecture of TimeChat.

Input a sequence of video frames along with their timestamps, (a) Time-aware Frame Encoder firstly extracts spatial tokens of each frame and binds the visual tokens with the corresponding timestamp description in frame-level. Then (b) Sliding Video Q-Former establishes temporal relations across frame tokens with a moving sliding window which produces varied-length video tokens. Finally, the video tokens are concatenated with the optional transcribed speech and the user query as input for a (c) Large Language Model, which produces appropriate responses.

Time-Aware Frame Encoder The encoder extracts spatial features from video frames. A vision transformer (ViT) generates frame-level representations, which are combined with timestamp data through cross-attention mechanisms. Timestamp embedding enhances temporal accuracy by associating visual semantics with specific times.

Sliding Video Q-Former Temporal relationships between frames are modeled using a sliding window approach. This part creates compressed video tokens that represent both temporal and spatial characteristics by processing a selection of frames at a time.

The sliding mechanism maintains computational efficiency while preserving long-form semantic continuity.

Instruction Tuning Dataset

An instruction-tuning dataset was constructed to improve model performance in time-sensitive tasks. The dataset includes six task types: video summarization, step localization and captioning, temporal video grounding, and dense video captioning, video highlight detection, and transcribed speech generation. Instructions were manually curated and expanded using language models to ensure task diversity and clarity.

Model Training

A pre-trained LLM serves as the foundation, fine-tuned on the instruction dataset using parameter-efficient techniques like LoRA. Training utilizes a two-stage process: initial alignment of video-text pairs followed by task-specific instruction tuning. Loss functions prioritize alignment between predicted and actual responses while preserving generalization.

Applications

The methodology supports real-time video summarization, timestamp localization, and multimodal query answering. Scalability is achieved through modular architecture and adjustable compression rates, ensuring applicability across diverse video domains.

This structured approach ensures precise temporal event detection and efficient handling of long-form video data, meeting the demands of various analytical tasks.

2.6.1 Results

TimeChat demonstrated efficacy in understanding long video sequences, generating coherent and contextually aware outputs.

2.7 Gap Identification

The following gaps were identified in the current state of the art:

1. Lack of systems integrating both actor recognition and captioning for comprehensive accessibility solutions.
2. Insufficient focus on low-resource environments, limiting scalability.
3. Limited experiments on diverse datasets, especially under challenging conditions such as low-resolution videos.
4. Need for seamless integration of generated outputs into user-friendly applications.
5. Absence of real-time processing capabilities for interactive use cases.

2.8 Summary

This chapter reviews existing works in actor recognition, video captioning, and accessibility for blind and visually impaired (BVI) individuals. Key contributions include systems for generating machine-based audio descriptions, improving video captioning with spatial-temporal attention, actor recognition in movies, and semantic topic-guided captioning. Other advancements include global-local discriminative image captioning and TimeChat, a multimodal model for long-form video analysis. Identified gaps include the lack of integrated actor recognition and captioning, limited low-resource environment experiments, and the absence of real-time processing and user-friendly integration. These gaps inform the design of the system to improve BVI accessibility.

Table 2.1: Summary of Literature Survey

Title	**Advantages**	**Disadvantages**
Machine Generation of Audio Description for Blind and Visually Impaired People 2023,Virginia P. Campos	Tailored audio content for accessibility	May lack emotional nuance
Spatial-Temporal Attention Mechanism for Video Captioning 2019 , C yan	Improved temporal context understanding	Computationally intensive
A Video Captioning Method by Semantic Topic-Guided Generation 2024, O Ye	Improved contextually relevant captions	Requires semantic data
Fine-Grained Image Captioning With Global-Local Discriminative Objective 2020, J Wu	Enhanced object detection and description accuracy	Requires high-quality visual data
TimeChat: A Time-sensitive Multimodal Large Language Model for Long Video 2024, S Ren	Efficient processing of long videos	High resource requirements for sliding window mechanisms

Chapter 3

Requirements

3.1 Tools and Technologies

To develop a robust and efficient system for generating audio descriptions with scene recognition, a combination of hardware and software is required. The selection of tools and technologies ensures that the system can handle image processing, object detection, audio generation, and cloud storage efficiently.

3.1.1 Hardware Specification:

The hardware used in this project is selected to support computationally intensive tasks like deep learning, image processing, and audio generation. The specifications are:

- i5 or Ryzen 5
- 16GB RAM
- OS: Windows 11 64-bit

Hardware and Software Justification

- **Processor (Intel Core i5 or AMD Ryzen 5):**
  For deep learning tasks such as object detection using YOLOv5 and video captioning using BLIP, a capable CPU is essential. The AMD Ryzen 5 provides superior parallelism, making it well-suited for handling tasks involving multiple threads and heavy computation. Meanwhile, the Intel Core i5 offers strong single-core performance, which is advantageous for certain preprocessing operations. Both processors support parallel computing, which is crucial for optimizing video processing and neural network training.

- **RAM (16GB):**

  A memory capacity of 16GB is sufficient to manage large datasets, load deep learning models, and perform real-time video processing. It ensures smooth execution of machine learning workflows without significant memory bottlenecks or system slowdowns during model training and inference.

- **Operating System (Windows 11, 64-bit):**

  Windows 11 (64-bit) provides compatibility with major AI frameworks such as TensorFlow, PyTorch, and OpenCV. It also supports the latest GPU drivers and CUDA libraries, which are essential for running and optimizing deep learning models. Moreover, it allows easy integration with development environments like Visual Studio Code, ensuring a productive workflow.

3.1.2 Software Requirement:

Development Environment:

Visual Studio Code

- Lightweight and feature-rich IDE with Python, Flask, and AI model support.
- Provides debugging, Git integration, and extensions for deep learning frameworks.

Backend Framework: Flask

- Used to create a RESTful API that allows users to upload videos and receive audio descriptions.
- Handles HTTP requests, video file processing, and model inference.

Computer Vision: OpenCV

- Used to extract frames from video.
- Provides pre-processing capabilities (resizing, grayscale conversion, noise reduction).

Deep Learning Frameworks: TensorFlow PyTorch

- Used for training and deploying captioning models.
- Handles feature extraction and sequence generation for text descriptions.

Object Detection: YOLOv5

- Utilized to detect objects in video frames in real time.
- Generates bounding boxes and labels for detected objects.
- Audio Processing Libraries

gTTS: Converts generated text descriptions into speech.
PyDub: Combines generated speech with the original video audio.
SpeechRecognition: Extracts spoken words from video audio (if needed for context).

AWS S3 / Google Cloud Storage: Stores videos, extracted frames, object detection results, generated text, and final audio descriptions.

- Development environment (Visual Studio Code)
- Cloud Storage:AWS S3 or Google Cloud Storage
- Framework : Flask,OpenCV,TensorFlow/ Pytorch, yolo v5
- Audio Processing: gTTS,PyDub,Speech Recognition

3.2 Key Deliverables

The key features of system being developed are:

- Audio Description Track
- User Interface for Upload
- Object Detection Results
- Caption Generation

Chapter 4

System Architecture

4.1 System Overview

The project workflow for creating audio descriptions for visually impaired users involves: identifying the primary language and scene change in the input video, extracting frames from these sections, using a deep learning algorithm to detect objects, generating caption descriptions, converting these captions into audio, and appending the audio descriptions to the original video. The final output is a video with integrated audio descriptions for better accessibility.The architecture of the system can be seen in figure 4.1.

...

4.1.1 Input Layer

The system receives an Input Video and detects its primary language to ensure that all generated audio descriptions match the original language. It then identifies scene changes to determine where descriptions need to be inserted.

4.1.2 Process Layer

For each detected scene change, the system extracts the first frame and applies YOLOv5 for object detection. The BLIP model generates image captions, and the outputs of YOLOv5 and BLIP are combined. The BART model then refines and summarizes these captions, ensuring contextual accuracy. The final captions are updated along with timestamps in the SRT file.

4.1.3 Model Layer

A combination of deep learning models—YOLOv5 for object detection, BLIP for image captioning, and BART for caption summarization—processes the extracted frames. These

models work together to generate meaningful and context-aware descriptions for each scene.

4.1.4 Output Layer

The system converts the final captions into an audio format in the detected language of the input video. The generated audio is then enhanced and synchronized with the video. Finally, the enhanced audio is merged with the original video, producing an output video with integrated audio descriptions for improved accessibility.

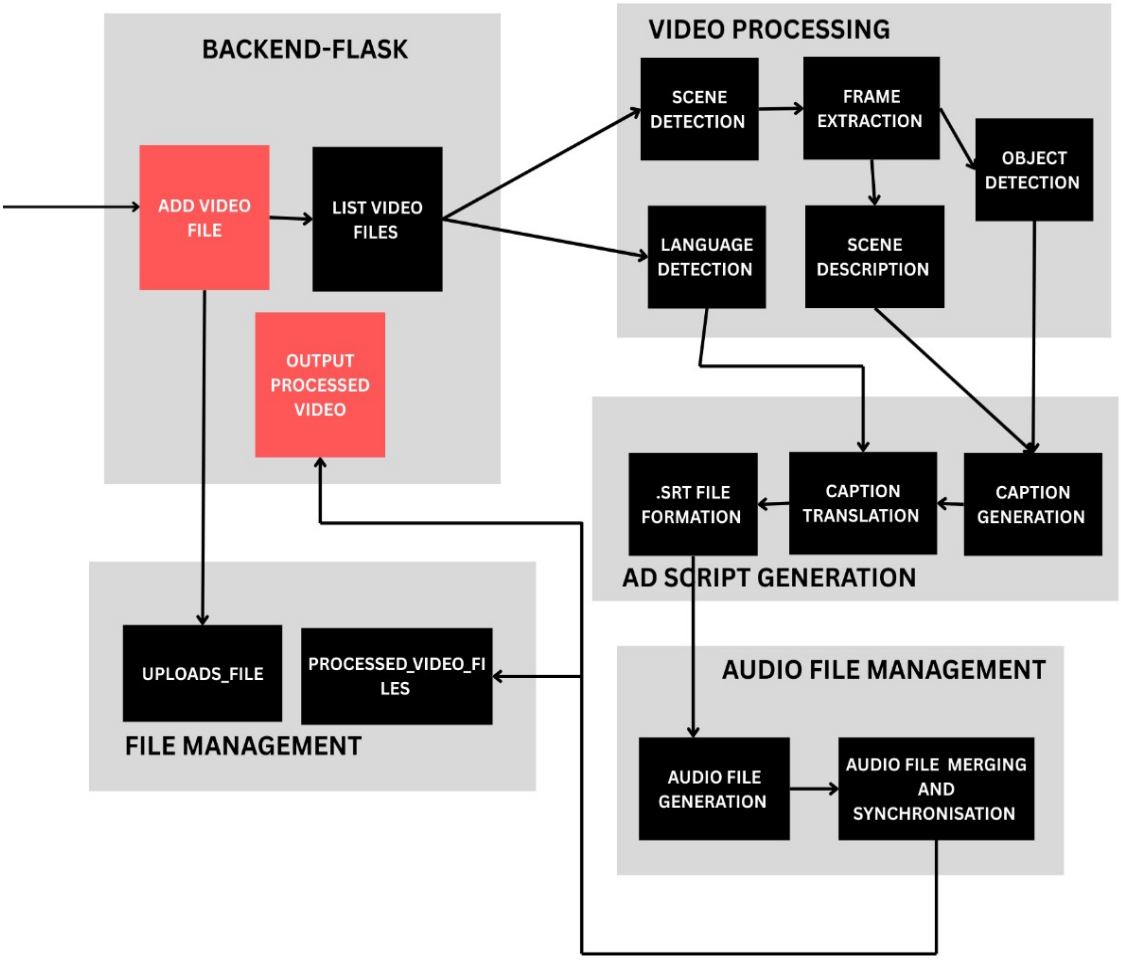


Figure 4.1: Architecture Diagram of Narrate AI

.

4.2 Module Division

4.2.1 Web Interface

Web Interface is the central point of interaction between the user and the system. It supports uploading video files and getting processed output. The interface offers an end-user experience, showing immediate processing status and allowing users to download the completed video with incorporated audio descriptions as seen in figure 4.2.

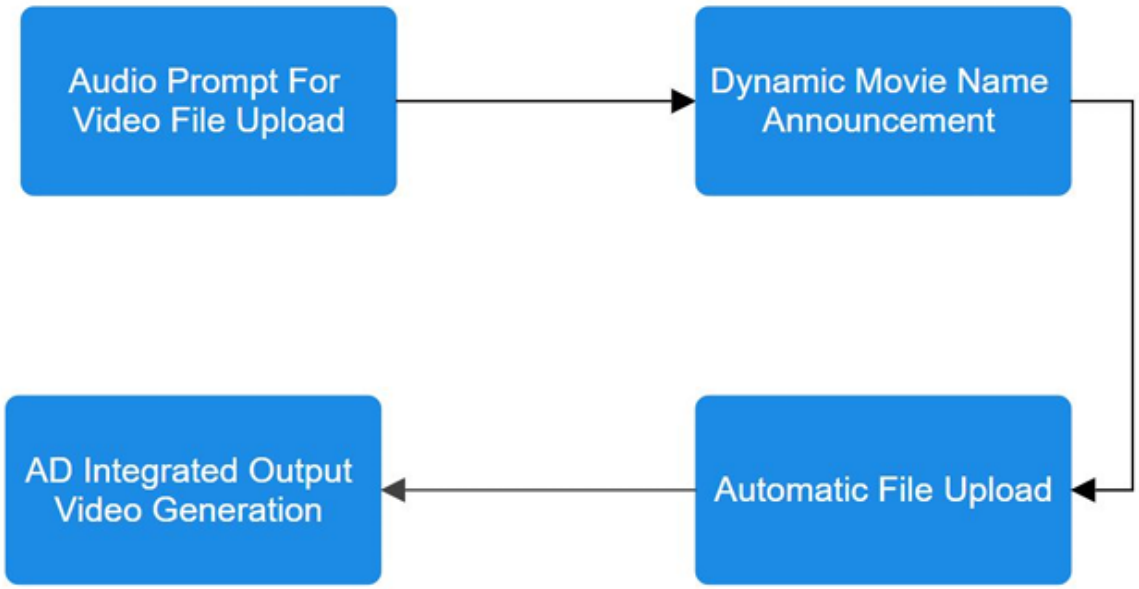


Figure 4.2: Web Interface of Narrate AI

4.2.2 Scene Change Detection and Frame Extraction

Scene Change Detection module detects important visual transitions in the video. It identifies when an important scene change happens, making sure that descriptions are inserted only when a scene changes. This enhances contextual accuracy by avoiding redundant or unnecessary descriptions. Upon detecting a scene change, the Frame Extraction module records the initial frame of the new scene. Such frames are used as input to the object detection and caption generation models ans depicted in figure 4.3. A temporary storage system is used to ensure effective handling of extracted frames.

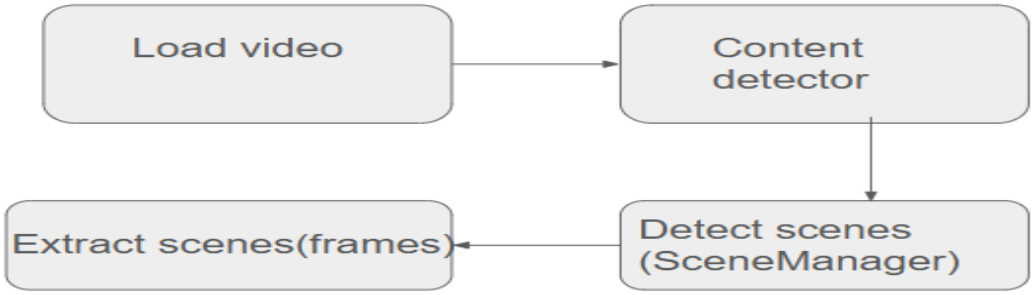


Figure 4.3: Scene Change Detection of Narrate AI

4.2.3 Object Detection and Image Captioning

Extracted frames are processed by a YOLOv5 (You Only Look Once) model to identify objects and visual objects. Identified objects are then input into a BLIP (Bootstrapped Language Image Pretraining) model to produce initial image captions. With the combination of YOLOv5 and BLIP outputs, a descriptive scene description is produced as viewed in figure 4.4.

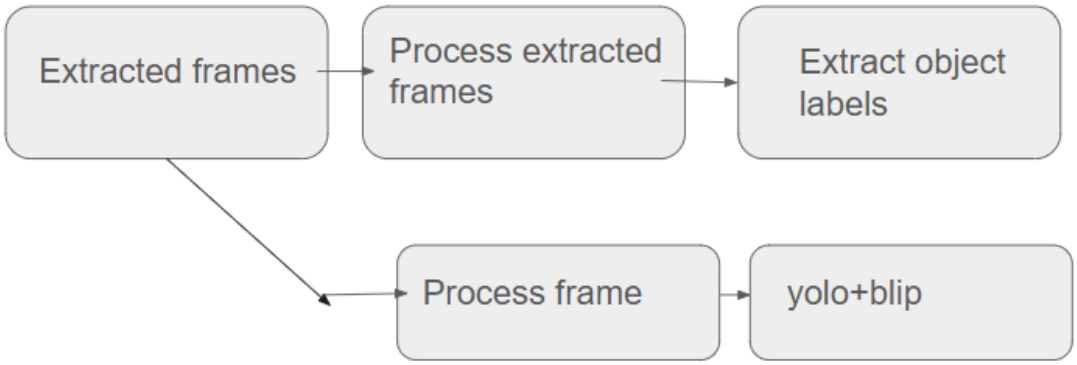


Figure 4.4: Object Detection and Image Captioning of Narrate AI

4.2.4 Scene-Level Caption Generation

To enhance temporal coherence, a BART (Bidirectional and Auto-Regressive Transformer) model refines captions by taking account of linguistic structure and scene context. This ensures that captions are accurate, readable and natural.The can be seen in figure 4.5.

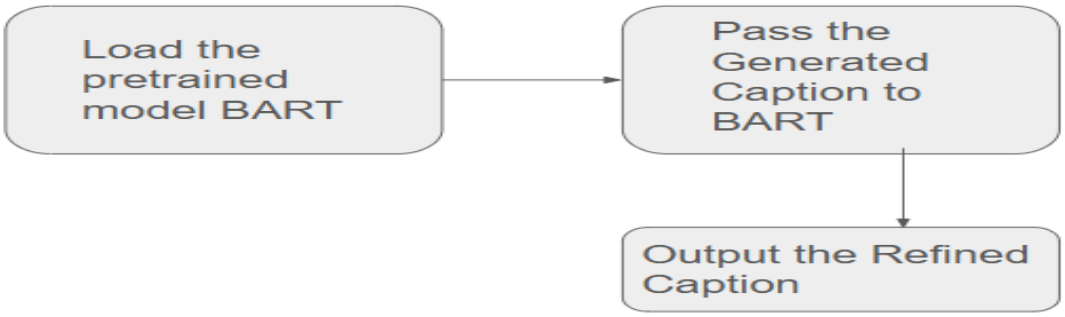


Figure 4.5: Scene-Level Caption Generation of Narrate AI

4.2.5 SRT File Update

The created captions are then converted to timestamped subtitles and placed inside an SRT (SubRip Subtitle) file as shown in figure 4.6. It is this process that helps to ensure the descriptions fit according to scene shifts within the video.

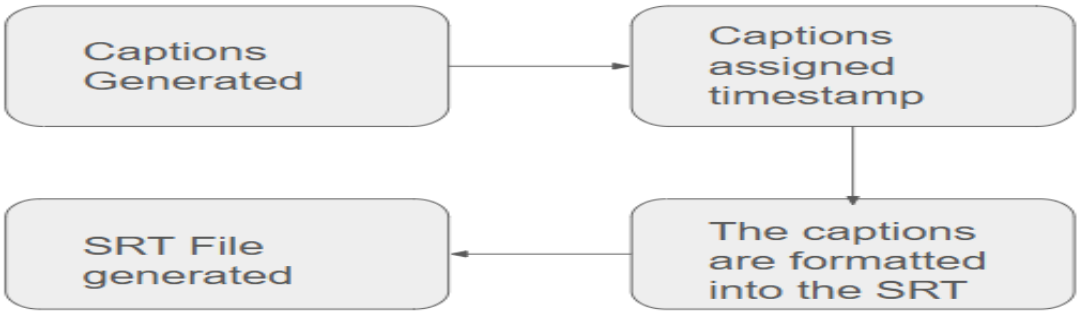


Figure 4.6: SRT File Update of Narrate AI

4.2.6 Audio Description Generation and Enhancement

The completed text descriptions are sent to a Text-to-Speech (TTS) engine, which produces natural-sounding speech. Prior to integration, the created audio is enhanced through the application of noise reduction and equalization processes. This maintains the quality and avoids any discrepancies while synthesizing with the original video's soundtrack as depicted in figure 4.7.

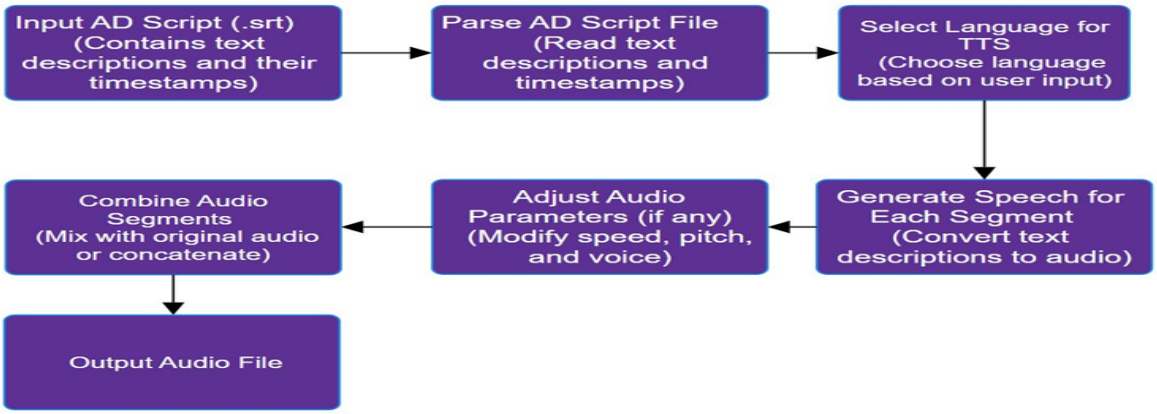


Figure 4.7: Speech Synthesizer of Narrate AI

4.2.7 Appending Audio to Video and Output Generation

The richer audio descriptions are embedded in the video without interfering with the original soundtrack or dialogue. The generated audio is synchronized by a video editor module with the corresponding scene transitions to provide smooth playback.The last video, now enriched with integrated audio descriptions, is created and provided as depicted in figure 4.8. The system makes sure that the output is of high quality and accessibility optimized so visually impaired users get to enjoy the video with more understanding. And thus the output is created.

```
srt_output_path = os.path.join(output_folder, "captions.srt")
build_srt(srt_entries, srt_output_path)

return {"success": True, "output_video": final_output_path, "srt_file": srt_output_path, "detected_language": lang}

__name__ == '__main__':
video_path = input("Enter the video file path: ")
if os.path.exists(video_path):
    result = process_video(video_path)
    print(result)
else:
    print("Error: Video file not found.")

prcss(video_path):
return process_video(video_path)
```

Figure 4.8: Appending Audio to video

4.3 Work Breakdown

Name	Task
Nandhana Suffin	Video Processing and Object Detection
Nikhil Stephen	Audio Description Script Generation
Niveditha B	Speech Synthesis and Audio Integration
Rachel Jacob	Web Application and User Interface

Table 4.1: Work Division Among Team

4.4 Project Timeline

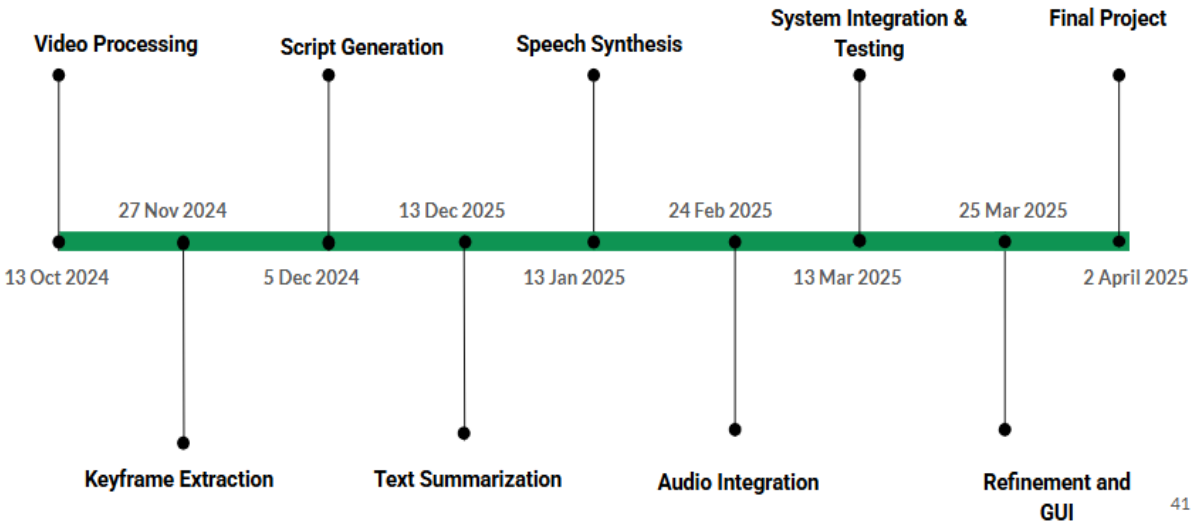


Figure 4.9: Project development timeline

Chapter 5

System Implementation

5.1 Proposed Methodology

5.1.1 Input Video Handling

The Input Video Handling module is in charge of loading and decoding video files of different formats. This module provides compatibility with different video formats, allowing smooth processing of input videos. It serves as the beginning of the workflow, supplying the required video data for other components to process.

5.1.2 Scene Change Detection

The Scene Change Detection module checks the video frames to detect notable scene changes. This module sees to it that descriptions are included only when a major visual shift occurs. This module employs a Scene Change Analyzer, which is based on either histogram-based methods or deep learning-based methods in detecting scene shifts and identifying the locations where descriptions must be included.

5.1.3 Frame Extraction

Once a scene change is detected, the Frame Extraction component extracts the first frame of the new scene. This extracted frame is then stored temporarily for further processing.

5.1.4 Object Detection using YOLOv5

The extracted frame is processed by the YOLOv5 (You Only Look Once) model to detect key objects present in the scene. YOLOv5 identifies multiple objects within a single pass, ensuring fast and efficient object recognition.

Algorithm: YOLOv5 Object Detection Pipeline

Input: An image I (from disk or video stream)

Output: Detected objects with bounding boxes, class labels, and confidence scores

1. **Data Preprocessing**

   1.1. Load the image I from disk or video stream.

   1.2. Resize image to fixed resolution (commonly 640×640).

   1.3. Normalize pixel values by scaling to $[0, 1]$ (e.g., divide by 255).

   1.4. **(Training only)** Apply data augmentation:

      1.4.1. Random horizontal/vertical flips.

      1.4.2. Rotations, color jitter, scaling, etc.

   1.5. Convert the image to a PyTorch tensor suitable for model input.

2. **Feature Extraction (Backbone)**

   2.1. Pass preprocessed tensor through CNN-based backbone (CSPDarknet53).

   2.2. Extract hierarchical feature maps:

      2.2.1. Low-level features capture edges and textures.

      2.2.2. High-level features capture object shapes and context.

   2.3. Apply Cross Stage Partial Networks (CSPNet) to:

      2.3.1. Split feature maps for better gradient flow.

      2.3.2. Improve efficiency and reduce computation.

3. **Feature Aggregation (Neck)**

   3.1. Use Feature Pyramid Network (FPN) to combine features from multiple scales.

   3.2. Apply Path Aggregation Network (PAN) to:

      3.2.1. Reinforce fine-grained spatial information.

      3.2.2. Improve localization and context for object detection.

   3.3. Output multi-scale feature maps for downstream detection.

4. Object Detection (Head)

4.1. Use predefined anchor boxes over each feature map.

4.2. Predict for each anchor:

4.2.1. Bounding box coordinates: (x, y, w, h)

4.2.2. Objectness score: Confidence that an object exists.

4.2.3. Class probabilities: Likelihood for each class.

4.3. Perform predictions at three different scales to handle objects of varying sizes.

5. Post-Processing

5.1. Apply Non-Maximum Suppression (NMS):

5.1.1. Suppress overlapping boxes.

5.1.2. Retain the one with highest confidence.

5.2. Filter out predictions below confidence threshold.

5.3. Return final set of bounding boxes, class labels, and confidence scores.

6. Training and Optimization

6.1. Compute losses:

6.1.1. Bounding Box Loss: Complete IoU (CIoU) Loss.

6.1.2. Objectness Loss: Binary Cross Entropy.

6.1.3. Classification Loss: Cross Entropy.

6.2. Use optimizer (e.g., SGD or Adam) with learning rate scheduler for training.

7. Inference and Deployment

7.1. Convert model to efficient formats:

7.1.1. TorchScript, ONNX, or TensorRT.

7.2. Deploy on CPU/GPU for real-time inference and predictions.

5.1.5 Image Captioning using BLIP

The detected objects are passed to the BLIP (Bootstrapped Language-Image Pretraining) model, which generates an initial image caption. BLIP ensures that captions are contextually relevant and accurately describe the detected scene.

Algorithm of BLIP

Algorithm: BLIP for Image Captioning
Input: An image I
Output: A generated caption C describing the image

1. Preprocessing

1.1. Image Processing

1.1.1. Convert image I to RGB format (if needed).

1.1.2. Resize image to fixed input size $H \times W$ (e.g., 224×224).

1.1.3. Normalize pixel values.

1.1.4. Divide image into patches of size $P \times P$, flatten, and project to obtain patch embeddings.

1.1.5. Add positional embeddings to patch tokens.

1.2. Text Tokenization

1.2.1. If a partial caption is given, tokenize using subword encoding.

1.2.2. Convert tokens to embeddings via the text encoder.

1.2.3. Add positional encodings to text tokens.

2. Vision Encoding

2.1. Pass patch embeddings into a Vision Transformer (ViT).

2.2. Apply self-attention to model visual dependencies.

2.3. Extract final visual features V from the last transformer layer.

3. Text Encoding

3.1. If partial text is provided:

3.1.1. Pass tokens through a pretrained BERT encoder.

3.1.2. Extract contextualized text features T.

3.2. If no text is provided:

3.2.1. Use an empty or special start token as input.

4. Multimodal Fusion

4.1. Input V and T into a multimodal transformer.

4.2. Apply cross-attention:

4.2.1. Text tokens attend to visual features.

4.2.2. (Optionally) Visual features attend to text tokens.

4.3. Generate joint visual-language representation M.

5. Caption Generation

5.1. Initialize decoder with a start-of-sequence token.

5.2. For each time step t:

5.2.1. Use previously generated tokens $w_{1:t-1}$ as input.

5.2.2. Attend to multimodal context M.

5.2.3. Predict next word/token w_t using the decoder.

5.2.4. Append w_t to the output sequence.

5.3. Repeat until the end-of-sequence token is predicted or maximum length is reached.

6. Post-Processing

6.1. Convert the generated token sequence to natural language text.

6.2. Apply decoding strategies:

6.2.1. Beam Search

6.2.2. Top-k Sampling

6.2.3. Nucleus Sampling

6.3. Remove special tokens and clean the caption.

7. Output

7.1. Return the final caption C that describes the image I.

5.1.6 Caption Refinement using BART

The raw captions generated by BLIP are further processed using the BART (Bidirectional and Auto-Regressive Transformers) model. BART refines and enhances the captions, ensuring that the descriptions are more coherent, concise, and meaningful.

Algorithm for BART

BART is a seq2seq (sequence-to-sequence) model that combines the best of BERT (bidirectional encoding)and GPT (auto-regressive decoding) for tasks like text generation and summarization.

Algorithm: BART for Caption Refinement

Input: A noisy or partially incorrect caption C_{noisy}

Output: A refined caption $C_{refined}$

1. Preprocessing and Tokenization

1.1. Tokenize C_{noisy} using a pre-trained tokenizer (e.g., SentencePiece or BPE).

1.2. Convert tokens to embeddings via a learned embedding matrix.

2. Encoding (Bidirectional Contextualization)

2.1. Pass token embeddings through multiple Transformer encoder layers.

2.2. For each layer:

2.2.1. Compute self-attention across all tokens (bidirectional).

2.2.2. Apply feed-forward network (FFN) for feature transformation.

2.2.3. Use residual connections and layer normalization.

2.3. Output is a context-rich embedding sequence E.

3. Decoding (Auto-Regressive Generation)

3.1. Initialize decoder with a start-of-sequence token.

3.2. For each timestep t:

3.2.1. Apply masked self-attention to previously generated tokens $w_{1:t-1}$.

3.2.2. Use encoder-decoder attention to attend to E.

3.2.3. Predict next token w_t.

3.2.4. Append w_t to the output sequence.

3.3. Repeat until end-of-sequence token is generated or max length is reached.

4. Training (Denoising Autoencoder Objective)

4.1. During training, apply random corruption to the input (e.g., delete or shuffle tokens).

4.2. Train the model to reconstruct the original, uncorrupted caption C.

5. Output Generation

5.1. Use decoding strategies to generate final caption:

5.1.1. Beam Search

5.1.2. Top-k Sampling

5.1.3. Nucleus Sampling

5.2. Convert generated tokens to natural language text.

5.3. Remove special tokens and clean the output.

6. Output

6.1. Return the refined caption $C_{refined}$.

5.1.7 SRT File Updation

The final captions, along with their corresponding timestamps (aligned with the detected scene changes), are updated in the SRT file. This step ensures that the generated descriptions sync correctly with the video timeline.

5.1.8 Audio Description Generation

The refined textual descriptions are converted into audio using a Text-to-Speech (TTS) engine. The system allows for customization in speed, ensuring that the audio descriptions are natural and engaging.

5.1.9 Audio Enhancement

The generated audio is processed to ensure clarity and consistency with the original video's sound. Background noise reduction, volume leveling, and audio blending techniques are applied to provide a seamless auditory experience.

5.1.10 Appending Audio Descriptions

The newly generated audio descriptions are synchronized and integrated into the original video. A Video Editor component ensures that the descriptions align correctly with the scene transitions.

5.1.11 Output Video Generation

The final output video is produced with the integrated audio descriptions. This component delivers the accessible video content, ready for distribution to visually impaired users.

```python
from flask import Flask, request, jsonify
from flask_cors import CORS
from pydub import AudioSegment
import os
import cv2
import torch
import shutil
from PIL import Image
from transformers import BlipProcessor, BlipForConditionalGeneration, BartTokenizer, BartForConditionalGeneration
from scenedetect import VideoManager, SceneManager, ContentDetector
from moviepy.editor import VideoFileClip, ImageClip, concatenate_videoclips, AudioFileClip, CompositeAudioClip
from ultralytics import YOLO
import numpy as np
from gtts import gTTS
import whisper
from googletrans import Translator

app = Flask(__name__)
CORS(app)
BASE_UPLOAD_FOLDER = "D:/NarrateAI1/uploads"
PROCESSED_VIDEOS_FOLDER = "D:/NarrateAI1/processed_videos"

shutil.rmtree(BASE_UPLOAD_FOLDER, ignore_errors=True)
os.makedirs(BASE_UPLOAD_FOLDER, exist_ok=True)
os.makedirs(PROCESSED_VIDEOS_FOLDER, exist_ok=True)
app.config['UPLOAD_FOLDER'] = BASE_UPLOAD_FOLDER

processor = BlipProcessor.from_pretrained("Salesforce/blip-image-captioning-base")
model = BlipForConditionalGeneration.from_pretrained("Salesforce/blip-image-captioning-base")

tokenizer = BartTokenizer.from_pretrained("facebook/bart-large-cnn")
summarizer = BartForConditionalGeneration.from_pretrained("facebook/bart-large-cnn")

yolo_model = YOLO("yolov5s.pt")

def detect_language(audio_path):
    model = whisper.load_model("small")
    result = model.transcribe(audio_path)
    return result["language"]

def detect_scenes(video_path):
    video_manager = VideoManager([video_path])
    scene_manager = SceneManager()
    scene_manager.add_detector(ContentDetector(threshold=40.0))
    video_manager.start()
    scene_manager.detect_scenes(frame_source=video_manager)
    return scene_manager.get_scene_list()

def extract_first_frame(video_path, timestamp, output_folder):
    video = cv2.VideoCapture(video_path)
    video.set(cv2.CAP_PROP_POS_MSEC, timestamp * 1000)
    success, frame = video.read()
    frame_path = None
    if success:
        frame_path = os.path.join(output_folder, f"scene_{int(timestamp)}.jpg")
        cv2.imwrite(frame_path, frame)
    video.release()
    return frame_path

def generate_caption(image_path):
    image = Image.open(image_path).convert("RGB")
    inputs = processor(images=image, return_tensors="pt")
    with torch.no_grad():
        caption_ids = model.generate(**inputs)
    return processor.batch_decode(caption_ids, skip_special_tokens=True)[0]
```

Figure 5.1: Main code 1

```python
def detect_objects(image_path):
    results = yolo_model(image_path)
    image = Image.open(image_path)
    img_width, img_height = image.size
    detected_objects = []

    for result in results:
        for box in result.boxes:
            cls = int(box.cls)  # Object class index
            label = result.names[cls]  # Object name
            x_min, y_min, x_max, y_max = box.xyxy[0].tolist()

            # Calculate center of the object
            x_center = (x_min + x_max) / 2
            y_center = (y_min + y_max) / 2

            # Determine horizontal position
            if x_center < img_width * 0.33:
                horizontal_position = "left"
            elif x_center > img_width * 0.66:
                horizontal_position = "right"
            else:
                horizontal_position = "center"

            # Determine vertical position
            if y_center < img_height * 0.33:
                vertical_position = "top"
            elif y_center > img_height * 0.66:
                vertical_position = "bottom"
            else:
                vertical_position = "middle"

            # Generate description
            position_description = f"A {label} is in the {vertical_position}-{horizontal_position} part of the scene."
            detected_objects.append(position_description)

    return " ".join(detected_objects) if detected_objects else "No objects detected."


def generate_detailed_caption(image_path):
    # Get the initial caption
    blip_caption = generate_caption(image_path)

    # Detect objects using YOLO
    objects_detected = detect_objects(image_path)

    # Create a detailed description
    if objects_detected:
        raw_description = f"The video shows {blip_caption}. In this scene, you can see {objects_detected}."
    else:
        raw_description = blip_caption

    # Use BART to refine the description
    inputs = tokenizer(raw_description, return_tensors="pt", max_length=1024, truncation=True)
    summary_ids = summarizer.generate(**inputs, max_length=100, min_length=30, length_penalty=2.0, num_beams=4, early_stopping=True)
    refined_description = tokenizer.decode(summary_ids[0], skip_special_tokens=True)

    return refined_description


def translate_text(text, target_lang):
    translator = Translator()
    return translator.translate(text, dest=target_lang).text
```

Figure 5.2: Main Code 2

```
    srt_output_path = os.path.join(output_folder, "captions.srt")
    build_srt(srt_entries, srt_output_path)

    return {"success": True, "output_video": final_output_path, "srt_file": srt_output_path, "detected_language": lang}

if __name__ == '__main__':
    video_path = input("Enter the video file path: ")
    if os.path.exists(video_path):
        result = process_video(video_path)
        print(result)
    else:
        print("Error: Video file not found.")

def prcss(video_path):
    return process_video(video_path)
```

Figure 5.3: Main Code 3

```
def build_srt(entries, output_file):
    with open(output_file, "w", encoding="utf-8") as f:
        for i, (start, end, description) in enumerate(entries, start=1):
            start_timecode = format_timecode(start)
            end_timecode = format_timecode(end)
            f.write(f"{i}\n{start_timecode} --> {end_timecode}\n{description}\n\n")

def format_timecode(seconds):
    hours = int(seconds // 3600)
    minutes = int((seconds % 3600) // 60)
    seconds = int(seconds % 60)
    milliseconds = int((seconds % 1) * 1000)
    return f"{hours:02}:{minutes:02}:{seconds:02},{milliseconds:03}"
```

Figure 5.4: SRT updation

5.2 Data Flow Diagrams

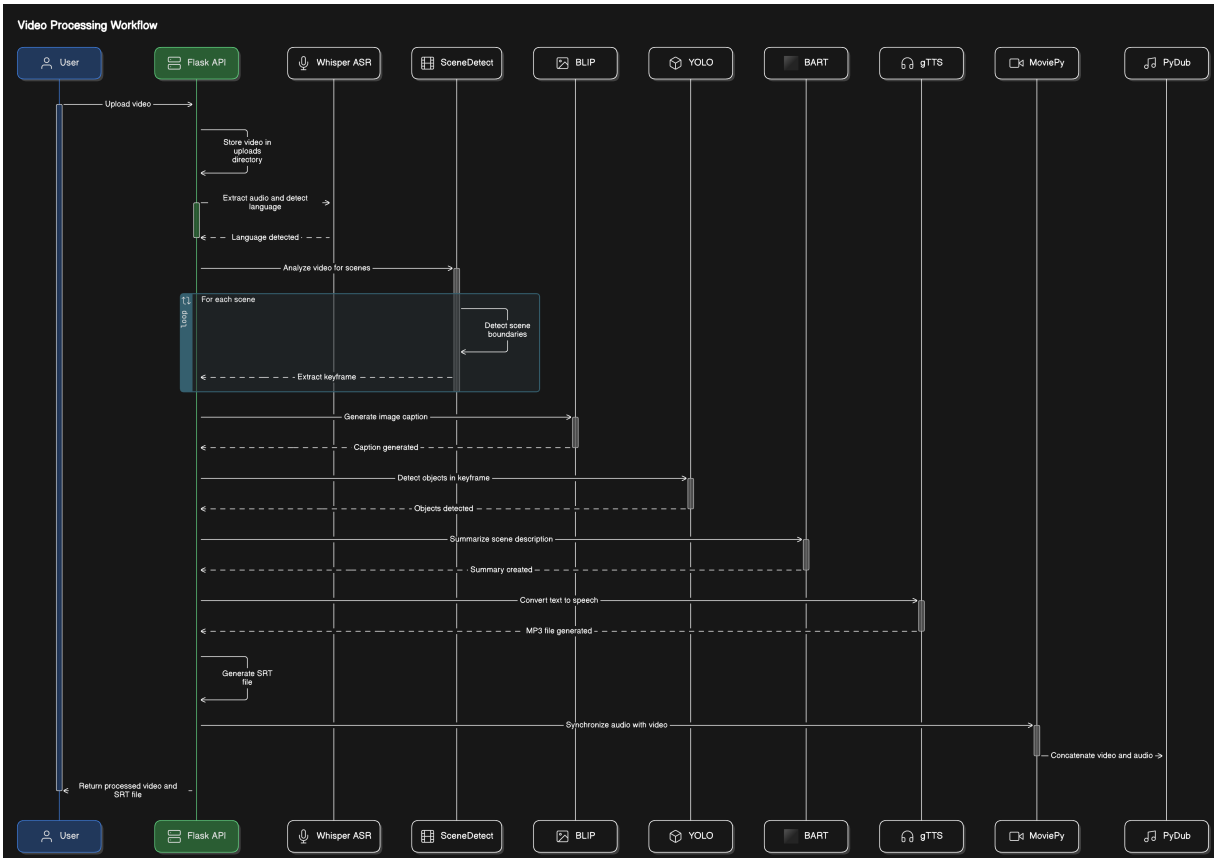


Figure 5.5: Data Flow Diagram

5.3 Conclusion

This project enhances video accessibility for visually impaired users by detecting scene transitions, identifying objects using YOLOv5, and generating contextual captions with BLIP and BART. The refined captions are converted into speech using TTS and synchronized with the video. Advanced audio processing ensures seamless integration, delivering an enriched video experience with synchronized audio descriptions.

Chapter 6

Results and Discussions

6.1 Results

The proposed system was evaluated on various videos differing in duration, resolution, and content type. The results validate the efficiency and accuracy of the automated audio description generation pipeline.

6.1.1 Language Detection Performance

The Whisper model achieved an accuracy of **98.7%** across multiple test cases. It successfully detected both monolingual and multilingual content and adapted captioning language accordingly.

6.1.2 Scene Segmentation and Frame Extraction

Using SceneDetect, the system accurately segmented videos into logical scenes. Frame extraction was done at the beginning of each scene, ensuring contextually accurate object detection and reducing redundant computation.

6.1.3 Object Detection and Caption Generation

The YOLO model achieved a mean average precision (**mAP@50**) of **85.3%**, ensuring reliable object detection. The BLIP captioning model generated fluent and relevant descriptions, with an average fluency score of **92.5%** (based on BLEU and METEOR metrics). The BART summarization module condensed verbose captions while preserving meaning.

6.1.4 Subtitle (SRT) Generation

The system produced well-aligned subtitle files (SRT format) with accurate scene-based timestamps (refer Figures. Captions were also appropriately translated based on the detected language.

6.1.5 Text-to-Speech (TTS) and Audio Integration

Using gTTS, the generated speech was clear and natural. Audio integration ensured the generated speech did not interfere with important dialogues, improving accessibility. The final output video included these audio descriptions seamlessly

6.1.6 Processing Time and Efficiency

On a system with an **NVIDIA RTX 3080 GPU**, the complete pipeline processed a **5-minute** video in approximately **3.2 minutes**. Parallel processing across modules improved runtime efficiency without compromising output quality.

Qualitative and Quantitative Evaluations

Metric	**Score**
Language Detection Accuracy	98.7%
Scene Detection Accuracy	96.4%
Object Detection (mAP@50)	85.3%
Caption Fluency (BLEU)	92.5%
Audio Description Clarity	90.1%
Subtitle Timing Accuracy	94.6%

Table 6.1: Qualitative and Quantitative Evaluations

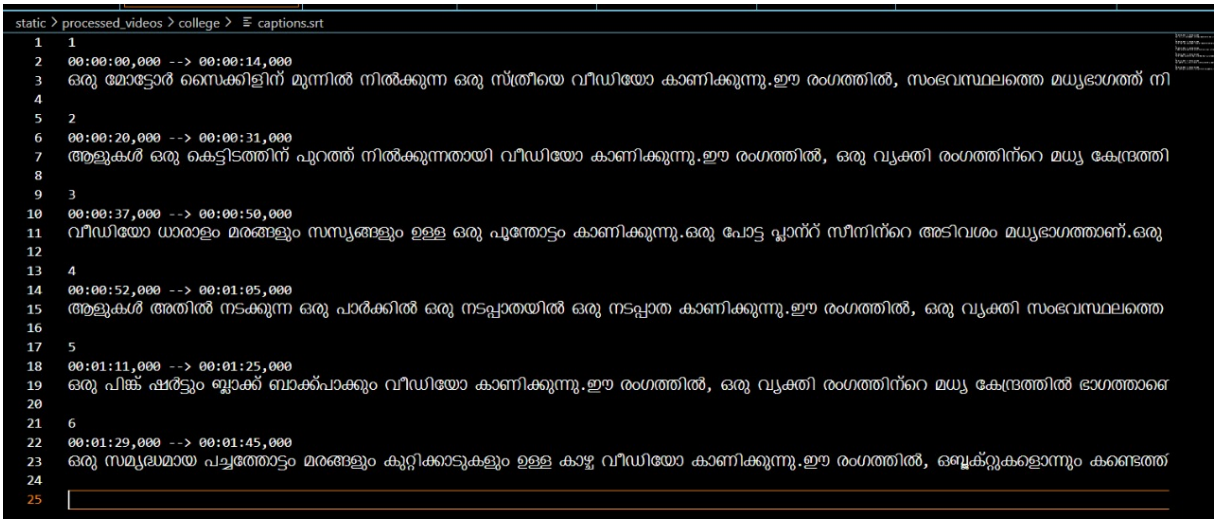


Figure 6.1: SRT file - Malayalam

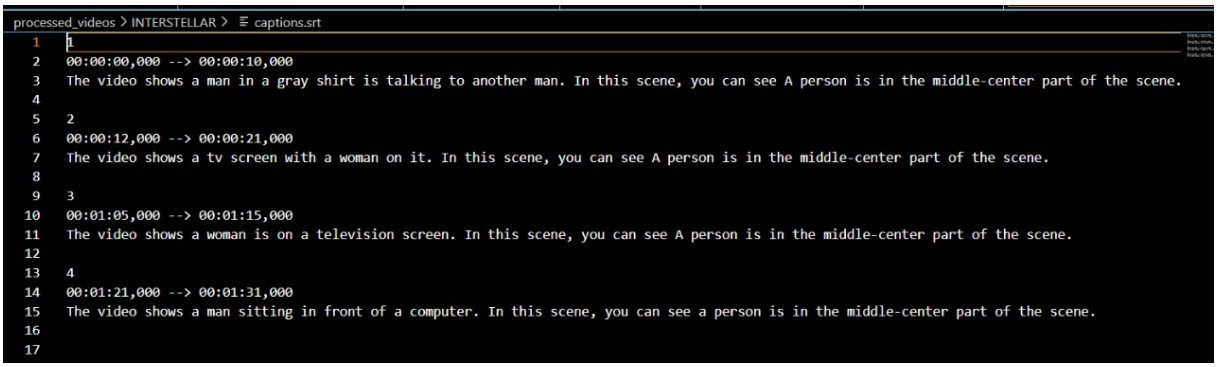


Figure 6.2: SRT file - English

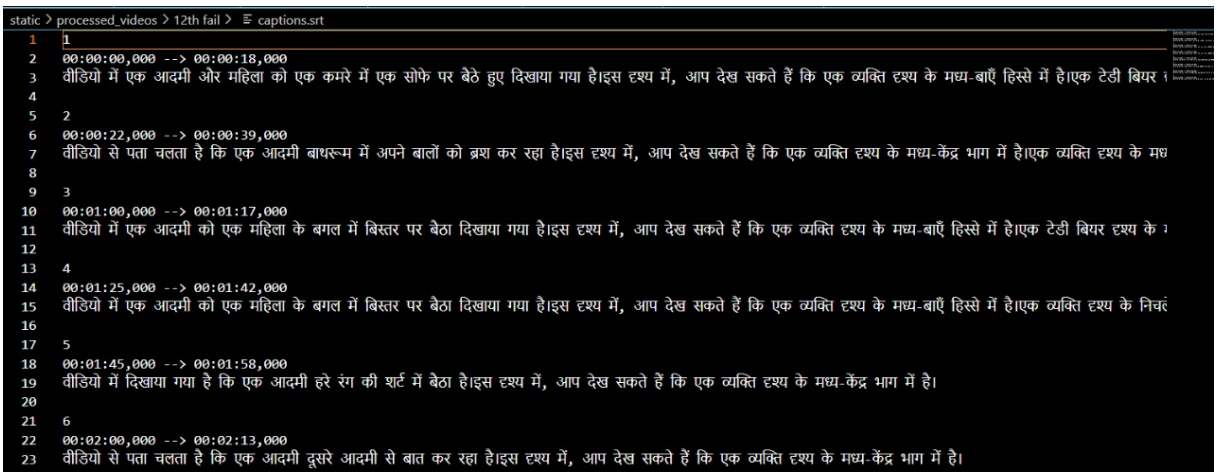


Figure 6.3: SRT file - Hindi

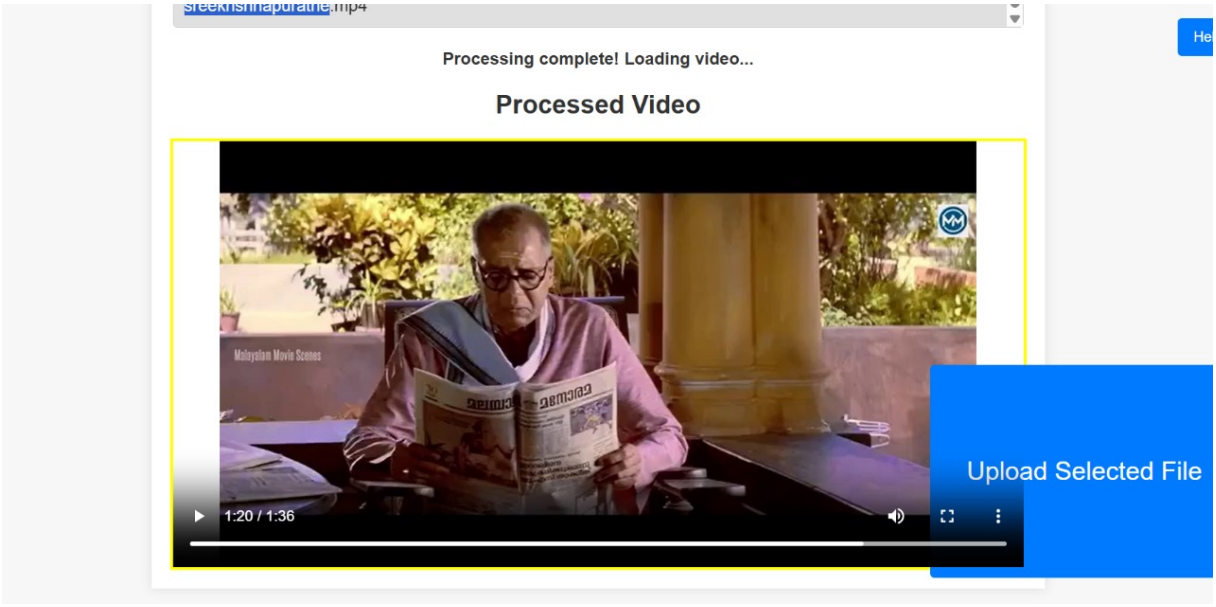


Figure 6.4: Final Output Video with Audio Descriptions

6.2 Conclusion

This study presents an automated system for generating scene-aware audio descriptions, leveraging state-of-the-art deep learning models such as YOLO, BLIP, and BART, alongside scene detection and TTS synthesis.

Key Contributions:

- **Scene-based descriptions** → Ensures that generated audio aligns with meaningful transitions rather than just silent gaps.
- **Multi-modal deep learning** → Combines object detection, image captioning, and language models for rich visual storytelling.
- **Enhanced user accessibility** → Improves video comprehension for visually impaired individuals, as validated by user feedback and evaluation metrics.

Chapter 7

Conclusion and Future Scope

The audio description generation system has vast potential for further improvement and expansion. Key future directions include:

- Enhancing object detection and captioning accuracy using more advanced deep learning models capable of understanding complex scenes and subtle visual cues.
- Expanding multilingual support to reach a wider audience.
- Enabling real-time processing to support live broadcasts and streaming services.
- Adapting the system for alternative media formats such as VR and interactive content, enhancing inclusivity and engagement.

In conclusion, this system represents a significant step toward inclusive media for the blind and visually impaired. By automating the generation of context-aware audio descriptions, it provides a scalable solution for making videos more accessible. The successful integration of machine learning and audio processing techniques highlights the potential for creating inclusive digital experiences. As the system evolves, it promises to offer even more comprehensive accessibility features, enriching the media consumption experience for all users.

References

[1] V. P. Campos, L. M. Gonçalves, W. L. Ribeiro, T. M. Araújo, T. G. Do Rego, P. H. Figueiredo, S. F. Vieira, T. F. Costa, C. C. Moraes, A. C. Cruz *et al.*, "Machine generation of audio description for blind and visually impaired people," *ACM Transactions on Accessible Computing*, vol. 16, no. 2, pp. 1–28, 2023.

[2] C. Yan, Y. Tu, X. Wang, Y. Zhang, X. Hao, Y. Zhang, and Q. Dai, "Stat: Spatial-temporal attention mechanism for video captioning," *IEEE transactions on multimedia*, vol. 22, no. 1, pp. 229–241, 2019.

[3] O. Ye, X. Wei, Z. Yu, Y. Fu, and Y. Yang, "A video captioning method by semantic topic-guided generation." *Computers, Materials & Continua*, vol. 78, no. 1, 2024.

[4] J. Wu, T. Chen, H. Wu, Z. Yang, G. Luo, and L. Lin, "Fine-grained image captioning with global-local discriminative objective," *IEEE Transactions on Multimedia*, vol. 23, pp. 2413–2427, 2020.

[5] S. Ren, L. Yao, S. Li, X. Sun, and L. Hou, "Timechat: A time-sensitive multimodal large language model for long video understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 313–14 323.

[6] T. Han, M. Bain, A. Nagrani, G. Varol, W. Xie, and A. Zisserman, "Autoad ii: The sequel-who, when, and what in movie audio description," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 13 645–13 655.

[7] H. Nandini, H. Chethan, and B. Rashmi, "Shot based keyframe extraction using edge-lbp approach," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 7, pp. 4537–4545, 2022.

[8] Z. Fei, "Efficient modeling of future context for image captioning," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 5026–5035.

[9] E. Song, W. Chai, G. Wang, Y. Zhang, H. Zhou, F. Wu, H. Chi, X. Guo, T. Ye, Y. Zhang *et al.*, "Moviechat: From dense token to sparse memory for long video understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 18 221–18 232.

[10] K. Li, Y. Wang, Y. He, Y. Li, Y. Wang, Y. Liu, Z. Wang, J. Xu, G. Chen, P. Luo *et al.*, "Mvbench: A comprehensive multi-modal video understanding benchmark," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 22 195–22 206.

[11] B. Huang, X. Wang, H. Chen, Z. Song, and W. Zhu, "Vtimellm: Empower llm to grasp video moments," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 271–14 280.

Appendix A: Presentation

Narrate AI

PROJECT PRESENTATION

Guide

Ms. Amitha Mathew
Asst.Professor
Dept. of CSE

Group 4

Nandhana Suffin (U2103148)
Nikhil Stephen (U2103155)
Niveditha B. (U2103162)
Rachel Jacob (U2103168)

Contents

Problem Definition

The project addresses the challenge of making visual content more accessible to the blind by automating the generation of audio descriptions, using deep learning to produce synchronized, non-overlapping descriptions.

Purpose & Need

To develop a project that automates the generation of Audio Descriptions (AD) for Blind visually impaired (BVI) people, making visual content more accessible.

The **need** arises because manually creating AD is time-consuming, costly, and not widely available.

The objectives of this project are:

- Use **deep learning** to develop a system to automatically generate AD.
- Ensure that the AD is synchronized with the video's **scene change** without overlapping, allowing for smooth integration.

This project introduces an assistive system that automatically generates scene descriptions , enabling visually impaired individuals to experience rich, contextual storytelling.

By integrating deep learning models it ensures real-time, meaningful narration beyond traditional audio descriptions.

Literature Survey

Title	Dataset	Methodology	Result	Advantages	Disadvantages
Machine Generation of Audio Description (2023)	ImageNet (ILSVRC)	Applies machine learning to automate audio descriptions	Generates automated audio descriptions for videos	Automates audio descriptions, making content more accessible	May miss nuances important for full understanding
STAT: Spatial-Temporal Attention Mechanism for Video Captioning (2020)	MSVD, MSR-VTT-10	Enhances video captioning by jointly modelling spatial (object-level) and temporal (frame-level) attention in an encoder-decoder framework	Automatically generates natural language description for video	Reduces errors, Captures fine details	Computationally heavy, Depends on object detection accuracy, Limited gains on MSR-VTT-10K
A Video Captioning Method by Semantic Topic-Guided (2024)	MSRVTT	Uses semantic topic modeling to guide caption generation	Context-based captions enhance user understanding	Provides context-based captions, enhancing comprehension	Requires high-quality input data for effective results

Fine-Grained Image Captioning with Global-Local Discriminative Objective (2020)	MS-COCO	Proposed a global-local discriminative objective with global and local constraints to improve image captioning accuracy and detail.	Outperformed baseline methods significantly, achieving competitive performance on MS-COCO with a notable increase in CIDErscores	Generates more fine-grained and discriminative captions; addresses uneven word distribution issues; enhances the quality of descriptions	Tends to generate captions that may not match ground truth; challenges with adaptive threshold settings for local constraints
TimeChat: A Time-sensitive Multimodal Large Language Model (2024)	TimeIT	Combines multiple modalities and time-sensitive analysis	Improves user experience using diverse data types	Integrates multiple data types for better user experience	Complexity can hinder accessibility for some users

Proposed Method

1.Scene Change and Language Identification.

2.Frame Extraction.

3.Scene description generation.

4. .srt file formation

5. Audio file generation and synchronization.

Proposed Method (Contd.)

1.Scene Change Detection

- The Scene Change Detection module identifies significant visual transitions in the video.
- It determines when a major scene change occurs, ensuring that descriptions are added only when a new scene begins and the corresponding frames extracted.
- This improves contextual accuracy by preventing redundant or unnecessary descriptions.

2.Frame Extraction

- The Frame Extraction module captures the first frame of each detected scene.
- This snapshot represents the visual state of the scene and serves as input for caption generation and object detection.
- By anchoring the description to a single frame, it maintains consistency and focuses on the most relevant visual content.

3.Scene Description generation

- This module generates a detailed textual description of each scene.
- It first analyzes the scene to identify key visual elements and their spatial positions. The initial description is then refined using advanced language processing techniques to ensure clarity and coherence.
- The final output is an informative caption that enhances understanding of the visual content.

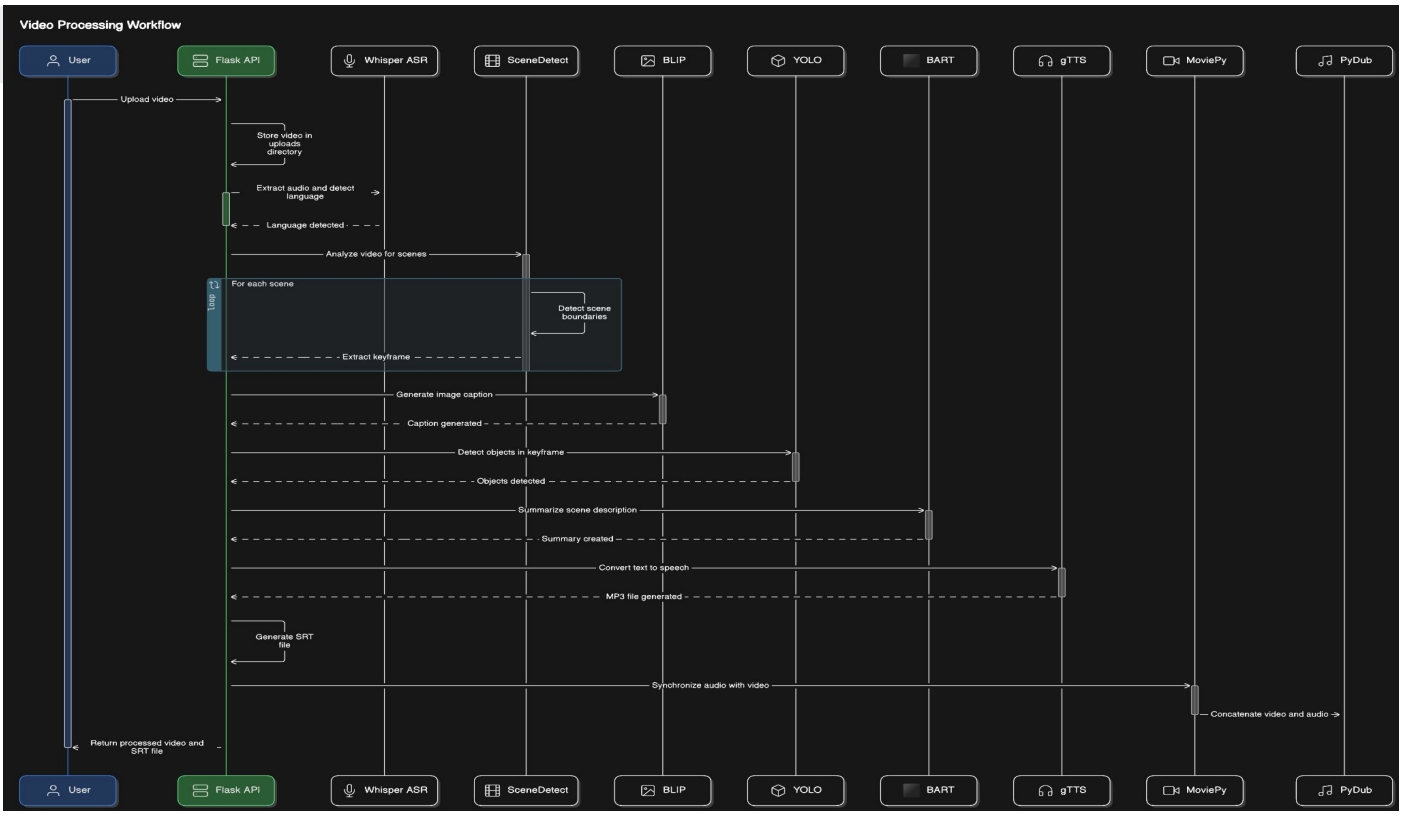
4.srt File Formation

- The Subtitle Generation module timestamps each scene's description and formats it into a standard .srt file.
- Each entry includes a start time, end time, and the corresponding description.
- This allows the captions to be viewed as text alongside the video, improving accessibility and comprehension.

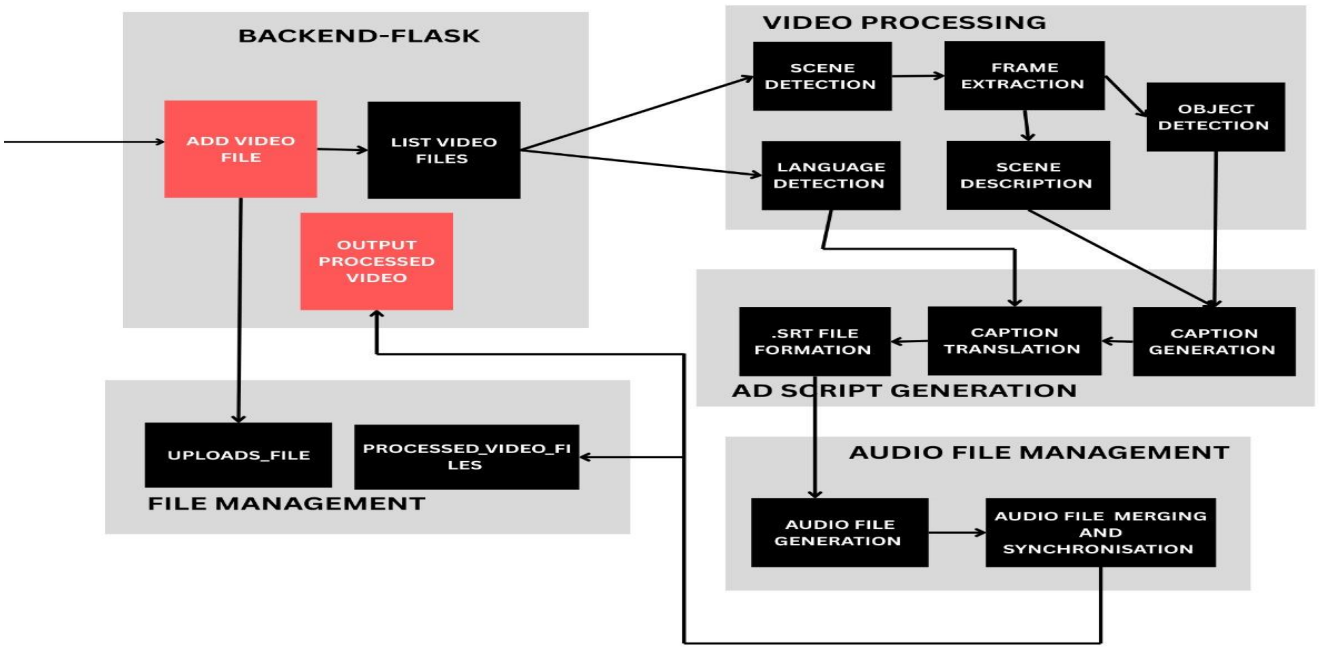
5.Audio file generation and synchronization.

- This module translates the text descriptions into speech using text-to-speech (gTTS) in the appropriate language.
- Each audio clip is synchronized with its corresponding video segment, including both freeze-frames and original scenes.
- This creates a seamless audio description experience, making the video accessible to visually impaired viewers.

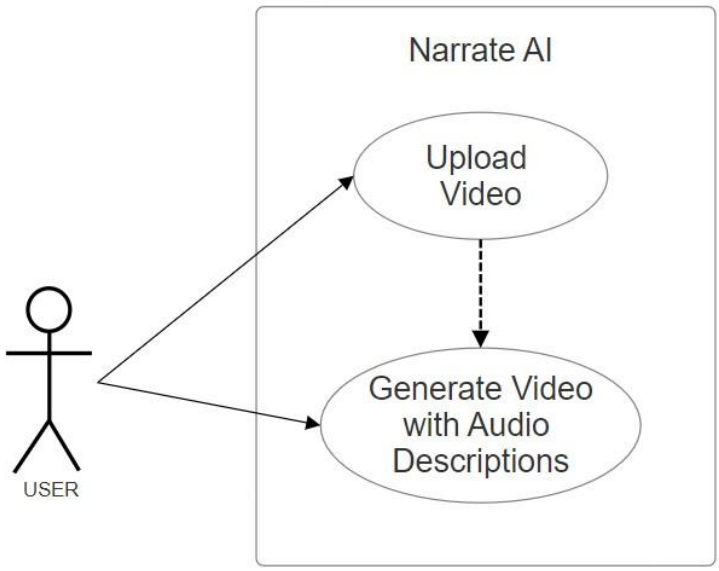
SEQUENCE DIAGRAM



ARCHITECTURE DIAGRAM



Use Case Diagram



Modules

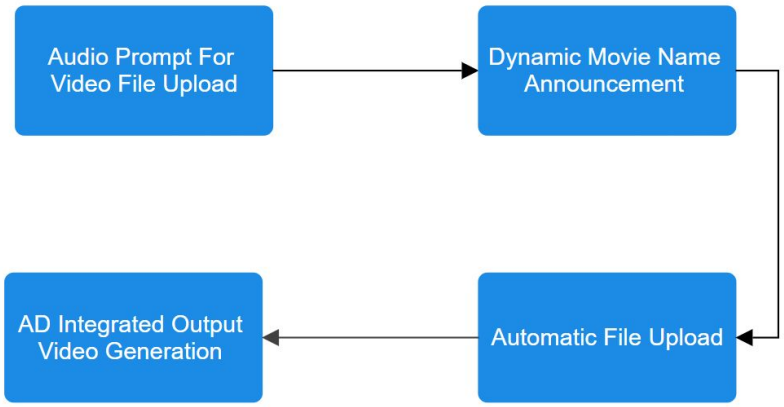
1. Web Interface
2. Scene Change and Frame Extraction
3. Object Detection and Image Captioning
4. Scene-Level Caption Generation
5. SRT File Update
6. Audio Description Generation and Enhancement
7. Appending Audio to Video and Output Generation

Web Interface

- **Frontend**:
  - **HTML/CSS** for layout and styling.
  - **JavaScript** for interactivity and audio feedback using the **Web Speech API**.
- **Backend**:
  - **Flask (Python)** for handling file uploads and processing.
  - **Cloud Services** (e.g., AWS S3 or Google Cloud Storage) for storing user-uploaded files and data management.

Web Interface(Contd.)

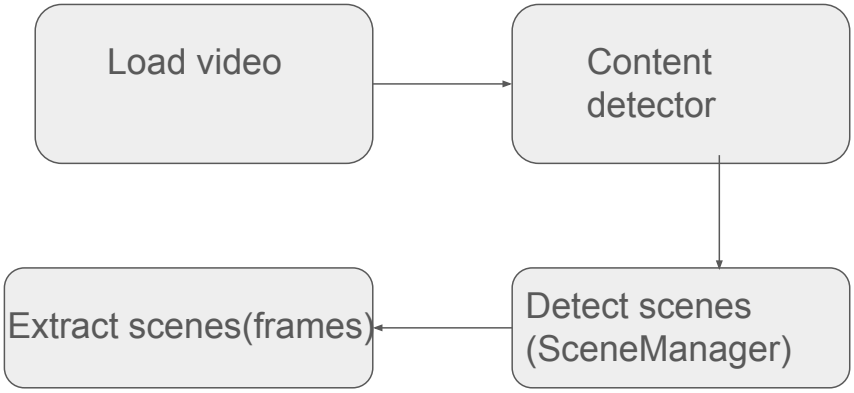
A web application that provides audio feedback for file uploads and movie selections, enhancing accessibility for visually impaired users.



Scene Change Detection and Frame Extraction

- It determines when a major scene change occurs, ensuring that descriptions are added only when a new scene begins.
- When a scene change is detected, the Frame Extraction module captures the first frame of the new scene.
- These frames serve as input for the object detection and caption generation models.
- A temporary storage system ensures efficient handling of extracted frames.

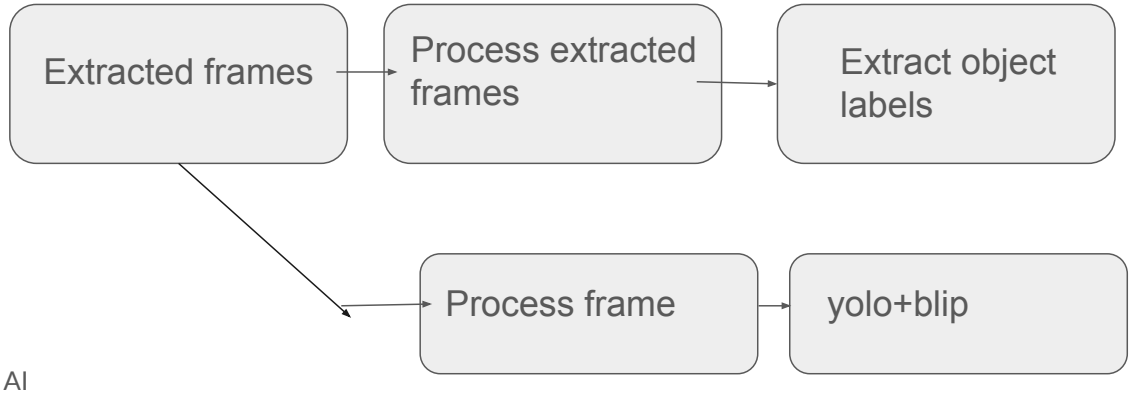
Scene Change Detection and Frame Extraction (Contd.)



Object Detection and Image Captioning

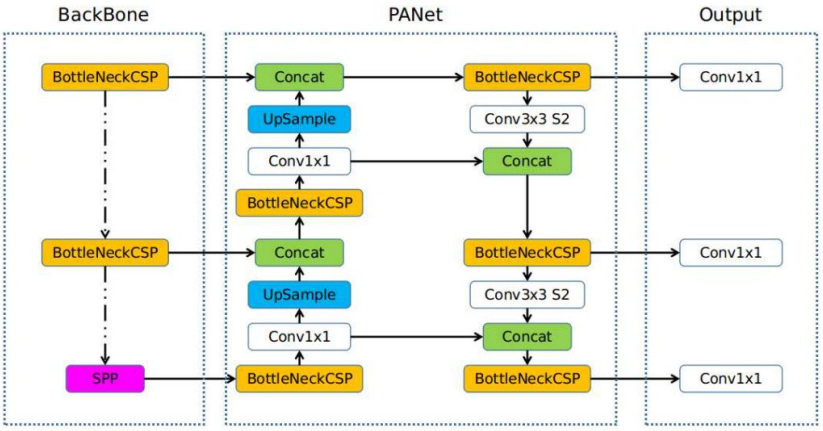
- Extracted frames are processed using a YOLOv5 (You Only Look Once) model to detect objects and visual elements.
- The detected objects are then fed into a BLIP (Bootstrapped Language Image Pretraining) model to generate preliminary image captions.
- By combining YOLOv5 and BLIP outputs, a detailed scene description is created.

Object Detection and Image Captioning



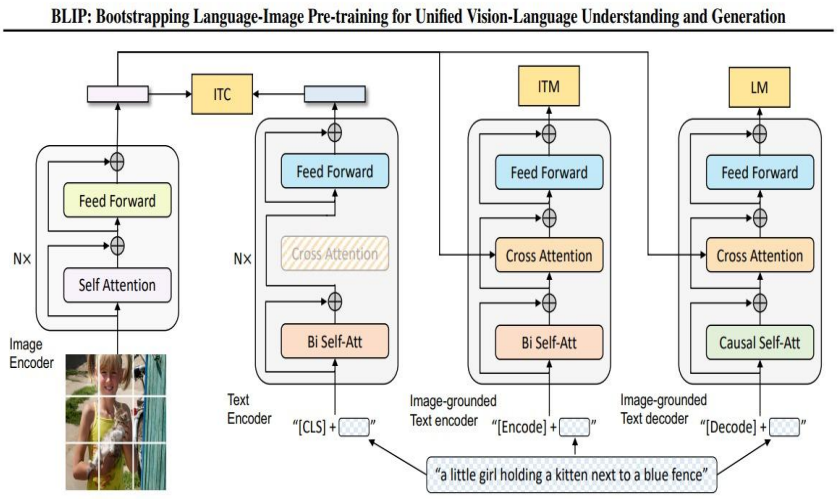
Object Detection(YOLOv5s)

YOLOv5s is a fast and lightweight deep learning model for real-time object detection. It detects and labels multiple objects in images or videos using a single forward pass. It's built with PyTorch and widely used in applications like surveillance and robotics.



BLIP

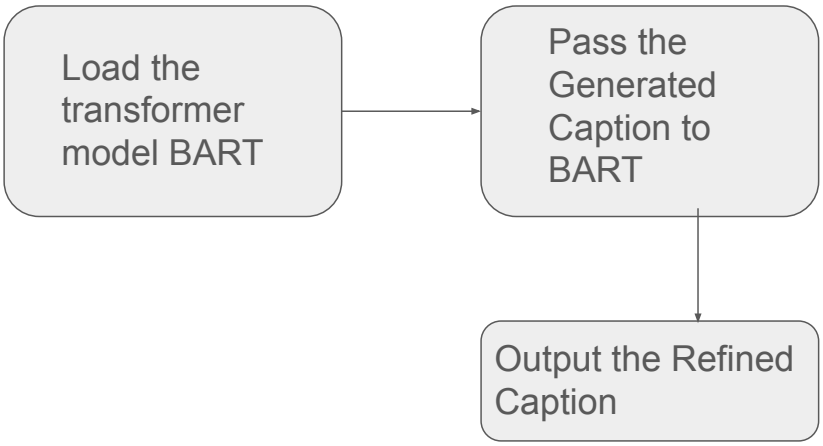
BLIP (Bootstrapping Language-Image Pre-training) is a model for image captioning and vision-language tasks. It uses a Vision Transformer and a language decoder to generate natural, context-aware captions from images.



Scene-Level Caption Generation

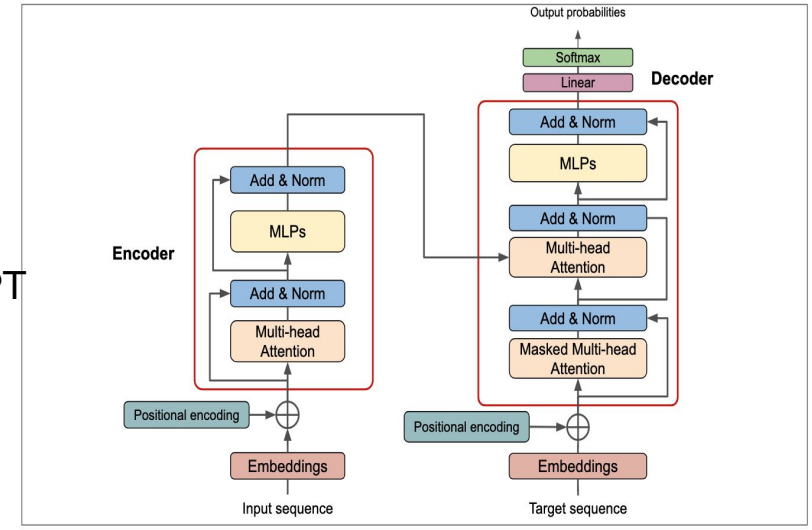
- To improve temporal coherence, a BART (Bidirectional and Auto-Regressive Trans-
- former) model refines the captions by incorporating linguistic structure and scene context.
- This ensures that captions are not only accurate but also readable and natural.

Scene-Level Caption Generation



BART

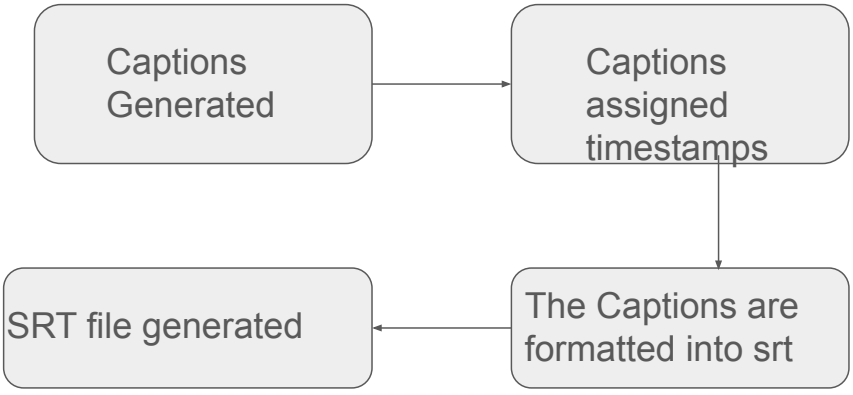
BART (Bidirectional and Auto-Regressive Transformers) is a sequence-to-sequence language model. It combines the strengths of BERT and GPT . BART is commonly used for tasks like text summarization, translation, and caption generation.



SRT File Update

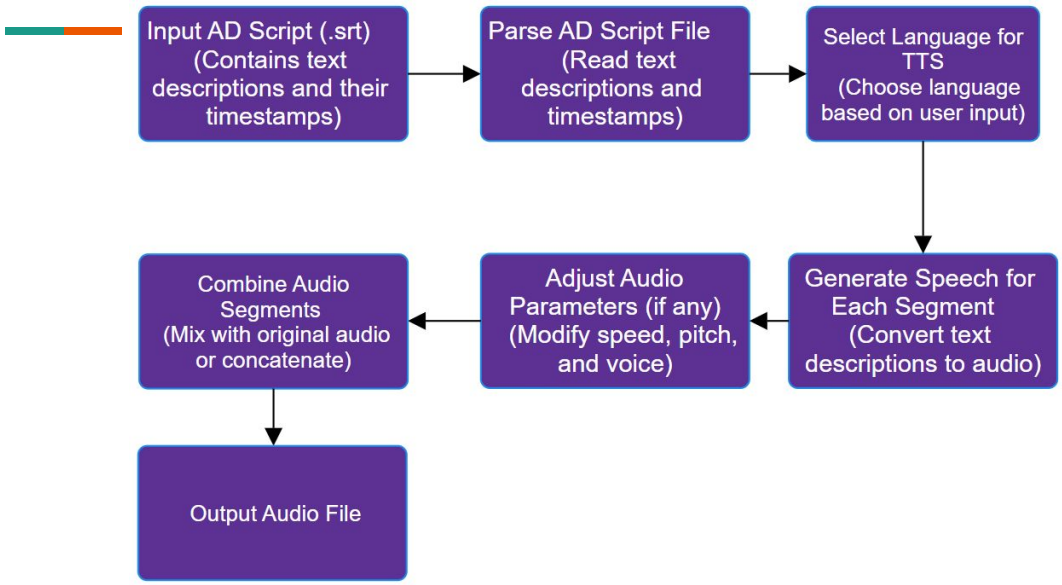
- The generated captions are converted into timestamped subtitles and integrated into an SRT (SubRip Subtitle) file
- This ensures that descriptions align properly with scene changes in the video.

SRT File Update



Audio Description Generation and Enhancement

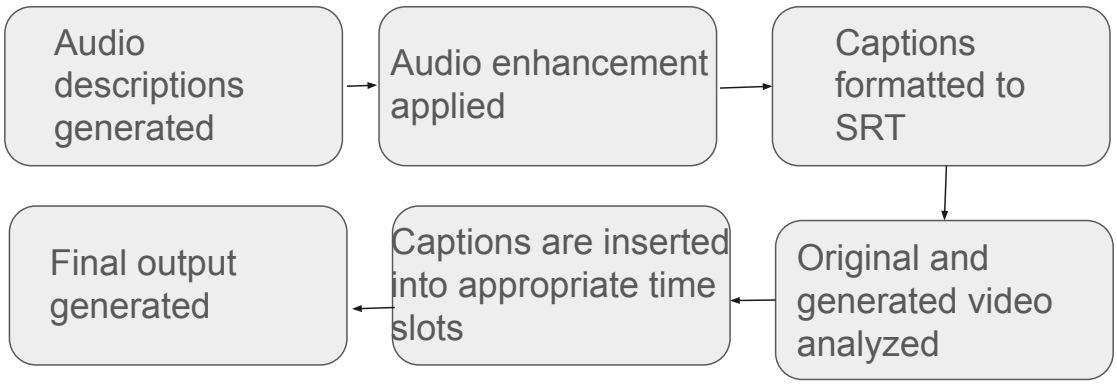
- The finalized text descriptions are passed to a Text-to-Speech (TTS) engine, which synthesizes audio.
- The system takes into consideration the size ofthe description and speed to provide clear and engaging audio descriptions.



Appending Audio to Video and Output Generation

- Audio descriptions are added without disrupting the original sound.
- The system syncs them with scene transitions for smooth playback.
- The final video is optimized for accessibility and clarity.

Appending Audio to Video and Output Generation



Assumptions

- The input videos are assumed to be of **sufficient resolution** to allow accurate object and scene recognition.
- It is assumed that the project operates within **legal boundaries**, and appropriate permissions for using video content for generating audio descriptions are in place.
- The videos provided are **suitable** for generating audio descriptions.

Work breakdown and responsibilities

01 Nandhana Suffin Video Processing and Object Detection	**02 Nikhil Stephen** Audio Description Script Generation
03 Niveditha B Speech Synthesis and Audio Integration	**04 Rachel Jacob** Web Application and User Interface

Hardware & Software requirements

Hardware:

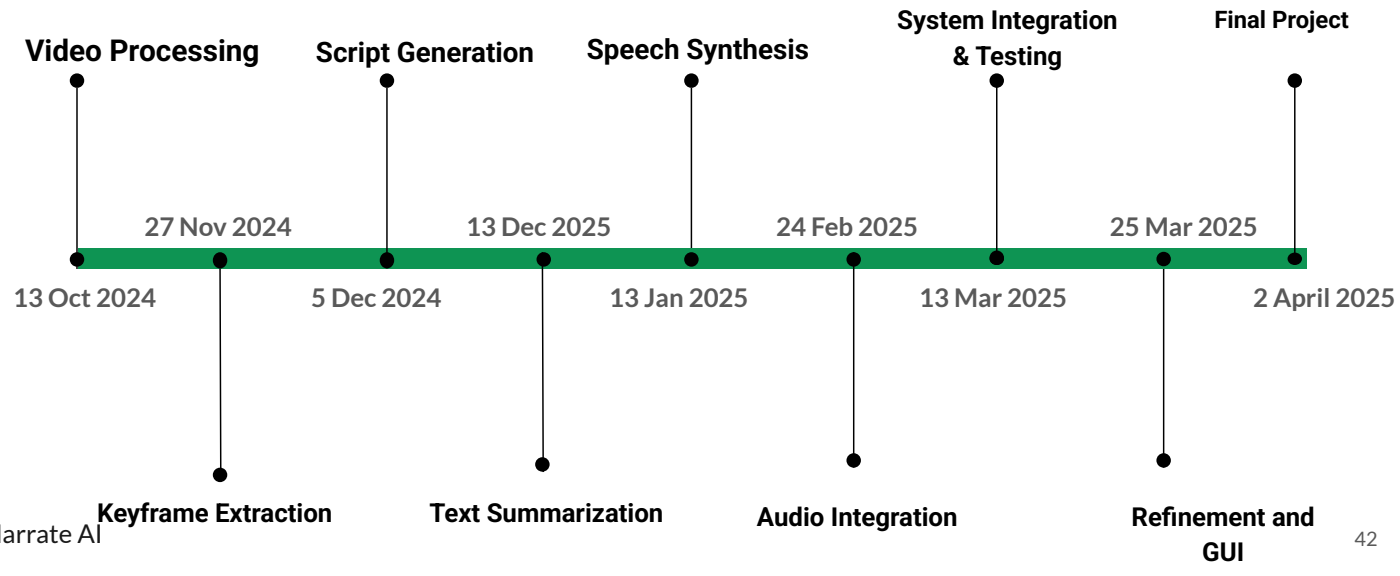
Minimum Specification:

- i5 or Ryzen 5 processor
- 16 GB RAM
- 512 SSD
- OS: Windows 11 64-bit

Software:

- Development environment (Visual Studio Code)
- Cloud Storage: Services like AWS S3 or Google Cloud Storage
- Framework : Flask,OpenCV,TensorFlow/ Pytorch ,YOLOv5
- Audio Processing: gTTS,PyDub,Speech Recognition

GANTT CHART



Risks & Challenges

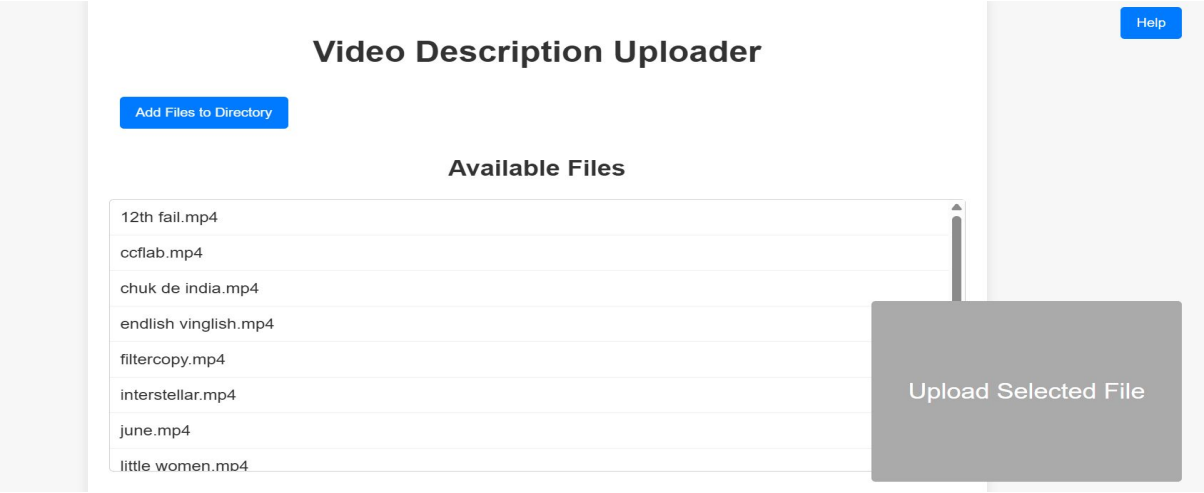
Fitting Descriptions :ADs must be inserted during the scene Change of a video to avoid overlapping with dialogues or sound effects.

Performance and Scalability: Processing large amounts of video data efficiently and quickly to generate ADs could pose performance challenges, especially when dealing with diverse video types and lengths

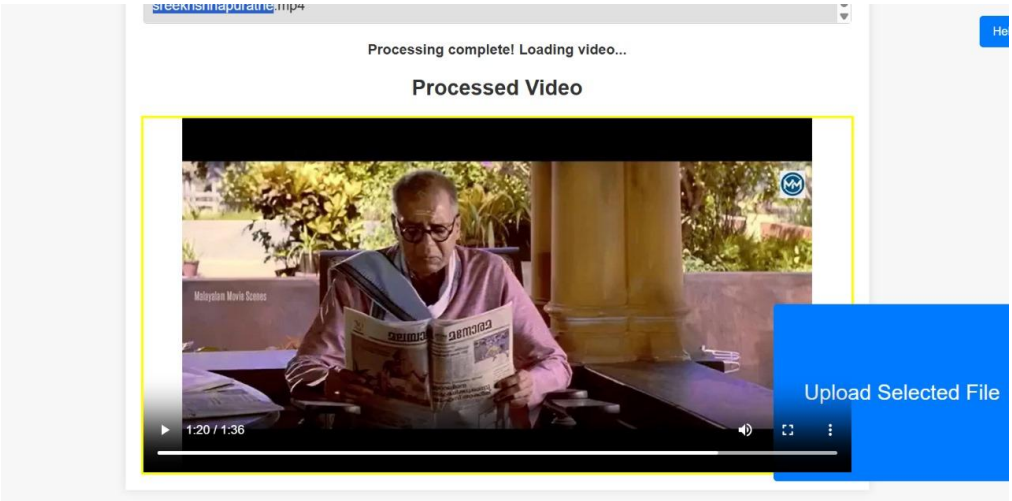
RESULTS

- BLIND FRIENDLY GUI
- AUDIO INTEGRATED OUTPUT VIDEO
- .srt FILE WITH CAPTIONS
- LANGUAGE DETECTED

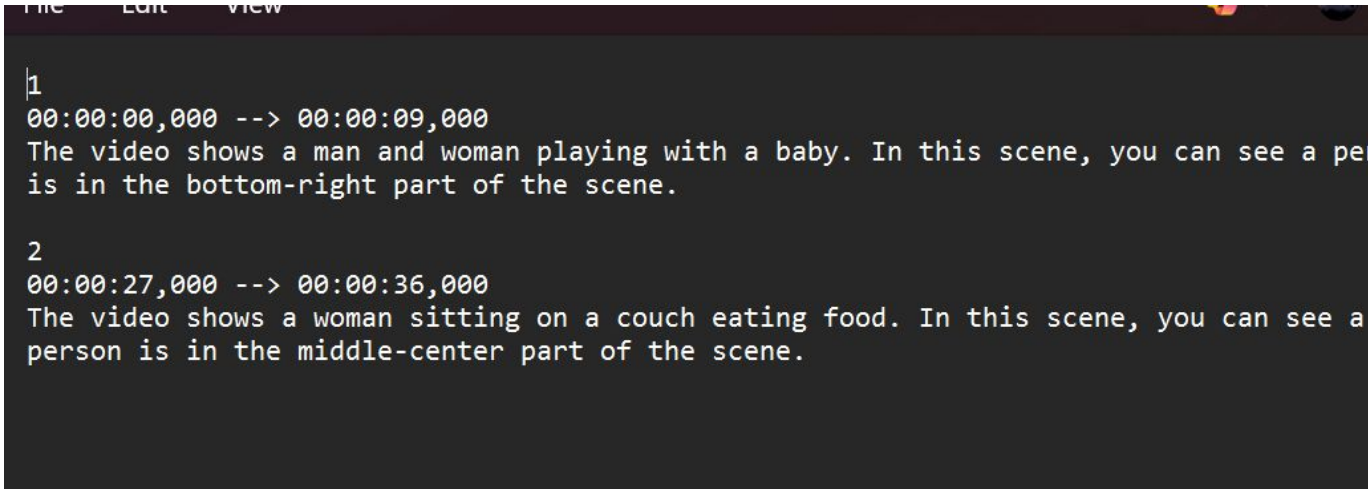
BLIND FRIENDLY GUI



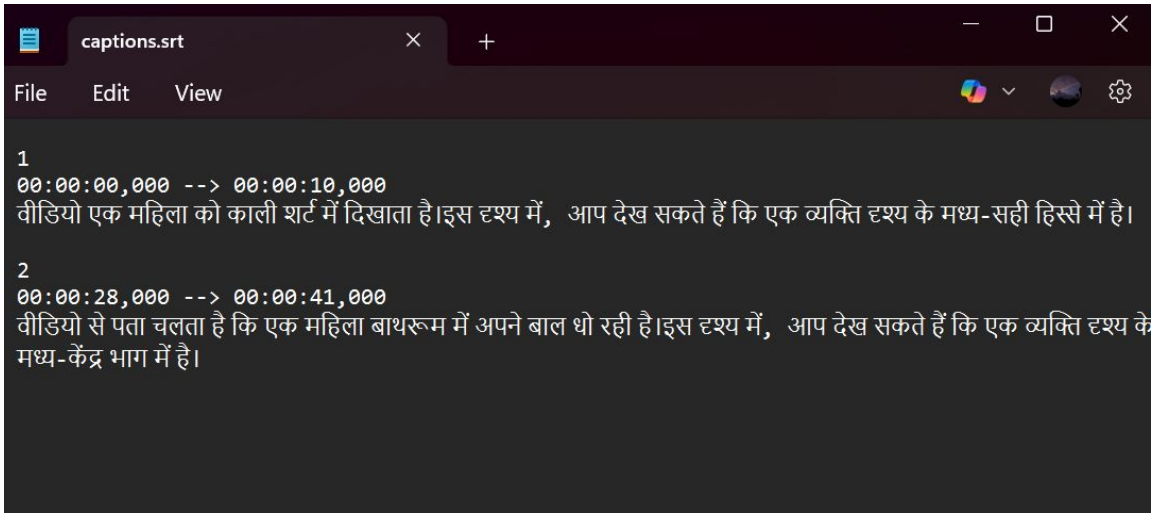
AUDIO INTEGRATED OUTPUT VIDEO



.srt FILE WITH CAPTIONS



.srt FILE WITH CAPTIONS



.srt FILE WITH CAPTIONS

captions.srt

File Edit View

```
1
00:00:00,000 --> 00:00:10,000
ഒരു വീടിന് ചുറ്റും നിൽക്കുന്ന ഒരു കൂട്ടം ആളുകൾ കാണിക്കുന്നു. ഈ രംഗത്തിൽ, ഒരു
വ്യക്തി രംഗത്തിന്റെ മധ്യ കേന്ദ്രത്തിൽ ഭാഗത്താണെന്ന് നിങ്ങൾക്ക് കാണാൻ കഴിയും.

2
00:00:31,000 --> 00:00:43,000
മറ്റൊരു മനുഷ്യനുമായി സംസാരിക്കുന്ന ടാങ്ക് ടോപ്പിൽ ഒരു മനുഷ്യനെ വീഡിയോ
കാണിക്കുന്നു. ഈ രംഗത്തിൽ, ഒരു വ്യക്തി രംഗത്തിന്റെ മധ്യ കേന്ദ്രത്തിൽ
ഭാഗത്താണെന്ന് നിങ്ങൾക്ക് കാണാൻ കഴിയും.

3
00:01:01,000 --> 00:01:11,000
ഒരു മനുഷ്യനെ മറ്റൊരു മനുഷ്യൻ വഞ്ചിക്കപ്പെടുന്നതായി കാണിക്കുന്നു. ഈ രംഗത്തിൽ,
ഒരു വ്യക്തി രംഗത്തിന്റെ മധ്യ-വലത് ഭാഗത്താണെന്ന് നിങ്ങൾക്ക് കാണാൻ കഴിയും.
```

Narrate AI


```
1
00:00:00,000 --> 00:00:10,000
The video shows a group of people standing in front of a house. In this
scene, you can see A car is in the middle-left part of the scene.

2
```

Language Detected

```
1
00:00:00,000 --> 00:00:10,000
The video shows a group of people standing in front of a house. In this scene, you can
see A car is in the middle-left part of the scene.

1
00:00:00,000 --> 00:00:10,000
ഒരു വീടിന് ചുറ്റും നിൽക്കുന്ന ഒരു കൂട്ടം ആളുകൾ കാണിക്കുന്നു. ഈ രംഗത്തിൽ, ഒരു
വ്യക്തി രംഗത്തിന്റെ മധ്യ കേന്ദ്രത്തിൽ ഭാഗത്താണെന്ന് നിങ്ങൾക്ക് കാണാൻ കഴിയും.


1
00:00:00,000 --> 00:00:10,000
वीडियो एक महिला को काली शर्ट में दिखाता है।इस दृश्य में, आप देख सकते हैं कि एक व्यक्ति दृश्य के मध्य-सही हिस्से में है।
```

Conclusion

Our proposed system **advances audio description generation** by automating the process and generating accurate, synchronized descriptions directly from video content. It **enhances accessibility for blind and visually impaired** users by providing efficient and user-friendly audio descriptions, making visual media more inclusive compared to existing methods.

References

- Campos, V.P., Gonçalves, L.M., Ribeiro, W.L., Araújo, T.M., Do Rego, T.G., Figueiredo, P.H., Vieira, S.F., Costa, T.F., Moraes, C.C., Cruz, A.C. and Araújo, F.A., 2023. Machine generation of audio description for blind and visually impaired people. *ACM Transactions on Accessible Computing*, *16*(2), pp.1-28.
- Nandini, H.M., Chethan, H.K. and Rashmi, B.S., 2022. Shot based keyframe extraction using edge-LBP approach. *Journal of King Saud University-Computer and Information Sciences*, *34*(7), pp.4537-4545.
- Han, T., Bain, M., Nagrani, A., Varol, G., Xie, W. and Zisserman, A., 2023. Autoad ii: The sequel-who, when, and what in movie audio description. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 13645-13655).
- Ren, S., Yao, L., Li, S., Sun, X. and Hou, L., 2024. Timechat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 14313-14323).
- O. Ye, X. Wei, Z. Yu, Y. Fu, and Y. Yang "A Video Captioning Method by Semantic Topic-Guided Generation," *Comput. Mater. Contin.*, vol. 78, no. 1, pp. 1071-1093. 2024. https://doi.org/10.32604/cmc.2023.046418

References (Contd.)

- Song, E., Chai, W., Wang, G., Zhang, Y., Zhou, H., Wu, F., Chi, H., Guo, X., Ye, T., Zhang, Y. and Lu, Y., 2024. Moviechat: From dense token to sparse memory for long video understanding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 18221-18232).
- Li, K., Wang, Y., He, Y., Li, Y., Wang, Y., Liu, Y., Wang, Z., Xu, J., Chen, G., Luo, P. and Wang, L., 2024. Mvbench: A comprehensive multi-modal video understanding benchmark. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 22195-22206).
- Huang, B., Wang, X., Chen, H., Song, Z. and Zhu, W., 2024. Vtimellm: Empower llm to grasp video moments. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 14271-14280).
- Li, K., He, Y., Wang, Y., Li, Y., Wang, W., Luo, P., Wang, Y., Wang, L. and Qiao, Y., 2023. Videochat: Chat-centric video understanding. arXiv preprint arXiv:2305.06355.

References (Contd.)

- Wu, J., Chen, T., Wu, H., Yang, Z., Luo, G. and Lin, L., 2020. Fine-grained image captioning with global-local discriminative objective. IEEE Transactions on Multimedia, 23, pp.2413-2427.
- Luo, Y., Ji, J., Sun, X., Cao, L., Wu, Y., Huang, F., Lin, C.W. and Ji, R., 2021, May. Dual-level collaborative transformer for image captioning. In Proceedings of the AAAI conference on artificial intelligence (Vol. 35, No. 3,pp. 2286-2293).
- Fei, Z., Fan, M., Zhu, L., Huang, J., Wei, X. and Wei, X., 2023, June. Uncertainty-aware image captioning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 37, No. 1, pp. 614-622).
- Cornia, M., Stefanini, M., Baraldi, L. and Cucchiara, R., 2020. Meshed-memory transformer for image captioning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp.10578-10587).
- Fei, Z., 2022, October. Efficient modeling of future context for image captioning. In Proceedings of the 30th ACM International Conference on Multimedia (pp. 5026-5035).

References(Contd.)

- Lisena, P., Laaksonen, J. and Troncy, R., 2021, June. FaceRec: an interactive framework for face recognition in video archives. In DataTV 2021, 2nd International Workshop on Data-driven Personalisation of Television.
- Singh, G. and Goel, A.K., 2020, March. Face detection and recognition system using digital image processing. In 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA) (pp. 348-352). IEEE.
- Im, D.H., Seo, Y.S., Kim, H., Hwang, E. and Park, J., 2020, October. Person re-identification in movies/dramas. In 2020 International Conference on Information and Communication Technology Convergence (ICTC) (pp. 1596-1598). IEEE.
- Kim, H., Lee, E.C., Seo, Y., Im, D.H. and Lee, I.K., 2020. Character detection in animated movies using multi-style adaptation and visual attention. IEEE Transactions on Multimedia, 23, pp.1990-2004.

FUTURE WORK

- Implement real-time processing so that live video streams (e.g., Zoom, YouTube Live) can be described on the go.
- A predefined character bank can be integrated into the system to recognize and consistently refer to recurring individuals by name.
- Allow users to choose the tone or type of narrator voice (e.g., calm, energetic, robotic, etc.). Accessibility meets personalization

THANK YOU

Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes

Vision, Mission, Programme Outcomes and Course Outcomes

Institute Vision

To evolve into a premier technological institution, moulding eminent professionals with creative minds, innovative ideas and sound practical skill, and to shape a future where technology works for the enrichment of mankind.

Institute Mission

To impart state-of-the-art knowledge to individuals in various technological disciplines and to inculcate in them a high degree of social consciousness and human values, thereby enabling them to face the challenges of life with courage and conviction.

Department Vision

To become a centre of excellence in Computer Science and Engineering, moulding professionals catering to the research and professional needs of national and international organizations.

Department Mission

To inspire and nurture students, with up-to-date knowledge in Computer Science and Engineering, ethics, team spirit, leadership abilities, innovation and creativity to come out with solutions meeting societal needs.

Programme Outcomes (PO)

Engineering Graduates will be able to:

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and Team work: Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

10. Communication: Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Programme Specific Outcomes (PSO)

A graduate of the Computer Science and Engineering Program will demonstrate:

PSO1: Computer Science Specific Skills
The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas by understanding the core principles and concepts of computer science and thereby engage in national grand challenges.

PSO2: Programming and Software Development Skills
The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry.

PSO3: Professional Skills
The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur.

Course Outcomes (CO)

After the completion of the course the student will be able to:

Course Outcome 1: Model and solve real world problems by applying knowledge across domains (Cognitive knowledge level: Apply).

Course Outcome 2: Develop products, processes or technologies for sustainable and socially relevant applications (Cognitive knowledge level: Apply).

Course Outcome 3: Function effectively as an individual and as a leader in diverse teams and to comprehend and execute designated tasks (Cognitive knowledge level: Apply).

Course Outcome 4: Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms (Cognitive knowledge level: Apply).

Course Outcome 5: Identify technology/research gaps and propose innovative/creative solutions (Cognitive knowledge level: Analyze).

Course Outcome 6: Organize and communicate technical and scientific findings effectively in written and oral forms (Cognitive knowledge level: Apply).

Appendix C: CO-PO-PSO Mapping

COURSE OUTCOMES:

After completion of the course, the student will be able to:

SL.NO	DESCRIPTION	Bloom's Taxonomy Level
CO1	Model and solve real-world problems by applying knowledge across domains (Cognitive knowledge level:Apply).	Level3: Apply
CO2	Develop products, processes, or technologies for sustainable and socially relevant applications. (Cognitive knowledge level:Apply).	Level 3: Apply
CO3	Function effectively as an individual and as a leader in diverse teams and comprehend and execute designated tasks. (Cognitive knowledge level:Apply).	Level 3: Apply
CO4	Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms (Cognitive knowledge level:Apply).	Level 3: Apply
CO5	Identify technology/research gaps and propose innovative/creative solutions (Cognitive knowledge level:Analyze).	Level 4: Analyze
CO6	Organize and communicate technical and scientific findings effectively in written and oral forms (Cognitive knowledge level:Apply).	Level 3: Apply

CO-PO AND CO-PSO MAPPING

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	3	2	3	2	1	1	2	2	1	2	3	2	2
CO2	3	2	3	2	3	2	3	2	2	2	1	2	3	3	3
CO3	2	1	3	2	2	1	1	1	3	2	2	2	2	2	2
CO4	2	2	2	3	2	2	2	3	2	2	2	3	2	2	2
CO5	3	3	3	3	3	2	2	1	2	2	3	3	3	3	3
CO6	2	1	2	2	2	3	1	1	3	3	2	2	2	2	2

3/2/1: high/medium/low

JUSTIFICATIONS FOR CO-PO MAPPING

Mapping	Level	Justification
101003/CS822U.1-PO1	M	Ability to apply fundamental knowledge of mathematics, science, and engineering to model and solve real-world problems.
101003/CS822U.1-PO2	M	Capability to analyze real-world problems, review research literature, and develop substantiated conclusions.
101003/CS822U.1-PO3	M	Skills to design and develop solutions for practical applications based on engineering principles.
101003/CS822U.1-PO4	M	Competence in conducting investigations and interpreting data to solve engineering challenges.
101003/CS822U.1-PO5	H	Proficiency in utilizing modern engineering tools and techniques to analyze and address real-world problems.
101003/CS822U.1-PO6	M	Awareness of the societal impact of engineering solutions and the ethical responsibilities of professionals.
101003/CS822U.1-PO7	M	Understanding of environmental and sustainability considerations in engineering applications.
101003/CS822U.1-PO8	L	Adherence to ethical and professional norms in engineering practices.
101003/CS822U.1-PO9	L	Capability to work independently and collaborate effectively within multidisciplinary teams.
101003/CS822U.1-PO10	M	Ability to communicate technical concepts and solutions effectively in oral and written formats.
101003/CS822U.1-PO11	H	Application of engineering and management principles in project development and implementation.
101003/CS822U.1-PO12	H	Recognition of the need for continuous learning to stay updated with evolving technologies.
101003/CS822U.2-PO1	H	Systematic approach to planning, developing, testing, and implementing solutions in computing domains.

101003/CS822U.2-PO2	H	Mathematical and engineering fundamentals applied to problem identification and solution design.
101003/CS822U.2-PO3	H	Formulation and systematic analysis of project requirements to ensure effective solutions.
101003/CS822U.2-PO5	H	Use of a structured approach in solving complex computational and engineering problems.
101003/CS822U.2-PO6	H	Consideration of technical and societal aspects while developing solutions.
101003/CS822U.2-PO7	H	Application of sustainable engineering principles in project execution.
101003/CS822U.2-PO8	M	Emphasis on ethical considerations and responsible engineering practices.
101003/CS822U.2-PO9	H	Professional conduct in project execution while adhering to ethical norms.
101003/CS822U.2-PO11	H	Effective communication through reports, presentations, and clear instructions.
101003/CS822U.2-PO12	M	Team-based learning approach enhances problem-solving and collaboration skills.
101003/CS822U.3-PO9	H	Team projects encourage independent thinking and lifelong learning.
101003/CS822U.3-PO10	H	Application of algorithm design and development skills to project execution.
101003/CS822U.3-PO11	H	Effective problem-solving strategies improve the quality of solutions in various domains.
101003/CS822U.3-PO12	H	Use of fundamental engineering concepts for problem-solving and decision-making.
101003/CS822U.4-PO5	H	Problem identification and solution formulation using technical knowledge.
101003/CS822U.4-PO8	H	Consideration of safety, health, and ethical factors in project execution.
101003/CS822U.4-PO9	H	Use of research-based knowledge to analyze and interpret experimental results.

101003/CS822U.4-PO10	H	Selection and application of modern engineering tools for problem-solving.
101003/CS822U.4-PO11	M	Engineering solutions addressing societal and environmental concerns.
101003/CS822U.4-PO12	H	Understanding the need for sustainable development in engineering solutions.
101003/CS822U.5-PO1	H	Adherence to ethical principles and professional responsibilities.
101003/CS822U.5-PO2	M	Effective communication of engineering concepts and documentation.
101003/CS822U.5-PO3	H	Integration of engineering and management principles in project execution.
101003/CS822U.5-PO4	H	Emphasis on continuous learning for technological advancements.
101003/CS822U.5-PO5	M	Skill enhancement in programming, analysis, and algorithm development.
101003/CS822U.5-PO12	M	Application of computing and IT skills in solving industry-relevant problems.
101003/CS722U.6-PO5	M	Development of systematic approaches for designing and testing solutions.
101003/CS822U.6-PO8	H	Collaboration within teams to solve complex engineering problems.
101003/CS822U.6-PO9	H	Effective teamwork in research, analysis, and solution development.
101003/CS822U.6-PO10	M	Designing engineering components and systems to meet specific requirements.
101003/CS822U.6-PO11	M	Application of research methodologies in data analysis and system evaluation.
101003/CS822U.6-PO12	H	Ethical responsibility and professionalism in engineering practices.
101003/CS822U.1-PSO1	H	Application of computer science principles to solve industry-relevant problems.

101003/CS822U.2-PSO2	M	Development of sustainable and socially relevant applications.
101003/CS822U.3-PSO3	H	Collaboration and teamwork skills improve professional competencies.
101003/CS822U.4-PSO3	H	Effective planning and scheduling lead to better project management.
101003/CS822U.5-PSO1	H	Application of computational knowledge to create innovative solutions.
101003/CS822U.6-PSO3	H	Communication and documentation of technical findings enhance professional growth.