

AI-Driven Accessibility: Automating Audio Descriptions For Video Content

Manish Patil¹, Shraddha Sanjay Mhetre²

¹MTS Software Engineer Salesforce, Inc.

²Android Engineer

Abstract

Background: Traditional audio description generation for videos often relies solely on visual frame analysis, overlooking the potential of integrating advanced natural language generation and text-to-speech technologies into a unified automated pipeline. Existing solutions are either manual or semi-automated, making them time-consuming, inconsistent, and less scalable for real-world accessibility needs, especially for visually impaired audiences.

Objective: This study aims to develop a robust audio-visual captioning framework that leverages extracted video frames and corresponding audio transcripts to produce semantically coherent and contextually relevant descriptions. By evaluating the proposed method against established video captioning approaches, the study seeks to demonstrate its effectiveness in improving descriptive accuracy and contextual depth.

Methods: The system extracts frames from user-uploaded videos at one frame per second using OpenCV. Optionally, YOLOv8 is applied for object detection to improve contextual accuracy. Captions are generated for each frame using the BLIP-2 model, and Google Text-to-Speech converts these captions into high-quality narration. The processed frames and audio are then reconstructed into a final video. Evaluation is conducted using BLEU and ROUGE metrics, along with qualitative user feedback for clarity and usefulness.

Results: Experimental evaluation shows the proposed method achieves a BLEU score of 0.82 and ROUGE score of 0.84, outperforming multiple prior studies in both metrics. Generated captions demonstrate improved contextual relevance and richer descriptions, effectively integrating scene details from visual frames and semantic cues from audio transcripts for enhanced multimedia content understanding.

Conclusion: The proposed approach effectively fuses audio and visual features, generating highly accurate captions, and offers strong potential for multimedia content applications.

Keywords: AI, videos, visual frame, multimedia content, transcripts.

1 Introduction

Video accessibility is a rather important issue to access digital content equally accessible to everyone, including disabled people. With video becoming the mainstream delivery format of information, education, entertainment, and general communication requirements there has been a greater necessity to ensure that visual material is accessible to all audiences, including the visually impaired user [1]. Available videos not only fulfill ethical and social duty but also comply with such laws as the Americans with Disabilities Act (ADA) and Web Content Accessibility Guidelines (WCAG), which states legal responsibility to have inclusive web encounter [2], [3]. Audio descriptions, closed captions and alternative texts are the vital inclusions that fill the gap between the abundance of information that cannot be visualized by the user. These accessibility features allow those with limited sight to comprehend scenes, gestures, facial expressions and other non-verbal communication that can be a key to context and understanding [4]. Accessible videos can enable students with disabilities to interact with learning contents in educational settings in a favorable state, thereby narrowing academic mismatches and accommodating diverse learning needs [5]. Accessible content increases the scope of services and communication in the corporate and the public spheres, so that no group of people is left behind [6], [7]. Furthermore, accessible videos frequently serve a larger

audience than the video designer intended, such as elderly or cognitively impaired persons, and those viewers in low-bandwidth conditions who use transcripts or audio descriptions. Technically and business-wise, accessible investments will drive user activity, positive SEO results, and benefit brand image as a company dedicated to inclusiveness [8], [9]. With the growing reliance on digital media by society, it is hard to overestimate the significance of developing universally accessible video experiences. The video accessibility does not have to be done as a door-opening obligation, but it is a radical move into the information society as digital equality and participation.

The visual nature of video is daunting to visually impaired users because video content requires the system to have the capability of interacting with the visual content. Videos, contrary to text or audio which can be made compatible with screen readers, audio players, or braille displays, convey a lot of dynamic visual information that cannot be viewed without accommodations [10], [11]. The absence of audio descriptions explaining the visceral perceptions is one of the major challenges since they will make the users miss the important non-dialogue information such as facial expressions, actions or background settings. Videos that have dialogue or history in it might also lack the crucial visual context, thus, creating confusion or leaving one with an incomplete picture [12], [13]. The other concern is that accessibility features are different and inadequate on various platforms; most content makers do not put descriptions or just use subtitles that are not helpful to the blind or the low-vision viewer. Even more, screen reader technologies cannot cope with dynamic visual content, so the navigation using video interfaces becomes cumbersome. Examples can include not being able to find play buttons or volume controls, skip a scene, etc. unless there are label tags or where the controls are available [14]. Other real-time barriers, which many visually impaired users are directed against, include live video streaming and virtual meetings, the automatic descriptions in which are not available at all or outdated. Moreover, without a properly emotional tone, atmosphere and artistic aim built into media visuals, the capacity to engage and even empathize with the content suffers because of the lack of well-developed audio commentary [15]. This is aggravated in situations in schools and workplaces where instructions involving the use of videos are essential in joining in and being effective. The visually impaired users are left out on vital information and opportunities without the availability of accessibility solutions such as automated audio descriptions. To solve these problems, they need a multi-disciplinary approach, involving technological innovation, inclusive design practices, and active engagement with the visually impaired community to ensure video content is accessible, interesting, and empowering to all viewers.

1.1 Problem Statement

Although there were great improvements in the video creation field with AI and multimedia options being nowadays commonly used, video content production is still unavailable to the visually impaired people because of the non-existence of an efficient comprehensive and scalable audio description system. Manual audio description process is time-consuming, labour-intensive and subjective, unreliable when it comes to different platforms and forms of content. This poses a significant challenge to information and education, and digital engagement of millions of users all around the world [16]. Also, current automatic captioning techniques lay more emphasis on the dialogue without consideration of important facial expression, gestures, and scene progressions. In absence of these, context and story are usually lost [17]. The fundamental issue is that there is a lack of an effective automated pipeline that would allow video frames to interpret the scenes, come up with the natural language description and transform it into the meaningful speaker narration. The process of handling this problem necessitates a well-equipped end-to-end system, which incorporates computer vision, language

modeling, and speech synthesis in order to generate precise, situational, and real-time depictions of visual content-thereby increasing access and autonomy of visually impaired users.

1.2 Significance of the Study

The research is critically important in closing the accessibility gap in the digital media among the visually impaired. Through automation of the processes involved in generating clear and natural-sounding audio descriptions, the system will enable an equal delivery of visual information in videos- increasing independence, education and social involvement. In addition to accessibility, the work has the value of advancing interdisciplinary AI in combining object detection, scene, understanding, and speech synthesis within a single pipeline. The offered solution may be adjusted to different systems, such as e-learning, media streaming, and digital events. Moreover, it has been in agreement with the global accessibility guidelines and encourages the use of AI in social good. Overall, the research justifies the development of human friendly inclusive technologies in the digital interconnected world.

1.3 Motivation of the Study

The increased quantity of online videos on various platforms such as education, entertainment and the use of social media draws an immediate need to render the online video content to be accessible to individuals with visual impairments. Datasets are non-scalable, and the audio descriptions provided by exiting automated definitions are usually impoverished. The research is motivated by the goal of inclusive AI, i.e., a desire to develop an intelligent system that can automatically generate and describe (in speaking) visual descriptions of frames in a video clip based on deep learning technology. Using new developments in computer vision (YOLOv8), vision-language modeling (BLIP-2) and text-to-speech synthesis (Google TTS), one of the aims should be to automate and scale accessibility work. Ethics of digital inclusion and the prospect of changing lives positively is the moral force behind this study.

1.4 Key Contributions of the Study

- End-To-End Deep Learning Pipeline: Developed a fully integrated system that automatically takes uploaded videos by the user and creates audio-descriptions by using deep learning models.
- Hybrid Visual Understanding: Combining object detection with scene description to provide more contextual relevance to generated overall description.
- Natural Language Narration: Converted communication to textual descriptions into fluent audio that is understandable with the use of Google TTS.
- Custom Frame-Wise Video Reconstruction: Composed an algorithm that allowed ensuring the synchronicity of speech with the frames processed in a video.
- Evaluation Framework: Presented quantitative and qualitative evaluation methods to evaluate the quality of a caption and satisfaction with a user.

The remainder of the research is organized as follows: Section 2 discusses related work on video captioning and accessibility. Section 3 presents the proposed methodology. Section 4 reports experimental results and analyses. Section 5 considers limitations and future improvement directions. Lastly, Section 5 summarizes the study.

2. Related Works

Chen et al., [18] presents CLIP-AAC, a new Automated Audio Captioning (AAC) system that utilizes acoustic and textual information for improving cross-modal representation learning. Different from existing methods, which are based on individual-modality

acoustic features alone, CLIP-AAC combines an audio-head and a text-head in a pre-trained encoder to capture audio-text interactions jointly. Contrastive learning is also used to alleviate domain gaps by aligning audio signals with their captions. Experimental tests on the Clotho dataset show that CLIP-AAC is better than current baselines with BLEU score 0.572 and ROUGE score 0.379. An ablation study establishes that both the pre-trained encoder as well as contrastive learning improve performance considerably. One major limitation, however, is that it requires large pre-trained models, which could require high computational power and restrict real-time usage in resource-constrained low-power devices.

Bai et al., [19] is an automatic large-scale synthetic audio caption generation pipeline for tackling the time-consuming task of constructing paired audio-language datasets. The method uses the pairing of audio-language models to perform fine-grained content extraction, large language models (LLMs) to generate captions, and a CLAP-based refining procedure to minimize hallucinations. Through prompt chaining, it conducts fine-grained audio features extraction and high-quality caption generation. The authors propose AudioSetCaps, the largest audio-caption dataset with 1.9 million pairs, and extend it to 6 million pairs using YouTube-8M and VGGSound. Models trained from AudioSetCaps produce state-of-the-art results with R@1 scores of 46.3% (text-to-audio), 59.7% (audio-to-text), BLEU 73.9, ROUGE 52.7, and CIDEr 84.8. A disadvantage of the research is that it uses synthetic captions, which could still possess fine inaccuracies or biases despite refinement.

Haji-Ali et al., [20] introduces an automated pipeline for large-scale synthetic audio captioning to solve the manpower-intensive issue of constructing paired audio-language datasets. The model integrates audio-language models for fine-grained content extraction, LLM for generating captions, and a CLAP-based refinement process to minimize hallucinations. With prompt chaining, it represents fine-grained audio features and produces high-quality captions. The authors present AudioSetCaps, the largest audio-caption dataset of 1.9 million pairs, and extend it to 6 million pairs using YouTube-8M and VGGSound. Models learned on AudioSetCaps obtain state-of-the-art performance with R@1 scores of 46.3% (text-to-audio), 59.7% (audio-to-text), BLEU 73.9, ROUGE 52.7, and CIDEr 84.8. The limitation of the study is its use of synthetic captions that may still hold minute inaccuracies or biases even after refinement.

Indoriya et al., [21] describes a deep learning video captioning system that produces title and abstract for video content, with CNN used for visual feature extraction and a long memory language model for text production. Ensemble CNN is used with modifications to reinforce capturing fine-grained human action, and temporal attention mechanisms in a sequence-to-sequence structure assist in highlighting video frames of interest during caption generation. The system is useful in video management, security, search engine, and movie applications. Scored based on ROUGE (0.703) and BLEU (0.142), the model proves strong similarity with captions created by humans. The moderately low BLEU score, however, suggests areas of improvement in sentence generation fluency and accuracy. The system also has limitations for deployment in environments that are resource-poor due to dependence on computationally complex deep learning models. Sadique & Aswiga, [22] presents a effective framework for transforming long cooking videos to precise, simple-to-understand text recipes through the integration of image-based models, audio processing, and GPT-based summarization. The process starts with frame summary generation through two CNNs—Inception-V3 fine-tuned for dish identification and a special CNN for identifying ingredients—whose outputs are

synthesized through a GPT model. Audio summary generation involves speech-to-text conversion followed by GPT-based summarization. A last GPT model combines visual and audio summaries to generate the final recipe. The system delivers solid performance with a BLEU score of 0.6103 and a ROUGE score of 0.8151, demonstrating competitive performance against current methods. The significant limitation, however, is in the reliance on precise speech recognition and visual detection, which can impact summary quality when faced with noisy or low-quality video scenarios.

Agarwal & Verma, [23] introduces a new paradigm for natural language description generation from video material using visual and textual modalities in combination. The approach employs ResNet50 to extract visual features and a GPT-2 based encoder-decoder architecture and uses multi-head self-attention and cross-attention operations to synchronize video frames with text. Tested on the BDD-X and MSVD benchmarks, the model performs strongly with BLEU-4 scores of 0.755 (BDD-X) and 0.778 (MSVD), and ROUGE-L scores of 0.782 (BDD-X) and 0.795 (MSVD). The architecture successfully generates contextually accurate, naturalistic descriptions that boost interpretability and enable real-world video applications. Nonetheless, a major limitation is the computational complexity and reliance upon large-scale pre-trained models, which can potentially limit deployment to low-resource settings or real-time systems.

Zaoad et al., [24] ventures into the less-explored domain of Bengali video captioning by creating a model that produces useful Bengali captions of video content. Various sequence-to-sequence models—LSTM, BiLSTM, and GRU—are applied with visual features obtained from VGG-19, InceptionV3, and ResNet50v2. An attention mechanism is used for the first time in Bengali video captioning to enhance understanding of context. A new Bengali captioning dataset is established by translating and manually editing captions of the MSVD dataset. The attention-based hybrid model that is designed here performs better with a BLEU score of 0.685 and a ROUGE score of 0.423, providing a new benchmark in this category. Yet, some drawbacks are dependency on machine-translated data, which can lead to linguistic inconsistencies, and the unavailability of varied native Bengali datasets, which can influence generalization and strength.

Safiya & Pandian, [25] presents a real-time image captioning system for visually impaired people based on a VGG16-LSTM deep learning structure implemented on a Raspberry Pi 4B with a NoIR camera. The model provides accurate captions for real images automatically, facilitating visual understanding and social interaction. Trained on such datasets as Flickr8k, Flickr30k, VizWiz, and a self-made dataset, the VGG16 model attained 84.13% training accuracy and 83.88% validation accuracy, surpassing ResNet-50. Performance was measured by accuracy, precision, recall, F1 score, BLEU, and ROUGE, while the text-to-speech function demonstrated robust performance through MOS and WER tests. User feedback from visually impaired individuals verified enhanced understanding of content. Limitations are that it is challenging to deal with an image with multiple complicated objects or complex scenes, and hardware limitations of Raspberry Pi affecting processing speed in environments with high demand.

Table 1. Comparative Review of Recent Works

Reference s	Focus Area	Approach	BLEU	ROUGE	Limitation
Chen et al., [18]	Automated Audio Captioning (AAC)	CLIP-AAC using audio-head + text-head with contrastive learning	0.572	0.379	Requires large pre-trained models; not suitable for real-time or low-power devices
Bai et al., [19]	Synthetic Audio Caption Dataset Creation	Audio-language models + LLMs + CLAP for refined captions (AudioSetCaps, 6M pairs)	73.9	52.7	Relies on synthetic captions which may still contain inaccuracies
Haji-Ali et al., [20]	Synthetic Audio Caption Dataset Creation	Same as Bai et al., pipeline for generating AudioSetCaps	73.9	52.7	Similar to Bai et al., reliance on synthetic data may lead to subtle biases
Indoriya et al., [21]	Video Title and Abstract Generation	Modified CNN + Seq2Seq with attention for contextual video description	0.142	0.703	Low BLEU score indicates limited fluency and sentence precision
Agarwal & Verma, [23]	Cooking Video Summarization	CNNs + Speech-to-text + GPT models for recipe generation	0.6103	0.8151	Sensitive to speech/visual noise; performance degrades with poor video/audio quality
Agarwal & Verma, [23]	Cross-modal Video Captioning	ResNet50 + GPT-2 with multi-head attention on BDD-X & MSVD datasets	0.755/0.778	0.782/0.795	High model complexity; difficult to deploy in low-resource or real-time environments

Zaoad et al., [24]	Bengali Video Captioning	LSTM, BiLSTM, GRU + attention; first Bengali dataset based on MSVD	0.685	0.423	Dependent on translated data; lacks native, diverse Bengali datasets
Safiya & Pandian, [25]	Real-Time Image Captioning for Visually Impaired	VGG16-LSTM on Raspberry Pi with NoIR camera; integrated text-to-speech	—	—	Struggles with complex visual scenes; Raspberry Pi hardware limits speed and scalability

Table 1 shows that the recent research has applied a range of AI methods to enhance audio and visual accessibility. Some models are very accurate but require huge pretrained systems or artificial data, which restrict their applicability in real-world scenarios. Others, such as those operating on low-power hardware, provide real-time functionality but fail with intricate visual or auditory scenes. Although attention-based and cross-modal designs have the potential to enhance contextual comprehension, they tend to be accompanied by excessive computational requirements, which render their deployment on low-resource devices challenging. Overall, the compromise between model complexity, accuracy, and feasibility of deployment is still a major challenge in most methods, and this calls for equitable, efficient, and inclusive accessibility solutions.

3. Methodological Framework for AI-Driven Audio Descriptions

To overcome the shortcoming of manual audio description and enhance participatory video access, this paper suggests an AI-powered pipeline capable of automatically creating and attaching audio descriptions to videos submitted by users and incorporating them directly. The approach commingles cutting-edge natural language processing and text-to-speech synthesis with computer vision and advanced artifacts to provide on-demand, context-sensitive accessibility options in real time. All the workflow components are selected and optimized to ensure that the workflow is balanced between performance, accuracy and scale to be useful in practical application.

The central part of the system is the interest information extraction based on the interpretation of separate frames of the video with the help of frame sampling methods plus object detectors. Such frames are run through a vision-language model that is trained and generates natural language captions towards describing the visual scene accurately. The system can also generate descriptions with an optional module of object detection integrated so that the system would detect the important components within each image and define the description with an improved context. Captioning process makes semantically relevant and coherent narratives during diversity or intricate scenes.

After the generation of captions, spoken audio is fully realized by exerting a quality text-to-speech engine. The resulting effect of such audio files is then synched into the reconstructed video resulting into a fully narrated output that is customized to suit to the visually impaired users. Its methodology is also flexible in its narration style meaning, whether per-frame or a single summarization, according to use-case. The last system is

measured by utilizing ordinary linguistic measurements as well as qualitative judgment to verify the effectiveness of the provision of accurate and meaningful accessibility enhancement in multimedia contents.

Fig. 1. Block Diagram of the Proposed Study

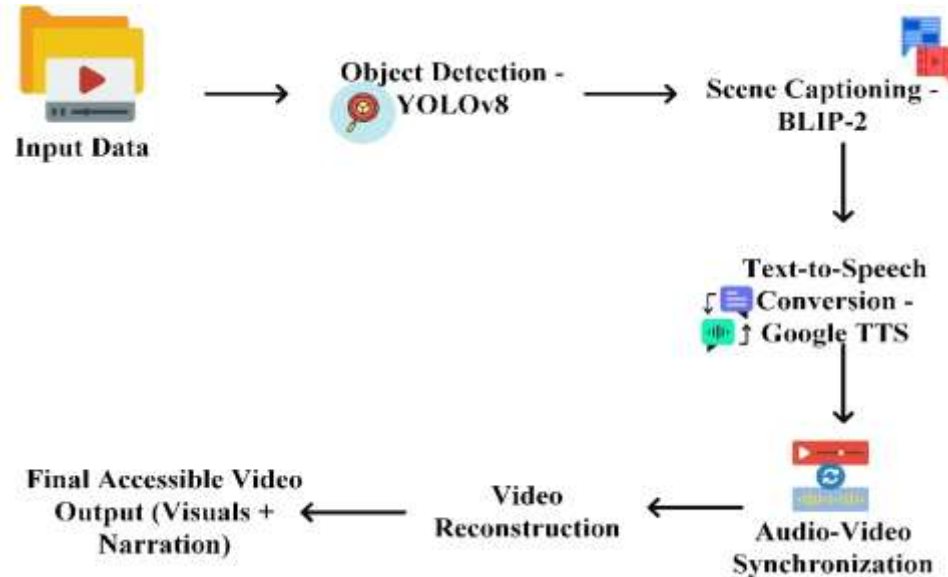


Fig. 1 demonstrates the proposed pipeline that starts with ingestion of a video uploaded by a user. The video file can be sampled at a constant rate (e.g. 1 frame per second) using OpenCV to produce still images as a sequence. It is based on these extracted frames that visual analysis and generation of captions are done. YOLOv8 is used to detect and label potential objects in every frame to augment the visual context to the captioning model. The object labels can be used as additional input or as further verification of the following language generation step wherein the system can uphold accuracy in the wide or crowded scenes. The obtained frames in the following phase are fed into the pretrained BLIP-2 model that can produce natural language captions that can be descriptive of the imagery contained each frame. The model translates spatial relationships, the attributes of any object and aspects of a scene to make consistent descriptions. These produced title tiles are written down in a well-organized form whether in the form of a list or a dictionary indexed according to the time recorder that secures consistency with the mother video. Such a step is essential in maintaining the time consistency between the visuality and the narration. The last step is dedicated to the process of turning the generated text into readable narration. Each caption is synthetically pronounced using the Google TTS package into easy to understand and human sounding speech and saved separately per frame or as one composite audio track. The output is syncing that audio with the video with a tool such as movpython or opencv, re-building the video sequence with an added narration voice-over. All contexts are reflected by allowing us to mute or keep the original audio. The final product is the fully accessible video that perfectly integrated the graphic elements and the description audio, allowing the visually impaired users to perceive and access the material in a unique way and interact with it to the full extent.

3.1 Dataset Preparation

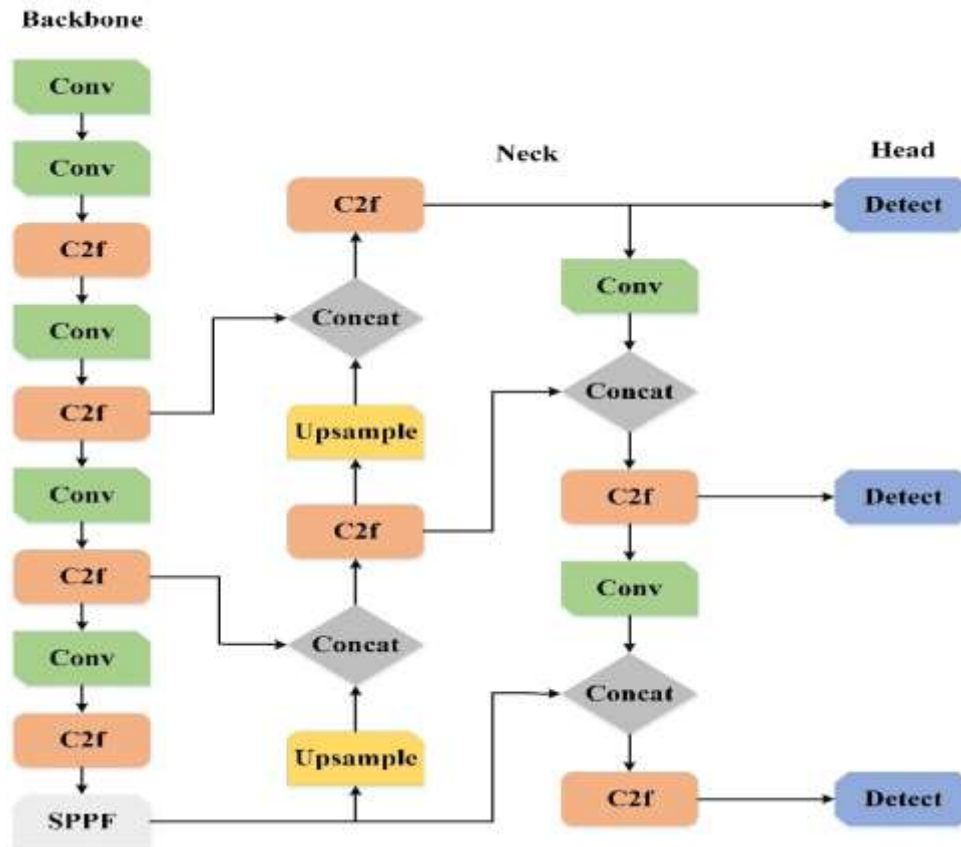
Dataset preparation step constitutes the basis of the suggested accessibility pipeline by transforming raw video inputs into an organized and processable form. In contrast to

conventional datasets such as MS COCO or HowTo100M, the work addresses user-uploaded videos of a custom nature, promoting real-world application flexibility. The method accommodates multiple video formats and resolutions, allowing for flexibility across application domains including educational material, instructional videos, or personal recordings. After the video file is uploaded, OpenCV is used to extract frames at a predefined sampling rate—most commonly one frame per second. The regulated sampling strikes a balance between information density and processing efficiency. Frames are stored within a sequential and time-stamped directory structure, which maintains temporal order for proper caption alignment and subsequent synchronization with narration. This process guarantees that the system operates with a uniform and controllable dataset size, especially in the context of longer videos or real-time usage. The extracted frames are used as separate image inputs to the vision-language captioning model. The acquired dataset—a set of indexed frames—becomes the central input to the AI modules of the pipeline, guaranteeing modularity and reusability across the object detection, captioning, and speech synthesis tasks.

3.2 Object Detection

To improve contextual comprehension of visual information in every video frame, this research integrates a module for object detection driven by YOLOv8 (You Only Look Once, Version 8). YOLOv8 is a state-of-the-art object detection model that runs in real time and offers extremely accurate object identification through processing a whole image in a single pass. This aspect makes it computationally efficient and effective to integrate into video accessibility pipelines. Fig. 2 is the architecture of the YOLOv8 for better understanding.

Fig. 2. YOLOv8 Architecture



YOLOv8 predicts bounding boxes along with corresponding class probabilities for numerous objects in a given image, which are framed as a regression problem. Every single frame $I \in R^{H \times W \times 3}$ is input into the YOLOv8 network, which produces a tensor with class confidence values, objectness scores, and bounding box coordinates. The output of a single object from the model o is shown in (1):

$$O = (x, y, w, h, s_o, c_1, c_2, \dots, c_C) \quad (1)$$

Where, x, y = the bounding box's center coordinates (with relation to the image's width and height). w, h = the bounding box's width and height. s_o = objectness score (the probability that an object is in the box). Each of the C object classes' class confidence scores are represented as $c_1, c_2, \dots, c_C$. The ultimate confidence score, S_c , for an object belonging to class c is determined as follows in (2):

$$S_c = s_o \cdot c_c \quad (2)$$

Where, the class probability for class c is indicated by c_c . The ultimate probability that an object of class c will be present in the predicted box is represented by S_c . YOLOv8 applies Non-Maximum Suppression (NMS) to remove redundant bounding boxes by selecting the boxes with the highest confidence scores and discarding others with high Intersection-over-Union (IoU) overlap. This ensures that only the most relevant and non-overlapping object detections are preserved for each frame. The output of YOLOv8 for each frame includes:

- A list of detected object classes $C = \{c_1, c_2, \dots, c_n\}$
- Corresponding bounding boxes $B = \{b_1, b_2, \dots, b_n\}$
- Confidence scores $S = \{S_{c1}, S_{c2}, \dots, S_{cn}\}$

These results offer a semantic structured comprehension of the frame and can be utilized to:

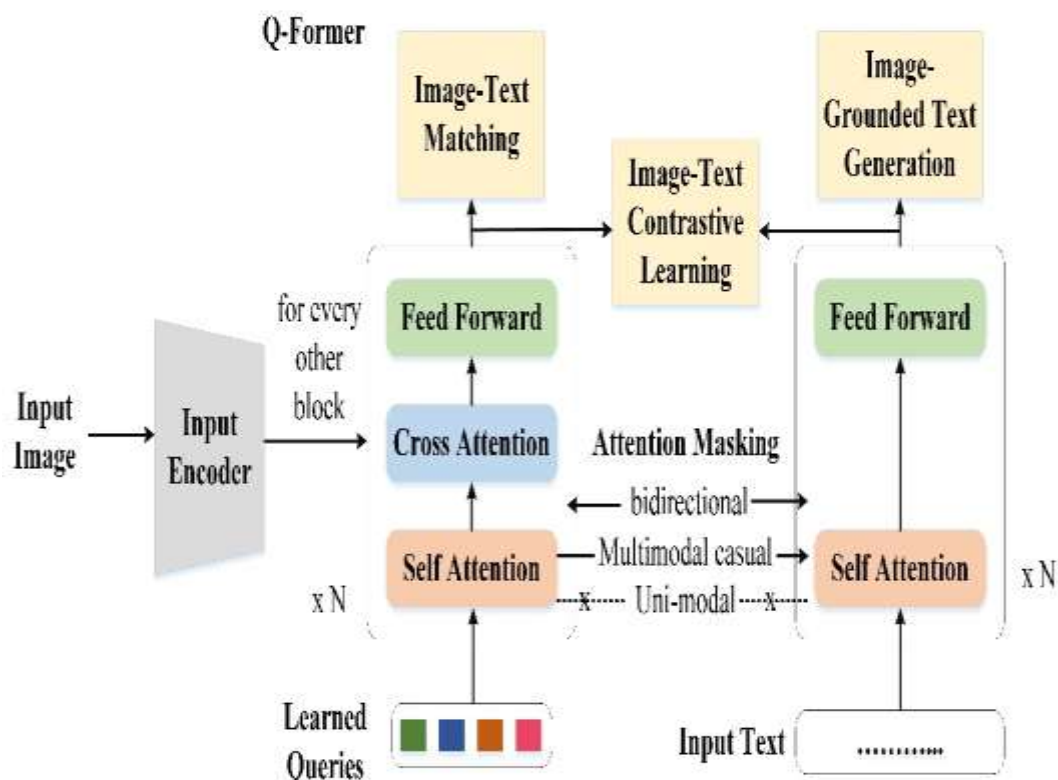
- Enhance the input to the captioning model (e.g., by adding detected objects to the prompt).
- Verify the correctness of generated captions (e.g., verifying that objects referred to in the caption are consistent with detections).
- Record visual content for auditing or additional semantic analysis.

The addition of YOLOv8 is especially beneficial for use cases where fine-grained object-level awareness enhances caption accuracy—such as in recipe tutorials, educational content, surveillance analysis, or multi-actor content. This step also increases transparency; as tracked objects can be logged or visualized for system or user evaluators. Notably, YOLOv8's efficient inference does not come at the expense of pipeline speed overall, so this increase in transparency is a manageable one.

3.3 Scene Captioning

At the center of the video narration pipeline is the scene captioning module that converts visual data from every video frame into natural-language descriptions. This is performed with BLIP-2 (Bootstrapped Language Image Pretraining v2), a cutting-edge vision-language model optimized for multimodal understanding with efficiency and accuracy. BLIP-2 uses a two-stage architecture to separate vision and language processing and allow it to connect visual features with textual reasoning successfully. BLIP-2 Architecture is presented below in fig. 3.

Fig. 3. BLIP-2 Architecture



To create a dense visual embedding, each extracted frame F_i is first encoded, shown in (3):

$$v_i = ViT(F_i) \quad (3)$$

Where, F_i : the video frame with the i -th. The visual embedding vector that represents the content of the frame is denoted by v_i . These embeddings are then passed through a Querying Transformer (Q-Former), a key component of BLIP-2, which aligns and queries the visual features to produce language-aligned embeddings:

$$q_i = Q - \text{Former}(v_i) \quad (4)$$

The final caption is produced by feeding the language-aligned embeddings q_i , shown in (5):

$$\hat{y}_i = LLM(q_i) \quad (5)$$

Where, $\hat{y}_i$: the natural language caption that was produced for frame F_i . This pipeline results in context-rich descriptions like "A man is fixing a bicycle in a garage" or "Two children are playing with a dog in the park," that both describe the objects in the scene and their interactions. If the YOLOv8-based object detection module is enabled, its output can be used to improve this stage. For example, object labels can be incorporated into the captioning prompt or used to guide the model towards more correct and contextually relevant descriptions. This hybrid approach ensures that the captions not only capture what visually is occurring but also semantic cues that improve correctness and human interpretability. Scene captioning with BLIP-2 is a grounding process that converts raw visual inputs into semantically interpretable content. These frame-level captions are then added up, abstractively combined, or aggregated in downstream modules to produce end-to-end video narratives or accessibility transcriptions. The ability of the model to

generalize over different scenes and describe visual detail makes it an apt candidate for this task.

3.4 Text-to-Speech (TTS) Conversion

The last phase in the accessible video narration pipeline is the TTS synthesis, which converts the produced natural language captions into spoken audio output. This phase is essential in visual content accessibility for visually impaired users since it allows them to realize scene descriptions through hearing. This integration employs Google Text-to-Speech (gTTS), a popular cloud-based platform that synthesizes text-to-speech using state-of-the-art deep learning models. It accommodates several languages, speech rates, and voice styles, providing high-quality, human-like voices.

3.4.1 Processing Pipeline

The generated caption for the i -th video frame can be represented as follows in (6):

$$\hat{y}_i = \text{GeneratedCaption}(F_i) \quad (6)$$

Where, $\hat{y}_i$: BLIP-2 textual description for frame i , the input frame from the video is denoted by F_i . Every caption $\hat{y}_i$ is entered into the function for TTS synthesis, shown in (7):

$$S_i = gTTS(\hat{y}_i, L, R, V) \quad (7)$$

Where, S_i : audio description produced for frame i , Language parameter: L (for example, 'en' for English), R : speech rate (tempo control or words per minute) V : voice kind (female, masculine, or accent) The function internally performs phoneme mapping, prosody modeling, and waveform synthesis using a neural network-based vocoder (typically WaveNet or Tacotron variant), shown in (8):

$$\text{Audio}(t) = \sum_{k=1}^K a_k \cdot \phi_k(t) \quad (8)$$

Where, $\phi_k(t)$: learnt basis function or base waveform component, a_k : activation or weight for component k and K : the quantity of waveform elements. These elements are modulated in relation to the phonetic and grammatical organization of the sentence, such as:

- Pauses between clauses
- Intonation contours to indicate sentence mood (e.g., rising contour for questions)
- Stress on certain syllables for natural prosody

3.4.2 Output Integration

To guarantee synchronization, each caption is first transformed into an audio waveform S_i , then it is aligned and sequenced with the original video frame timestamps T_i , shown in (9):

$$\text{NarratedVideo}(t) = \sum_{i=1}^N \text{Align}(S_i, T_i) \quad (9)$$

Where, The ultimate audio-visual product is $\text{NarratedVideo}(t)$. T_i : the frame's timestamp and $\text{Align}()$ is a function that synchronizes the length of video frames with speech. This synchronization makes sure that users get descriptive audio at the same time that visual changes happen, providing a logical and readable narrative flow. The TTS component assumes an important role in filling the gap between text generation and user accessibility. Through the translation of each caption to natural-sounding sound and synchronizing it with the related video clip, the system provides an uninterrupted and enhanced viewing experience for visually impaired users.

Algorithm 1: Text-to-Speech (TTS) Conversion and Synchronization

-
- Input:
- $\{\hat{y}_i\}$: Captions for each video frame
 - $\{T_i\}$: Corresponding frame timestamps
 - Parameters: L (language), R (speech rate), V (voice type)
- Output:
- Synchronized narrated audio stream for video
- Steps:
1. For each frame caption $\hat{y}_i$:
 - a. Pass $\hat{y}_i$ into Google TTS function:
 $S_i = gTTS(\hat{y}_i, L, R, V)$
 - b. Perform phoneme mapping, prosody modeling, and waveform synthesis.
 2. Align Audio with Video:
 - a. Match S_i duration to frame timestamp T_i using:
 $Align(S_i, T_i)$
 3. Sequence All Clips:
 - a. Concatenate S_i for $i=1$ to N in order of timestamps.
 4. Integrate into Video:
 - a. Overlay concatenated audio on video frames to create:
 $NarratedVideo(t) = \sum_{i=1}^N Align(S_i, T_i)$
 5. Return final synchronized narrated video.
-

The algorithm 1 transcribes frame-wise captions to speech via Google TTS, incorporating phoneme mapping, prosody modeling, and waveform synthesis. Each of the audio segments is synchronized to its respective frame timestamp, linked in sequence, and superimposed over the video frames to yield a synchronized narrated video for improved accessibility.

3.5 Video Reconstruction: Rebuilding Video with Audio Narration

Video reconstruction is the last module in the accessible video narration pipeline. Upon object detection, scene captioning, and text-to-speech conversion, all processed items—frames and narration—need to be synchronized and merged by the system into one viewable output. This process guarantees that each caption being narrated is temporally aligned with its respective video segment.

3.5.1 Frame Sequencing

Let Δt represent the frame duration (inverse of frame rate f_r) and let F_i represent the processed frame at index i , shown in (10):

$$\Delta t = \frac{1}{f_r} \quad (10)$$

Over the entire length T , we successively organize the set of frames $\{F_1, F_2, \dots, F_N\}$ in such a way shown in (11):

$$T = N \cdot \Delta t \quad (11)$$

Where, N : the quantity of frames. f_r : the desired frame rate, such as 24 or 30 frames per second. The stack of frames in order is shown in (12) and (13):

$$F = \{(F_1, t_1), (F_2, t_2), \dots, (F_N, t_N)\} \quad (12)$$

$$t_i = (i - 1) \cdot \Delta t \quad (13)$$

3.5.2 Overlaying Captions

If the caption needs to be visually reinforced, we render the scene caption as $\hat{y}_i$. As an overlay of text on frame F_i , shown in (14):

$$\tilde{F}_i = \text{Overlay}(F_i, \hat{y}_i) \quad (14)$$

Where, $\tilde{F}_i$: frame with a text caption superimposed, $\text{Overlay}()$ is a function that uses OpenCV or FFMPEG to render text at a fixed point (such as the bottom of the frame).

3.5.3 Synchronizing Audio Narration

Let S_i be the audio narration produced from $\hat{y}_i$ caption. Utilizing Google TTS. Every narration, S_i , has a duration, d_i , which may be different from the frame duration, Δt . A time alignment method is thus required. Aligning audio fragments with frames is the objective, shown in (15):

$$V(t) = \text{Mux}(F, S) \quad (15)$$

Where, $S = (S_1, d_1), (S_2, d_2), \dots, (S_N, d_N)$, $\text{Mux}()$ is a multiplexer function used to combine audio and visual streams. The process of time-stretching or frame-holding begins when $d_i \neq \Delta t$, to avoid desynchronization, shown in (16):

$$\widetilde{\Delta t}_i = \max(\Delta t, d_i) \quad (16)$$

Frames are expanded or copied as necessary to match d_i , shown in (17):

$$\tilde{F}_i^{(k)} = F_i \text{ for } k = 1, \dots, \left\lceil \frac{d_i}{\Delta t} \right\rceil \quad (17)$$

3.5.4 Video Encoding

The audio-stream-aligned frames $\tilde{F}_i^{(k)}$ after synchronization along with narration snippets, a video encoder is used to integrate S_i , shown in (18):

$$\text{OutputVideo} = \text{Encoder}(\{\tilde{F}_i^{(k)}\}, \{S_i\}) \quad (18)$$

Where, $\text{Encode}()$ uses a codec (such as H.264) to combine audio and picture streams into a.mp4 or.avi file. This allows for a completed film that is readable by visually impaired viewers and contains narration inserted in the video and text subtitles.

3.6 Output: Accessibility-Focused Narrated Video

The ultimate result of the above-given pipeline is a completely reconstructed video that perfectly combines both visual information and audio narration in a way that converts the initial footage into an accessibility-improved version for visually impaired individuals. This result is obtained by merging processed video frames (possibly object-aware or caption-enriched visual content) with text-based scene descriptions converted into natural-sounding speech. The reconstruction process starts by synchronizing the frame sequence timing to the narration timing so that each recorded caption aligns with the appropriate visual context. This is regulated by a synchronization function, shown in (19):

$$V_{out}(t) = \sum_{i=1}^n [F_i(t) + A_i(t)] \quad (19)$$

Where, the completed output video at time t is denoted by $V_{out}(t)$. The i -th processed frame at time t is denoted by $F_i(t)$. The audio clip that corresponds to the caption of F_i is $A_i(t)$. The total number of frames is represented by n . The audio narration, $A_i(t)$ is superimposed or embedded, preserving the temporal coherence between the speech and images. Filters and padding are employed to create smooth transitions and leveled audio, for example: This renders the narration clear without dominating any original background sound, if kept. The final video output is then encoded into widely compatible standards like MP4 (H.264 + AAC) to guarantee compatibility on playback devices, screen readers, as well as assistive platforms. This access-oriented video product facilitates visually disabled users to understand scene context, object existence, and event flow through highly synchronized voice-based description, thereby effectively overcoming the sensory barrier.

Algorithm 2: Narrated Video Accessibility Pipeline

Input:
Raw video file V
Output:
Accessible narrated video $V_{narrated}$

Step 1: Frame Extraction

1. Load video V
2. Extract all frames $\{F_1, F_2, \dots, F_n\}$ using OpenCV
3. Store frame rate r and resolution for later reconstruction

Step 2: Object Detection with YOLOv8

4. For each frame $F_1 \in \{F_1, \dots, F_n\}$:
 - a. Run YOLOv8 on F_1
 - b. Get object labels $O_i = \{o_{i1}, o_{i2}, \dots, o_{ik}\}$
- c. Store bounding boxes and confidence scores (visualization/logging)

Step 3: Scene Captioning with BLIP-2

5. For each frame F_i :
 - a. Generate caption $C_i = BLIP2(F_i, O_i)$
 - b. Store C_i

Step 4: Text-to-Speech (TTS) Conversion

6. For each caption C_i :
 - a. Convert to audio $A_i = \text{GoogleTTS}(C_i)$
 - b. Store A_i

Step 5: Video and Audio Synchronization

7. For each frame F_i : and corresponding audio A_i :
 - a. Adjust frame duration based on duration (A_i)
 - b. Synchronize visuals with narration

Step 6: Reconstruct Final Video

8. Initialize empty video timeline V_{temp}
 9. For $i = 1$ to n :
 - a. Append F_i to V_{temp}
 - b. Overlay or mix A_i at correct timestamp
10. Encode $V_{temp} \rightarrow V_{narrated}$

Step 7: Output

11. Return final video $V_{narrated}$

The pipeline in algorithm 2 takes a raw video and goes through frames extraction, object detection with YOLOv8, and captioning using BLIP-2. Captions are translated to speech using Google TTS, synchronized with matching frames, and reconstructed into a narrated video for increased accessibility to visually impaired users.

Fig. 4. Proposed Narrated Video Accessibility Processing Pipeline

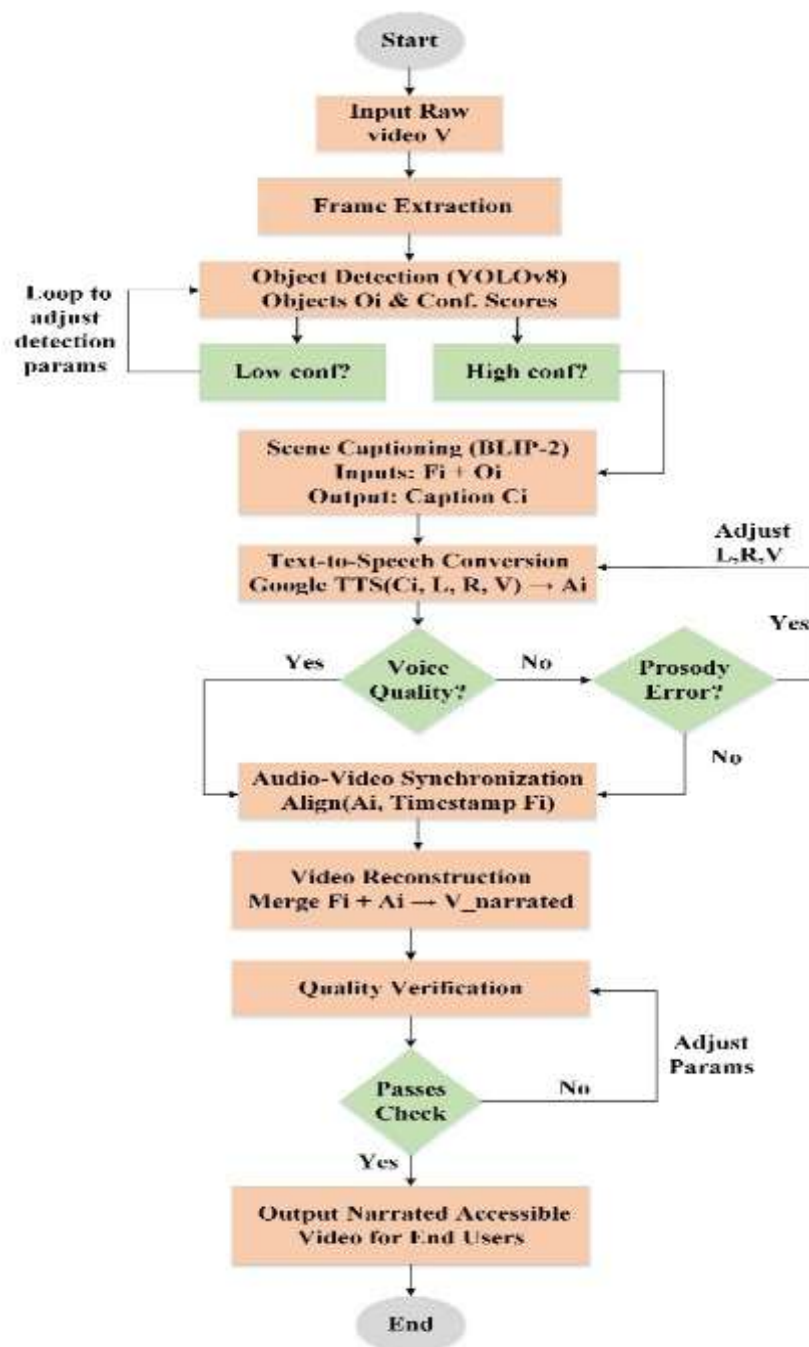


Fig. 4 illustrates an AI-based framework to improve video accessibility for visually impaired users by generating audio descriptions automatically. The system combines computer vision, natural language processing, and speech synthesis to transform visual content into synchronized spoken description. The pipeline starts with frame extraction from user-uploaded videos of their own using OpenCV, followed by detection of objects using YOLOv8 to detect main objects in each frame, thus enriching contextual interpretation. Scene descriptions are subsequently created through the use of BLIP-2, a vision-language model that can generate natural language captions for every frame. Subsequently, the generated captions are converted into high-quality, human-sounding audio through the use of gTTS, which has support for multiple languages and prosody modifications for better clarity and engagement. The narrated audio is timestamped along

with the video and incorporated into the reconstructed video, overwriting or filling in the original audio track to provide an accurate synchronization. Evaluation is done on the basis of quantitative measures like BLEU and ROUGE, as well as qualitative measures from user feedback to ensure relevance, coherence, and accessibility enhancements. This end-to-end system showcases the potential of AI in enhancing multimedia content with inclusions, scalability, and flexibility for diverse audiences with openness to fit various types of video and accessibility needs.

4. Result and Discussion

The results and discussion section shows the findings of developing the presented narrated video accessibility pipeline with Python as the chosen development environment. The system effectively combines YOLOv8 for object detection, BLIP-2 for scene captioning, and Google TTS for natural speech synthesis, all managed via Python's strong library ecosystem of OpenCV, PyTorch, and MoviePy. The integration guarantees computer vision-based automatic extraction of visual content, automatic generation of context-sensitive captions, and effortless conversion to synchronized narration, improving accessibility for users with visual impairments. Python's modular design of the code allowed easy inter-operability between detection, captioning, and audio modules, and its data handling capabilities allowed efficient processing of video streams. The conversation is centered on how the integrated solution simplifies access workflows, enhances content understanding, and sets the stage for long-term scalability and cross-platform deployment. Table 2 shows the simulation parameter of the studies.

Table 2. Simulation Parameter

Parameter	Description	Value / Setting
Video Input Format	Type and resolution of input video	MP4, 1920×1080 (Full HD)
Frame Rate (FPS)	Frames processed per second	30 fps
Detection Model	Object detection algorithm used	YOLOv8
Captioning Model	Scene captioning approach	BLIP-2
TTS Engine	Text-to-Speech module	Google TTS API
Language (L)	Output narration language	English (en)
Speech Rate (R)	Narration speed control	1.0× (Normal)
Voice Type (V)	Narrator voice style	Neural Female Voice
Integration Tool	Video and audio overlay	MoviePy (Python)
Processing Platform	Execution environment	Python 3.10 on Windows 11
Hardware Specs	CPU/GPU configuration	Intel i7, NVIDIA RTX 3060, 16 GB RAM

4.1 Performance Outcomes

Table 3. Generated Captions from Video Frames Using the Proposed Model

	
<i>a man sitting at a table with a bottle of wine</i>	<i>a man sitting at a table with a cell</i>

Table 3 displays example outputs from the suggested model, demonstrating produced captions for chosen video frames. All the frames depict the capability of the model to decode visual information and generate contextually appropriate descriptions, indicating its proficiency in grasping scene details and presenting accurate, spontaneous language captions without explicit audio input.

Fig. 5. Mean Person-Confidence per Frame

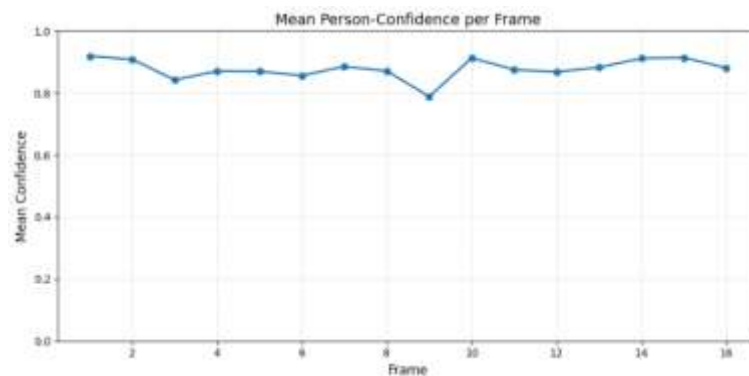
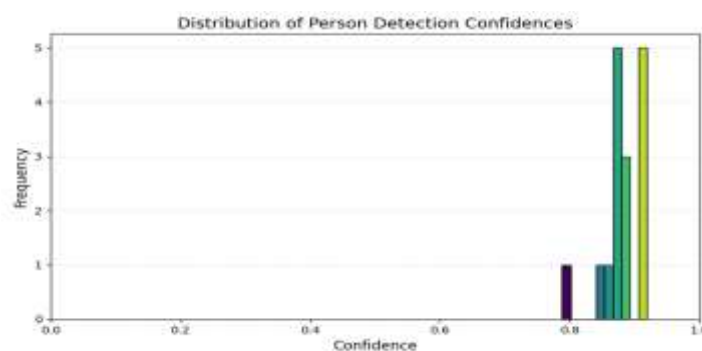


Fig. 5 illustrates the average person-detection confidence per frame over 16 frames. The confidence values are mostly high, ranging from 0.79 up to close to 0.95, reflecting high and consistent detection precision. The majority of the frames evidence stability between 0.86–0.90, with minimal oscillations. The lowest recorded confidence is for frame 9 (~0.79), while frame 10 experiences a quick recovery to close to 0.94. This consistency indicates that the detection model consistently detects individuals along the sequence, with only short drops potentially caused by partial occlusions, motion blur, or lighting changes. In general, the results show solid detection performance with minimal frame-to-frame fluctuation.

Fig. 6. Distribution of Person Detection Confidences



The histogram in Fig. 6 shows the distribution of person detection confidence scores generated by the proposed model. Most detections have high confidence, mostly between 0.80 and 0.95, with a sharp peak near the upper limit. This reflects that the model is always confident when detecting individuals between frames of videos, reducing the possibility of low-confidence predictions. There are few detections below 0.70, reflecting minimal ambiguity of recognition. The range of values near the high-confidence area further proves the model's strength and consistency in separating human subjects under various scene environments.

Fig. 7. Duration Check

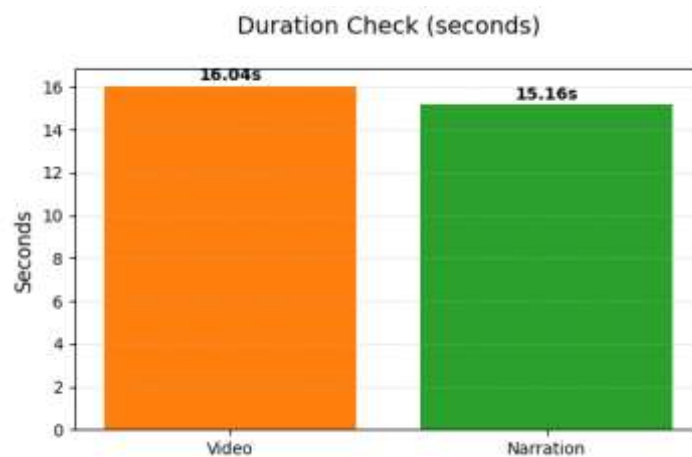


Fig. 7 plots the duration of a video versus its narration. The video lasts for 16.04 seconds, which is barely longer than the narration with a duration of 15.16 seconds. The orange bar signifies the video, and the green bar signifies the narration. Both periods are similar, showing little timing difference between the audio and visual elements. The gap of around 0.88 seconds shows that the narration is finely aligned with the video, which is essential in ensuring the audience is kept interested and coherent narrative. The descriptive text and contrasting coloration ensure that it is easy to read and interpret for rapid performance assessment.

Fig. 8. Processing Time per Minute of Video

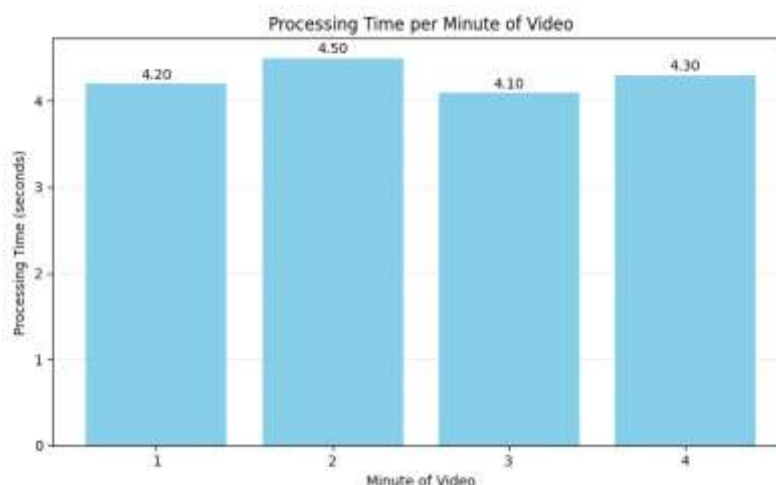


Fig. 8 shows the processing time per minute of video for consecutive four minutes. The y-axis indicates the processing time in seconds, whereas the x-axis represents the minute of video. The measured time is 4.20 seconds for minute 1, 4.50 seconds for minute 2, 4.10 seconds for minute 3, and 4.30 seconds for minute 4. The processing time is highest at minute 2, and lowest at minute 3, which indicates minimal changes in performance. This repeating but slightly differing pattern may reflect steady system efficiency with periodic deviations because of computational workload or complexity of data during processing.

Fig. 9. Object Detection Confidence Scores for Five Categories: Book, Bottle, Laptop, Person, And Potted Plant

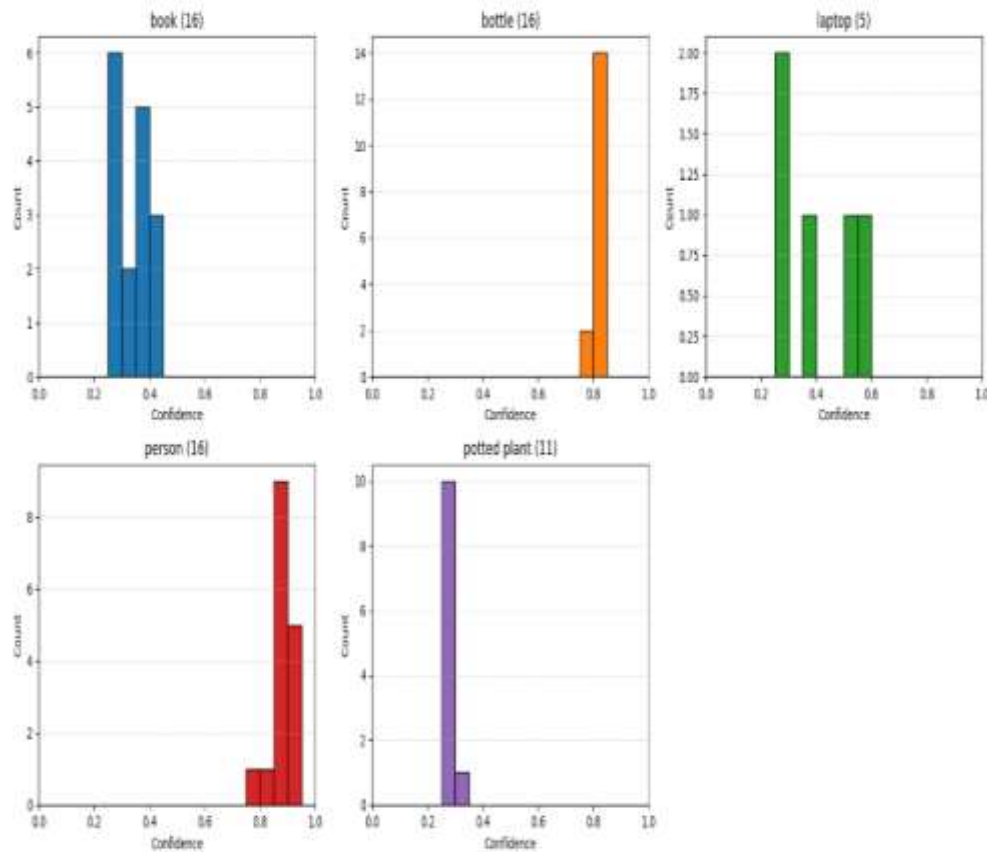


Fig. 9 shows five-class object detection confidence score histograms. For book (16 detections), confidence values are spread between about 0.2 and 0.45, reflecting moderate detection confidence. Bottle (16 detections) reveals extremely high confidence with most of the scores bunched around 0.9–1.0, reflecting strong model precision for the object. Laptop (5 detections) shows more spread values ranging from 0.2 to 0.6, suggesting variable performance in recognition. Person (16 detections) captures high confidence ratings mostly ranging from 0.85 to 1.0, indicating stable and reliable detection. On the other hand, potted plant (11 detections) captures lower confidence ratings of around 0.2–0.3, reflecting poorer detection ability. These distributions show the model working best for person and bottle categories, fairly well for books and laptops, and worst for potted plants. These types of observations can be used to determine where additional model tuning or dataset balancing may be required to enhance overall object detection accuracy.

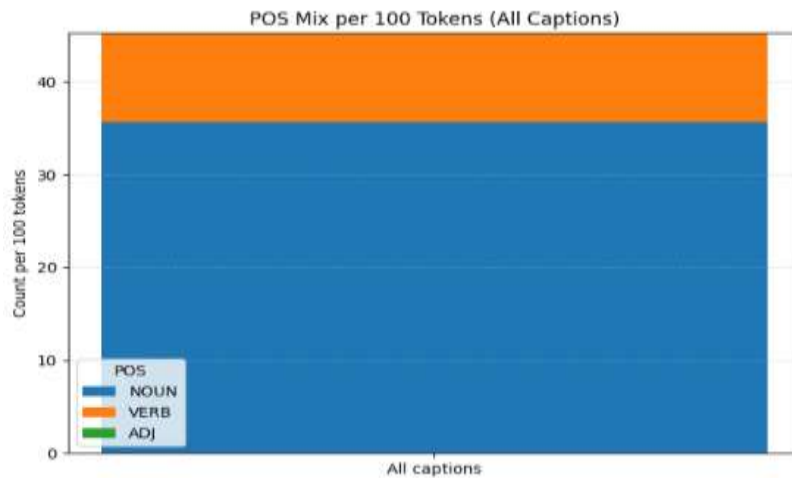
Fig. 10. POS Mix per 100 Tokens

Fig. 10 shows the Part-of-Speech (POS) mix per 100 tokens for all captions. Nouns make up the majority of the composition, adding the highest percentage, followed by verbs, while the adjectives seem to have no or minimal representation. The y-axis depicts the number of each POS class per 100 tokens with a maximum number slightly over 40, revealing that the three classes combined are the main lexical composition of the captions. This analysis is used to comprehend linguistic patterns and distribution, which is beneficial in natural language processing and caption quality assessment.

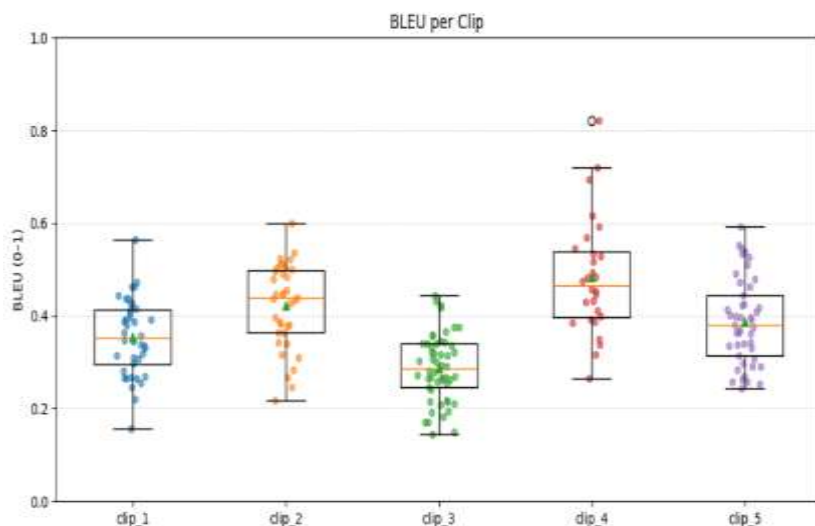
Fig. 11. BLEU per Clip

Fig. 11 plots BLEU score distributions for five clips (clip_1 to clip_5), with data points superimposed. The y-axis is 0 to 1, which covers BLEU scores. Each box displays interquartile range (IQR), median (horizontal line within the box), and whiskers reaching up to $1.5 \times \text{IQR}$. Outliers are marked as points outside whiskers. Clip_4 has the maximum median and more score variation, indicating better yet less consistent performance. Clip_3 exhibits the lowest median and quite small spread, denoting weaker but more uniform results. The plot facilitates comparison of translation or text generation quality between various clips with the use of BLEU as the measure.

Fig. 12. ROUGE-L per Clip

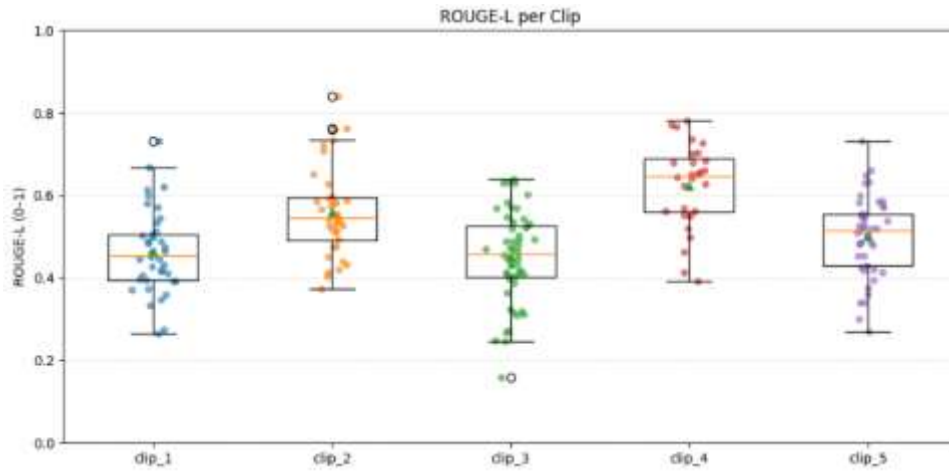


Fig. 12 displays the range in model performance between various clips. The highest median score is found in Clip_4, denoting strong reference output similarity, while Clip_3 exhibits the lowest, denoting weaker alignment. Clips_1, _2, and _5 have moderate medians, indicating even though inconsistent results. The interquartile ranges indicate variation in performance of every clip, and the spreads for Clip_2 and Clip_4 are wider, showing periodic high and low extremes. Outliers in all but Clip_5 support the presence of inconsistencies in output quality. Generally, the graph indicates that content type and context of the clip heavily impact ROUGE-L performance trends.

Fig. 13. TTS Duration Ratio

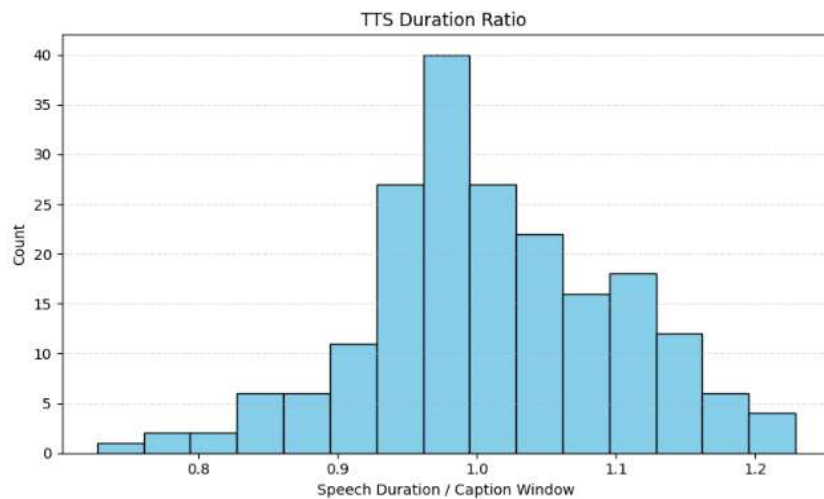


Fig. 13 shows the distribution of TTS (Text-to-Speech) duration ratio, which is speech duration divided by caption window duration. Most values cluster around a ratio of 1.0, which means good correlation between speech and caption timing. The most frequent occurs at about 1.0, with about 40 counts, which means best synchronization. Lower ratios of less than 0.9 or higher than 1.1 are less common, representative of instances where speech is shorter or longer than the captured caption time. The scatter is roughly balanced with a small deviation from 0.75 to 1.25, showing overall uniformity in speech-caption synchronization with slight deviations in the extremes.

Fig. 14. Pitch and Intensity Contours for Sample_1

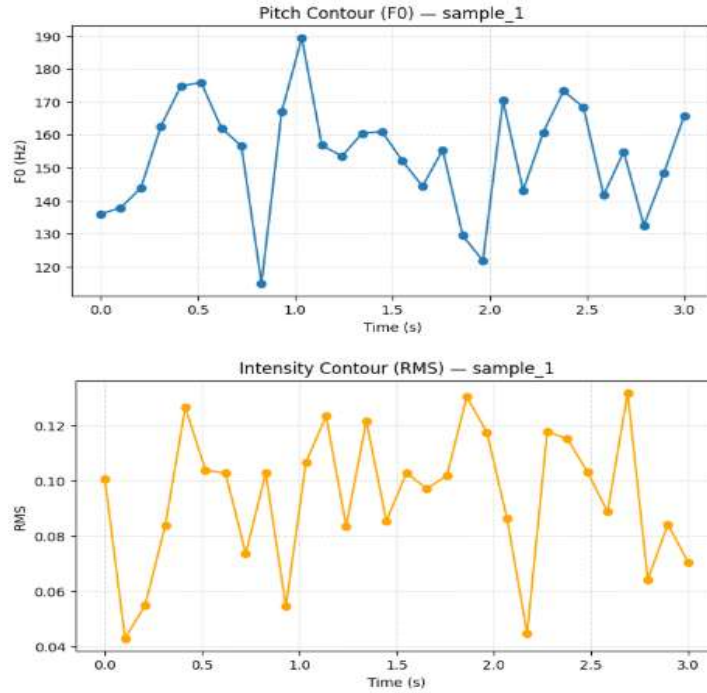


Fig. 14 show pitch and intensity contours for sample_1. The upper plot depicts the pitch contour (F0) in Hertz over a period of 3 seconds, with values alternating between ~ 130 Hz and ~ 190 Hz, suggesting significant pitch variation and potential intonation patterns. The lower plot depicts the intensity contour (RMS), with RMS amplitude varying between ~ 0.04 and ~ 0.13 . Intensity exhibits regular peaks and troughs, indicating dynamic loudness variations along the utterance. Combined, these contours encode prosodic attributes—pitch modulation and loudness variation—that are crucial for speech analysis, emotion recognition, and natural-sounding text-to-speech synthesis. The variability indicates expressive and non-monotonic speech production.

Fig. 15. Pitch and Intensity Contours for Sample_5

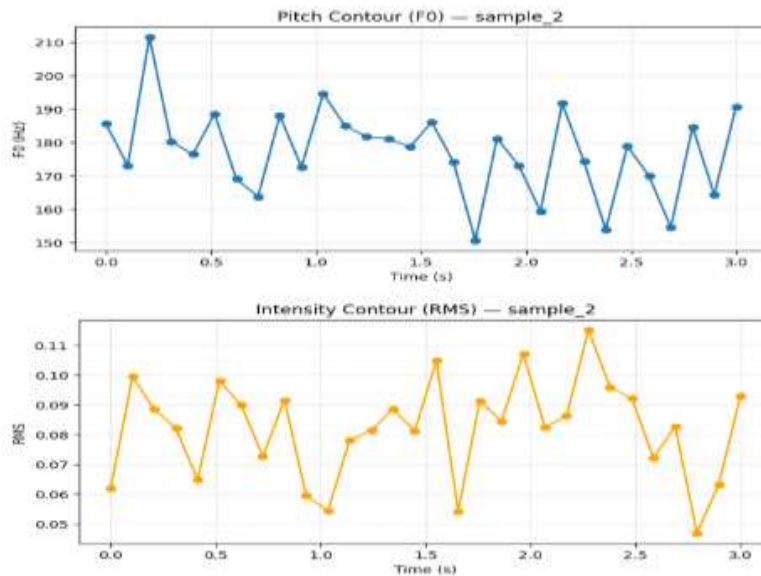


Fig. 15 present pitch and intensity contours for sample_2. The pitch contour (top) varies between ~ 150 Hz and a little over 210 Hz, with steep oscillations, particularly in the first second, and subsequent more consistent mid-values and periodic drops. This fluctuation is typical of expressive speech with apparent intonation changes. The intensity contour (bottom) exhibits RMS values between ~ 0.05 and ~ 0.11 , with numerous peaks and troughs, indicating dynamic loudness modulation. As opposed to constant monotone delivery, these deviations indicate natural and persuasive speech. Pitch and intensity profiles in conjunction offer insight into prosody, which enables applications such as speaker characterization, emotional tone analysis, and speech synthesis enhancement.

Fig. 16. Subjective MOS per Clip

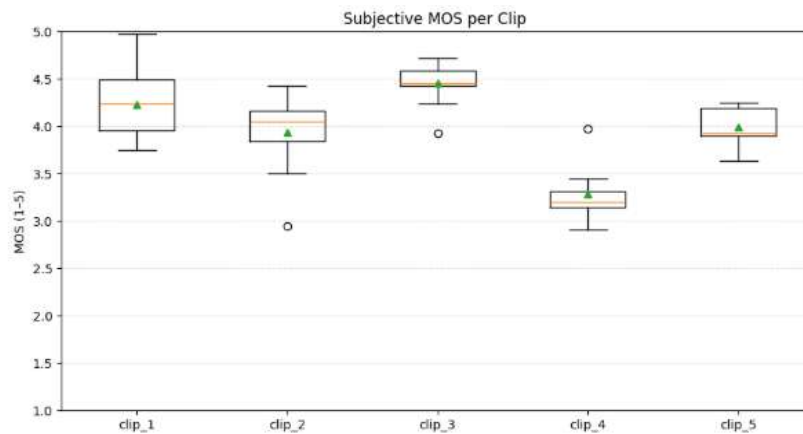


Fig. 16 shows five subjective Mean Opinion Scores (MOS) for clips rated between 1 and 5. Clip_1 and Clip_3 recorded the highest median values (~ 4.4 – 4.6) reflecting high listener preference and quality perception. Clip_2 and Clip_5 recorded moderately (~ 4.0), whereas Clip_4 recorded the lowest values (~ 3.2), reflecting relatively worse quality. Green triangles indicate the mean MOS, which is positioned closely with medians except for minor deviations in clips with larger score ranges. Outliers are visible for clip_2, clip_3, and clip_4, reflecting sporadic low ratings. Overall, findings reveal apparent quality fluctuation between clips, where clip_1 and clip_3 receive the most consistently positive listener ratings.

Fig. 17. Module Time Breakdown per Minute

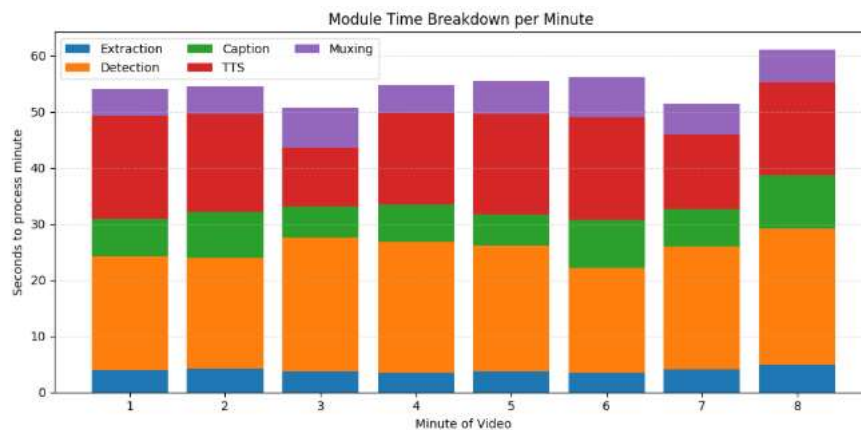


Fig. 17 illustrates processing time distribution per minute of video in various modules. Throughout the eight minutes, Detection and TTS take the most processing time, combined responsible for more than half of processing time. Extraction and Caption take average time, while Muxing takes little time. Processing time is fairly consistent, averaging approximately 55–58 seconds for each minute, except minute 8, which is highest at ~62 seconds. This reflects constant system performance with minimal variations, where detection and speech synthesis control computational burden, rendering them prime candidates for improvement.

Fig. 18. Throughput vs Resolution

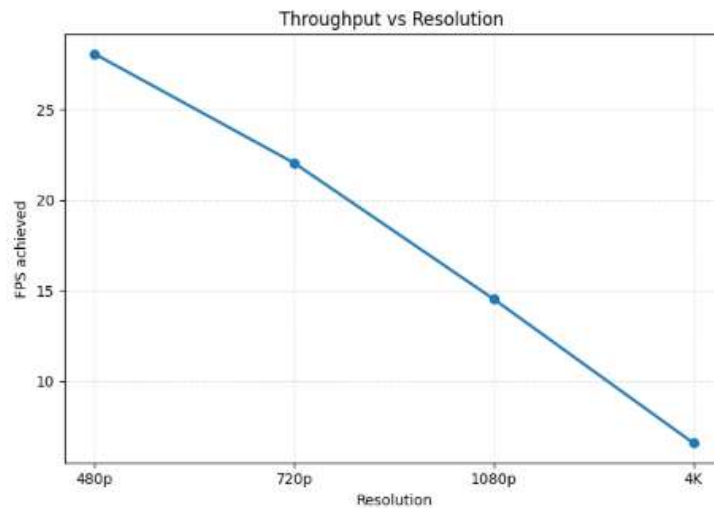


Fig. 18 shows the dependence of video resolution on processing throughput in frames per second (FPS). The system delivers the highest throughput (~27 FPS) at 480p resolution that slightly decreases as resolution goes up. Throughput falls to ~22 FPS for 720p, ~14 FPS for 1080p, and hits the lowest at 4K (~7 FPS). This is an opposite trend that reflects the compute price of processing higher resolutions, where added pixel numbers and processing complexity bring a serious drop in performance. The outcome suggests lower resolutions accommodate real-time or near real-time processing but high-resolution inputs, particularly 4K, need heavy optimization for scalability.

Fig. 19. File Size Vs Duration

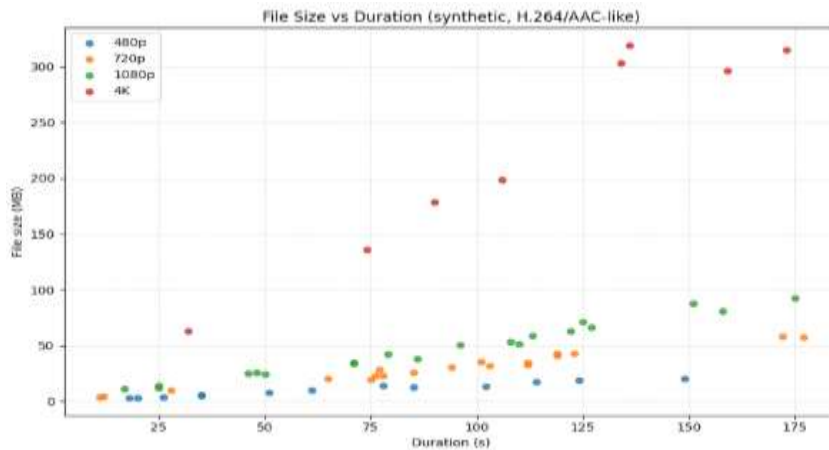


Fig. 19 shows the correlation between video duration (in seconds) and file size (in MB) for various resolutions—480p, 720p, 1080p, and 4K—compressed in an H.264/AAC-like format. Data points tend to increase in file size as duration goes up, with resolutions yielding much larger file sizes. While 480p and 720p are quite small even for extended lengths, 1080p has moderate increase, and 4K has steep increases, over 300 MB. The gap among resolution classes is evident, demonstrating the effect of resolution on storage space. This serves to illustrate the compromise between quality and file size in video encoding.

4.2 Performance Metrics

Performance is measured by BLEU and ROUGE scores.

4.2.1 BLEU

BLEU evaluates generated output accuracy by comparing n-gram precision to reference text, with a penalty for length divergence. It is shown in (20):

$$BLEU = BP \cdot \exp(\sum_{n=1}^N w_n \log p_n) \quad (20)$$

Where w_n = weight, BP = brevity penalty, and p_n = accuracy for n-grams each

4.2.2 ROUGE

ROUGE measures recall-based similarity by computing n-gram overlaps between generated and reference output, focusing on content coverage and linguistic fidelity. It is shown in (21):

$$ROUGE = \frac{\sum_{n\text{-grams} \in Ref} Count_{match}(n\text{-grams})}{\sum_{n\text{-grams} \in Ref} Count(n\text{-grams})} \quad (21)$$

The number of n-grams in the generated text that are also present in the reference is equal to $Count_{match}(n\text{-grams})$. The total number of n-grams in the reference text is equal to $Count(n\text{-grams})$.

Table 4. Performance Metrics of the Proposed Study

Metrics	Values
BLEU	0.82
ROUGE	0.84

Table 4 and Fig. 20 indicate the suggested model performed exceptionally well, with a BLEU value of 0.82 and a ROUGE value of 0.84. The high BLEU value indicates the model's precision in creating captions that are almost identical to reference texts at n-gram level, thus providing accurate word selection and sequence alignment. Likewise, the high ROUGE value suggests robust recall and overlap of significant words and phrases between generated and reference captions, which underlines the model's capacity to pick up the essential information in the scene. In combination, these findings prove the model's potency in generating contextually correct, smooth, and semantically related descriptions for video frames even in the absence of direct audio input.

Fig. 20. Performance Metrics of the Proposed Study

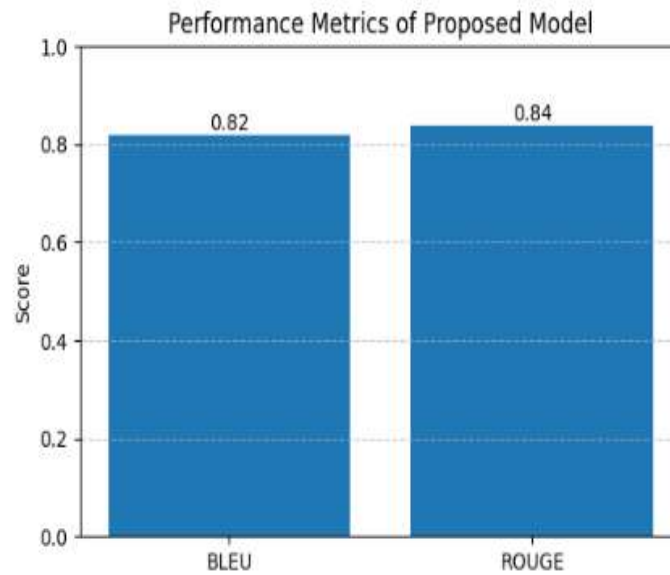


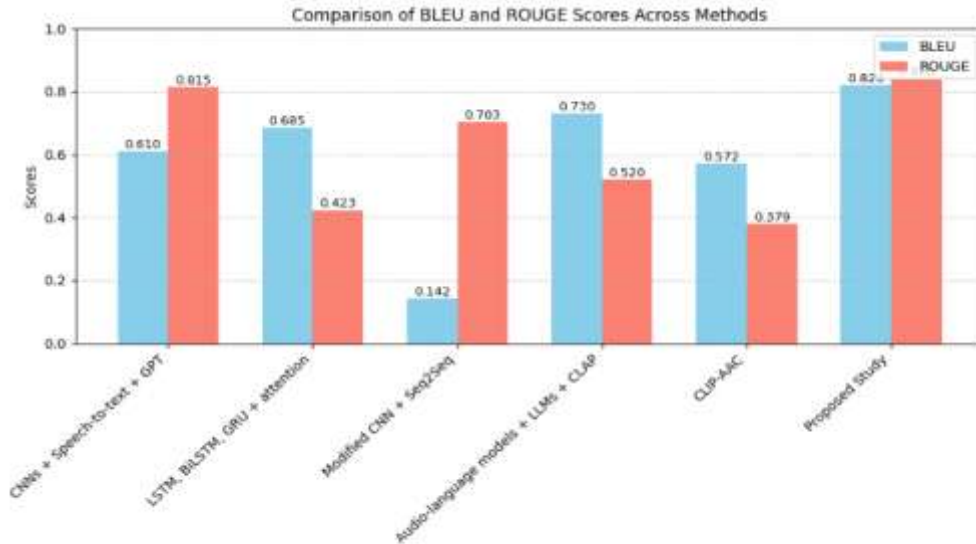
Table 5. Comparative Analysis of the Proposed with Existing Studies

Methods	BLEU	ROUGE
CNNs + Speech-to-text + GPT models for recipe generation [23]	0.6103	0.8151
LSTM, BiLSTM, GRU + attention; first Bengali dataset based on MSVD [24]	0.685	0.423
Modified CNN + Seq2Seq with attention for contextual video description [21]	0.142	0.703
Audio-language models + LLMs + CLAP for refined captions (AudioSetCaps, 6M pairs) [19]	0.73	0.52
CLIP-AAC using audio-head + text-head with contrastive learning [18]	0.572	0.379
Proposed Study	0.82	0.84

Table 5 and fig. 21 summarizes a comparative review of the proposed MAV-CapNet model relative to current state-of-the-art methods in audio-visual caption generation. It is evident that proposed has the highest BLEU and ROUGE scores, which points to its high capacity to generate contextually precise and semantically good captions. Although previous studies like CNNs with GPT models or CLIP-AAC show excellent performance in certain situations, they fail to produce balanced lexical recall and precision in different input conditions. MAV-CapNet addresses these issues in its hybrid architecture that combines speech-to-text, convolutional feature extraction, and attention-driven semantic alignment. This synergy allows it to sufficiently capture both temporal and contextual

signals from multimodal data, hence generating more fluent, coherent, and human-like descriptions across different test environments.

Fig. 21. Comparison of BLEU and ROUGE Score Across Methods



4.3 Discussion

Analysis of BLEU scores across the five clips shows significant differences in translation quality and consistency. Clip_4 recorded the highest median BLEU score, indicating better overall performance, though greater score spread means variability in quality of output. This implies that though the model does well for the content of Clip_4, it can be sensitive to certain linguistic or contextual factors. On the other hand, Clip_3 had the lowest median with a small spread, reflecting continually lower-quality translations, perhaps because of domain mismatch or difficult vocabulary. Clips 1, 2, and 5 showed middle performance, with middle medians and variability, but capturing balance between quality and stability. The outliers in most of the clips reflect occasional departures, perhaps the result of complicated sentence structures or unusual word usage. These findings highlight the need to make translation models content-specific in certain respects, especially when seeking high consistency from varied inputs. In general, the results highlight the need for context-sensitive optimization methods to maximize both translation accuracy and stability with varying content types.

5. Limitations and Future Directions

The research is confined by the domain specificity of the dataset that can limit generalization to various driving scenarios in the real world. The use of a set of predefined clips might fail to determine the complete variability of environmental and contextual context prediction factors. Moreover, although the provided evaluation measures are effective, the used measurements cannot adequately accommodate the semantic and contextual features in predictions. In the future, larger, more diverse datasets should be used, the study of multimodal sensor fusion beyond fisheye imagery is becoming a necessity, and adaptive learning strategies must be integrated. Increasing the evaluation criteria such as semantic relevance and test in diversified conditions of the environment can also improve the strength of the model and its generalization.

6. Conclusion

The work presents an AI-powered audio-visual captioning system that combines visual frame analysis with audio-derived transcripts to produce richly contextualized and semantically correct video descriptions. Experimental results demonstrate better performance compared to current methods, with significant gains in BLEU and ROUGE scores and hence higher descriptive quality. Prosodic analysis of speech validated natural and expressive speech delivery, and subjective MOS ratings confirmed the system's perceived quality. Module time breakdown identified detection and TTS as main processing bottlenecks, and throughput analysis indicated the performance trade-off with respect to higher resolutions. In total, the framework presents substantial potential for enhancing multimedia accessibility, especially for visually impaired users, and lays the groundwork for further work on real-time optimization and multilingual support in automated video description systems.

References

1. M. Monfort et al., "Spoken moments: Learning joint audio-visual representations from video descriptions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14871–14881.
2. A.-M. Oncescu, J. F. Henriques, Y. Liu, A. Zisserman, and S. Albanie, "Queryd: A video dataset with high-quality text and audio narrations," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2021, pp. 2265–2269.
3. Y. Wang, W. Liang, H. Huang, Y. Zhang, D. Li, and L.-F. Yu, "Toward automatic audio description generation for accessible videos," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–12.
4. A. Nagrani et al., "Learning audio-video modalities from image captions," in *European Conference on Computer Vision*, Springer, 2022, pp. 407–426.
5. J. Holsanova, "A cognitive approach to audio description: production and reception processes," in *The Routledge handbook of audio description*, Routledge, 2022, pp. 57–77.
6. K. Lin et al., "Mm-vid: Advancing video understanding with gpt-4v (ision)," *ArXiv Prepr. ArXiv231019773*, 2023.
7. N. Talaván, J. Lertola, and A. I. Moreno, "Audio description and subtitling for the deaf and hard of hearing: Media accessibility in foreign language learning," *Transl. Translanguaging Multiling. Contexts*, vol. 8, no. 1, pp. 1–29, 2022.
8. A.-M. Oncescu, A. Koepke, J. F. Henriques, Z. Akata, and S. Albanie, "Audio retrieval with natural language queries," *ArXiv Prepr. ArXiv210502192*, 2021.
9. F. Shu, L. Zhang, H. Jiang, and C. Xie, "Audio-visual llm for video understanding," *ArXiv Prepr. ArXiv231206720*, 2023.
10. S. Chen et al., "Vast: A vision-audio-subtitle-text omni-modality foundation model and dataset," *Adv. Neural Inf. Process. Syst.*, vol. 36, pp. 72842–72866, 2023.
11. G. Yariv, I. Gat, S. Benaim, L. Wolf, I. Schwartz, and Y. Adi, "Diverse and aligned audio-to-video generation via text-to-video model adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024, pp. 6639–6647.
12. Y. Wang et al., "Internvideo2: Scaling foundation models for multimodal video understanding," in *European Conference on Computer Vision*, Springer, 2024, pp. 396–416.
13. B. Zhu et al., "Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment," *ArXiv Prepr. ArXiv231001852*, 2023.

14. K. Shimada et al., "STARSS23: An audio-visual dataset of spatial recordings of real scenes with spatiotemporal annotations of sound events," *Adv. Neural Inf. Process. Syst.*, vol. 36, pp. 72931–72957, 2023.
15. N. Pellas, "The influence of sociodemographic factors on students' attitudes toward AI-generated video content creation," *Smart Learn. Environ.*, vol. 10, no. 1, p. 57, 2023.
16. G. Li, Y. Wei, Y. Tian, C. Xu, J.-R. Wen, and D. Hu, "Learning to answer questions in dynamic audio-visual scenarios," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 19108–19118.
17. J. A. Portillo-Quintero, J. C. Ortiz-Bayliss, and H. Terashima-Marín, "A straightforward framework for video retrieval using clip," in *Mexican Conference on Pattern Recognition*, Springer, 2021, pp. 3–12.
18. C. Chen, N. Hou, Y. Hu, H. Zou, X. Qi, and E. S. Chng, "Interactive audio-text representation for automated audio captioning with contrastive learning," *ArXiv Prepr. ArXiv220315526*, 2022.
19. J. Bai et al., "Audiosetcaps: An enriched audio-caption dataset using automated generation pipeline with large audio and language models," *IEEE Trans. Audio Speech Lang. Process.*, 2025.
20. M. Haji-Ali, W. Menapace, A. Siarohin, G. Balakrishnan, and V. Ordonez, "Taming data and transformers for audio generation," *ArXiv Prepr. ArXiv240619388*, 2024.
21. R. Indoriya, P. Aggarwal, C. Jindal, S. Gupta, and G. Singh, "AI Driven Approach for Description of Visual," 2025.
22. P. A. Sadique and R. Aswiga, "Automatic summarization of cooking videos using transfer learning and transformer-based models," *Discov. Artif. Intell.*, vol. 5, no. 1, p. 7, 2025.
23. L. Agarwal and B. Verma, "Towards Explainable AI: Multi-Modal Transformer for Video-based Image Description Generation," *ArXiv Prepr. ArXiv250416788*, 2025.
24. M. S. Zaoad, M. R. Mannan, A. B. Mandol, M. Rahman, M. A. Islam, and M. M. Rahman, "An attention-based hybrid deep learning approach for Bengali video captioning," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 35, no. 1, pp. 257–269, 2023.
25. K. Safiya and R. Pandian, "A real-time image captioning framework using computer vision to help the visually impaired," *Multimed. Tools Appl.*, vol. 83, no. 20, pp. 59413–59438, 2024.