Deep Spatio-Temporal Video Captioning for CCTV Surveillance

D Balaji Anirudh
Dept. of Information Technology
NIT Karnataka
Surathkal, Karnataka
balajianirudh.221it026@nitk.edu.in

Jyotsana Achal
Dept. of Information Technology
NIT Karnataka
Surathkal, Karnataka
jyotsanaachal.221it032@nitk.edu.in

Sricharan Sridhar
Dept. of Information Technology
NIT Karnataka
Surathkal, Karnataka
sricharans.221it066@nitk.edu.in

***Abstract*—This report presents a hybrid deep learning framework for automated video captioning and semantic timestamp retrieval, designed to interpret and summarize surveillance or general video streams. The proposed system integrates spatial and temporal cues through a combined Convolutional Neural Network (CNN) and Fast Fourier Transform (FFT) representation. The CNN (ResNet-50 backbone) extracts high-level spatial features from individual frames, while the FFT captures temporal periodicity and motion dynamics in the frequency domain. These fused spatio-temporal embeddings are processed by a Bidirectional Long Short-Term Memory (BiLSTM) network with an attention mechanism to generate context-aware captions describing scene activity. Finally, the generated captions are embedded using Sentence-BERT and indexed in a vector database, enabling semantic search and timestamp forwarding based on user queries. This approach demonstrates the potential of frequency-domain temporal modeling and neural attention for bridging visual understanding and natural language generation in intelligent video analysis systems.**

***Index Terms*—Video Captioning, CNN, FFT, BiLSTM, Attention Mechanism, Semantic Search, Deep Learning, Spatio-Temporal Features.**

I. Introduction

Video captioning is an advanced task in computer vision and natural language processing (NLP) that involves generating descriptive natural language sentences from video content. It integrates spatio-temporal understanding, semantic representation, and linguistic generation, requiring models to capture both visual dynamics and contextual meaning. Recent advancements in deep learning have led to effective architectures that combine convolutional neural networks (CNNs) for visual feature extraction with recurrent neural networks (RNNs) for sequential language modeling.

While traditional approaches rely primarily on temporal pooling or recurrent units to summarize sequential frames, such methods often lose fine-grained temporal information critical for complex scenes. To address this limitation, frequency-domain techniques such as the Fast Fourier Transform (FFT) have been explored to encode motion periodicity and temporal consistency. By combining CNN-derived spatial features with FFT-based temporal representations, a hybrid feature space can more effectively model video dynamics.

Furthermore, attention mechanisms have enhanced the interpretability and accuracy of sequence generation by allowing the decoder to focus on salient frames at each step of caption prediction. In this work, we propose a hybrid framework that fuses CNN and FFT representations, employs a bidirectional LSTM (BiLSTM) encoder, and integrates attention-based decoding for improved spatio-temporal modeling. The system is further extended with semantic search capabilities through vector embeddings, enabling timestamp-level video retrieval based on generated captions.

The remainder of this paper is structured as follows: Section II reviews related literature in video captioning architectures. Section III details the proposed methodology, including hybrid feature extraction, sequence modeling, and semantic indexing. Section IV discusses implementation and experimental observations. Section V concludes with insights and future directions.

II. Literature Survey

Automatic image and video captioning has seen significant advancements in recent years due to the rise of deep learning. Early methods relied on templates or retrieval-based approaches, which often generated rigid captions and failed to capture the full semantic content of images or videos. With the introduction of encoder-decoder frameworks, models began to generate captions that were more descriptive and contextually accurate [1], [3]–[5].

A. Sequence-to-Sequence Learning

Sutskever et al. [4] proposed the sequence-to-sequence framework, which laid the foundation for modern captioning systems. This approach uses two separate LSTM networks: an encoder and a decoder. The encoder compresses an input sequence $x_1, x_2, ..., x_T$ into a fixed-size context vector v, summarizing the entire input. The decoder then generates the output sequence $y_1, y_2, ..., y_{T'}$ word by word using:

$$p(y_1, ..., y_{T'} | x_1, ..., x_T) = \prod_{t=1}^{T'} p(y_t | v, y_1, ..., y_{t-1}) \qquad (1)$$

This formulation allows the model to handle sequences of different lengths, a critical requirement for video captioning, where the number of frames can vary.

B. CNN-LSTM Architectures

Donahue et al. [3] proposed Long-term Recurrent Convolutional Networks (LRCN) that combine CNNs and LSTMs for sequential visual tasks. Each video frame is processed through a CNN to extract spatial features:

$$f_t = CNN(x_t) \tag{2}$$

The extracted feature vectors are then passed to LSTM layers to model temporal dependencies:

$$i_t = \sigma(W_{xi}f_t + W_{hi}h_{t-1} + b_i) \tag{3}$$

$$f_t = \sigma(W_{xf}f_t + W_{hf}h_{t-1} + b_f) \tag{4}$$

$$o_t = \sigma(W_{xo}f_t + W_{ho}h_{t-1} + b_o) \tag{5}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc}f_t + W_{hc}h_{t-1} + b_c) \tag{6}$$

$$h_t = o_t \odot \tanh(c_t) \tag{7}$$

This architecture allows the model to understand both the content of each frame and how it changes over time. LRCNs have been widely used for activity recognition and video captioning due to their simplicity and effectiveness.

C. Attention Mechanisms

Attention mechanisms enhance captioning models by allowing them to focus on the most relevant parts of the input sequence at each decoding step [5], [9]. For BiLSTM outputs h_i, attention weights a_i are computed as:

$$a_i = \frac{\exp(e_i)}{\sum_j \exp(e_j)}, \quad e_i = \tanh(W^T h_i + b) \tag{8}$$

The context vector c_t is then a weighted sum of hidden states:

$$c_t = \sum_i a_i h_i \tag{9}$$

This mechanism helps the decoder generate more accurate captions by dynamically selecting which frames are most important for each word.

D. Frequency-Domain Temporal Modeling

Recent works have explored the use of Fast Fourier Transform (FFT) to capture temporal patterns in videos [8]. FFT converts the temporal sequence of frame features into the frequency domain:

$$F_k = \sum_{t=0}^{T-1} f_t \, e^{-2\pi ikt/T}, \quad k = 0, 1, ..., T-1 \tag{10}$$

Frequency-domain features highlight repeating motions or periodic patterns, such as walking or running, which are less obvious in the time domain. Combining these with spatial CNN features creates a richer, hybrid representation for caption generation.

E. Graph-based Caption Summarization

Dense video captioning requires summarizing multiple events into coherent sentences. Graph Convolutional Networks (GCNs) have been applied to model semantic relationships between words in generated captions [4]. Each word is a node in a graph, and its features Z_i are updated as:

$$\hat{Z}_i = \tanh\left(Z_i + W \sum_{j \in \mathcal{P}_i} \alpha_{ij} Z_j\right) \tag{11}$$

where α_{ij} are attention weights. This approach preserves the logical flow and relationships across segments, improving the quality of summarized captions.

F. Survey Insights

Comprehensive surveys [9], [10] confirm that modern video captioning relies heavily on the encoder-decoder paradigm. Important enhancements include attention mechanisms, BiLSTMs, transformer-based models, and hybrid feature representations. The surveys also emphasize the importance of large-scale datasets such as MS COCO, Flickr8k/30k, and ActivityNet, along with evaluation metrics like BLEU, METEOR, ROUGE-L, and CIDEr.

In summary, the literature shows a clear evolution from basic CNN-LSTM models to sophisticated systems that combine spatial, temporal, and frequency-domain features, attention mechanisms, and graph-based summarization. These advancements form the foundation for the hybrid captioning system implemented in this work.

III. Methodology

The proposed video captioning system integrates Convolutional Neural Networks (CNN), Fast Fourier Transform (FFT), and a Bidirectional Long Short-Term Memory (BiLSTM) network, followed by an Attention-based decoder. The architecture aims to capture both the spatial and temporal dependencies present in videos while ensuring efficient and coherent caption generation. Figure 1 shows the overall flow of the system.

A. Overview of the System

The system begins with the user providing a video input. Each video undergoes preprocessing, where frames are extracted and normalized. During training, each video is associated with one or more reference captions used for supervision. The core pipeline consists of four major stages: feature extraction, feature fusion, sequence modeling and decoding, and output generation.

B. Dataset and Preprocessing

The preprocessing stage involves frame sampling at uniform intervals to reduce redundancy. Each video frame is resized and normalized before being passed to the CNN for spatial feature extraction. Initially, image enhancement was performed using *Contrast Limited Adaptive Histogram Equalization (CLAHE)* to improve contrast; however, this method showed limitations when dealing with diverse subjects, varying illumination, and differences in clothing color. To overcome these challenges, we adopted a deep learning-based enhancement technique, Zero-DCE (Zero-Reference Deep Curve Estimation), which dynamically adjusts pixel intensities through a learned curve estimation process. Zero-DCE optimizes multiple loss components such as *spatial consistency loss*, *exposure control loss*, and *color constancy loss*, thereby ensuring balanced illumination and preserving natural color distributions. Integrating Zero-DCE into the preprocessing workflow significantly improved the visual quality of frames, leading to more robust and discriminative feature extraction by the CNN.

Formally, if a video $V = \{F_1, F_2, ..., F_T\}$ consists of T frames, each enhanced and preprocessed frame is transformed into a feature vector:

$$f_t = \text{CNN}(F_t), \quad t = 1, 2, ..., T \tag{12}$$

where $f_t \in \mathbb{R}^d$ represents the spatial descriptor of frame t.

C. Feature Extraction using CNN and FFT

CNNs are responsible for capturing the spatial context of each frame, such as objects, textures, and interactions between entities. However, video sequences also contain motion and frequency patterns over time. To capture these temporal variations, FFT is applied across the sequence of CNN feature vectors:

$$F_k = \sum_{t=0}^{T-1} f_t \, e^{-2\pi i k t / T}, \quad k = 0, 1, ..., T-1 \tag{13}$$

Here, F_k represents the frequency-domain representation of frame-level features, highlighting periodic or repetitive actions like walking or waving.

D. Feature Fusion

Both feature types—spatial (f_t) and frequency-domain (F_k)—are then fused to form a combined representation. A simple yet effective approach is linear concatenation followed by a dense transformation:

$$z_t = W_z[f_t; F_t] + b_z \tag{14}$$

where $[f_t; F_t]$ denotes the concatenation of features and W_z, b_z are trainable parameters. This fused representation z_t captures both instantaneous visual details and long-range temporal structure.

E. Sequence Modeling with BiLSTM and Attention

The fused feature vectors are passed into a Bidirectional LSTM to capture contextual dependencies from both forward and backward temporal directions:

$$\overrightarrow{h_t} = \text{LSTM}_{\text{fwd}}(z_t, \overrightarrow{h_{t-1}}) \tag{15}$$

$$\overleftarrow{h_t} = \text{LSTM}_{\text{bwd}}(z_t, \overleftarrow{h_{t+1}}) \tag{16}$$

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \tag{17}$$

An attention mechanism computes the relevance of each hidden state to the current decoding step. The attention weights a_t are obtained using:

$$a_t = \frac{\exp(e_t)}{\sum_j \exp(e_j)}, \quad e_t = v_a^T \tanh(W_a h_t + b_a) \tag{18}$$

The context vector is then:

$$c = \sum_t a_t h_t \tag{19}$$

Finally, the decoder LSTM uses this context to predict the next word token in the caption sequence.

F. Training and Evaluation

During training, the model minimizes cross-entropy loss between the predicted word distribution and the ground truth word:

$$\mathcal{L} = -\sum_{t=1}^{T'} \log p(y_t | y_{1:t-1}, c) \tag{20}$$

Performance is evaluated using metrics such as BLEU, METEOR, and CIDEr to measure caption accuracy, fluency, and relevance.

G. Live Captioning and Keyword-Based Navigation

For real-time inference, the trained model directly generates captions as the video plays. A post-processing module performs keyword-based timestamp forwarding, enabling navigation to specific segments of the video corresponding to key events or objects identified in the generated captions. For example, when the keyword “car” appears in a caption, the system can automatically highlight or jump to that segment in the video.

H. Summary of Workflow

The methodology can thus be summarized as follows:

1) Input video and associated captions are preprocessed.
2) Spatial features are extracted via CNN, while FFT captures temporal frequency information.
3) The two feature streams are fused to form a hybrid representation.
4) BiLSTM with attention decodes these representations into meaningful textual captions.
5) During live captioning, generated keywords enable timestamp-based navigation.

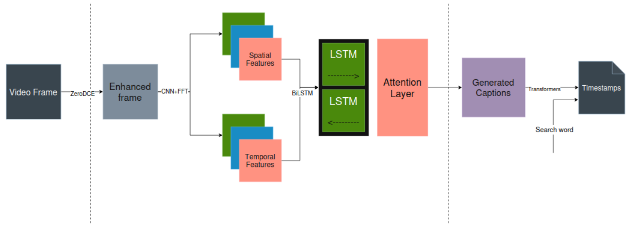


Fig. 1: Proposed CNN + FFT + BiLSTM Video Captioning System.

IV. Implementation

A. Dataset

The experiments in this work utilize the CCTV Action Recognition Dataset, which consists of short video clips specifically designed for human action recognition in surveillance footage. The dataset includes 13 distinct action categories, such as *Fall*, *Gun*, *Hit*, and *Struggle*, with 200 clips per category. Each video clip is approximately 3 to 4 seconds long and captures the target action clearly.

Videos are named using the convention `{Name_of_source}{Name_of_video_type}` and `{Action_Category}`, which allows for easy identification of the source, type, and action of each clip. Since the dataset was not formally captioned and clips are short, captions were generated automatically based on the action category name. These captions serve as the ground truth for training and evaluation of the frame captioning model.

The dataset also provides predefined training and testing splits to facilitate reproducibility and benchmarking. The structured naming and split conventions make it a practical resource for developing, evaluating, and comparing action recognition models in surveillance scenarios.

B. Feature Extraction: CNN + FFT for Hybrid Spatio-Temporal Representation

The feature extraction stage combines Convolutional Neural Networks (CNN) with Fast Fourier Transform (FFT) to create a hybrid spatio-temporal representation of each video. This approach allows the model to learn both the spatial context within frames and the temporal dynamics across consecutive frames.

Each video from the CCTV Action Recognition Dataset is divided into individual frames. These frames are resized and normalized before being passed into a pre-trained CNN backbone used for spatial feature extraction. The CNN encodes the spatial information such as shapes, movements, and object appearances relevant to recognizing the performed action. By leveraging transfer learning, the lower-level filters capture generic visual patterns, while the upper layers are fine-tuned to adapt to the specific CCTV domain.

To complement spatial understanding with temporal features, frame-wise CNN embeddings are processed using the Fast Fourier Transform (FFT). The FFT converts the time-domain sequence of feature activations into the frequency domain, emphasizing motion periodicity and temporal transitions. This transformation helps identify motion energy and intensity patterns over time, allowing the model to distinguish subtle differences in actions that might appear visually similar.

The output feature vectors from both CNN and FFT components are concatenated and normalized to produce a joint feature representation that captures both spatial and temporal information. As described in the methodology section (Eq. (2)–(3)), this hybrid encoding forms the input to subsequent modules such as the attention-based BiLSTM for frame captioning.

Overall, this feature extraction pipeline enables efficient representation learning from short video clips, ensuring that both static and dynamic cues of the human actions are preserved.

C. Frame Captioning: Attention-Based BiLSTM

After obtaining the hybrid spatio-temporal features by concatenating the spatial features from the CNN and the temporal features from the FFT, the next step involves generating captions for each video using an Attention-based Bidirectional Long Short-Term Memory (BiLSTM) network.

1) Feature Preparation: The extracted spatial and temporal features are first matched and aligned for each video. Temporal features, which may have a lower dimension than the number of frames, are expanded to match the spatial feature time steps. The concatenation produces a hybrid feature tensor for each video. The resulting hybrid features are then standardized to have consistent dimensions across the dataset and padded along the temporal dimension to accommodate variable video lengths. This ensures that all videos can be processed in batch form by the BiLSTM network.

2) BiLSTM Captioning Network: The BiLSTM network is used for sequence modeling to generate captions from the hybrid features. The network processes the concatenated feature vectors in both forward and backward temporal directions, allowing it to capture past and future context simultaneously. An Attention mechanism is incorporated into the decoder to focus on relevant frames while generating each word in the caption, improving the descriptive accuracy.

The overall process can be summarized as follows:

1) The hybrid features $X_{features}$ are fed into the BiLSTM encoder.
2) The decoder generates captions one word at a time, attending to important frames through the Attention mechanism.
3) The predicted sequence is compared to the ground truth caption $y_{captions}$ for training, using a suitable loss function such as categorical cross-entropy.

3) Training and Caption Generation: During training, the model learns to map the hybrid feature sequences to corresponding captions. During inference, the trained BiLSTM network generates captions for new videos based on their fused features. This enables the system to produce meaningful textual descriptions that reflect the actions observed in the input videos.

The combination of BiLSTM and Attention allows the system to effectively handle temporal dependencies in the video frames and selectively emphasize critical information, resulting in more accurate and contextually relevant captions.

D. Timestamp Forwarding: Semantic Search using Vector Database

The final stage of the system is the timestamp forwarding module, which allows users to navigate videos efficiently by jumping to relevant moments based on keywords or semantic queries. This module leverages the captions generated in the previous stages and performs both direct keyword search and semantic search using a vector database.

1) Keyword-Based Timestamp Extraction: Initially, the system supports a simple keyword search approach. Captions for each video are downloaded using `yt_dlp` and parsed in `WebVTT` format. The module searches through the text for the presence of a user-provided keyword and returns the timestamps corresponding to all occurrences. This enables users to quickly jump to specific events or actions mentioned in the captions.

2) Semantic Search with Sentence Embeddings: To provide more intelligent and context-aware navigation, the system uses a semantic search approach. Captions are split into small chunks, typically covering three consecutive lines, and each chunk is converted into embeddings using a pre-trained `SentenceTransformer` model. The user query is similarly encoded, and cosine similarity is computed between the query embedding and each caption chunk embedding.

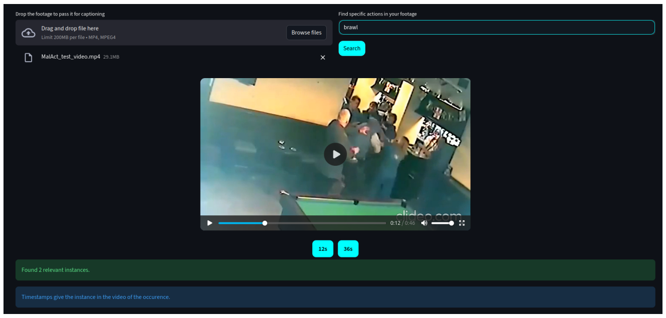
Chunks with similarity scores above a defined threshold are considered relevant, and their timestamps are converted to clickable links that direct the user to the corresponding point in the video. This approach allows for approximate matching and retrieval of semantically related content, even when the exact keyword is not present.

3) Implementation Details:

1) Caption Extraction: Captions are downloaded automatically from YouTube using `yt_dlp`, and stored temporarily in `.vtt` format.
2) Chunking and Embedding: Captions are grouped into consecutive chunks to maintain context, and embeddings are generated using the `all-MiniLM-L6-v2` model.
3) Similarity Computation: Cosine similarity between the query embedding and each chunk embedding determines relevant moments.
4) Timestamp Conversion: Caption start times are converted to total seconds, forming clickable video links for direct navigation.

While some captions may be imperfectly aligned with video events, the combination of keyword and semantic search provides a robust mechanism for timestamp forwarding. Any discrepancies or gaps in the captions are noted as potential areas for future improvement.

V. Results and Analysis


Drop the footage to pass it for captioning

Drag and drop file here

Limit 200MB per file • MP4, MPEG4

MalAct_test_video.mp4 29.1MB

Find specific actions in your footage

brawl

Found 2 relevant instances.

Timestamps give the instance in the video of the occurence.

Fig. 2: Web Application of the Product

A. Experimental Setup

The entire workflow was developed using the Google Colab environment. Python was adopted as the coding language. Libraries predominantly used include NumPy, Pandas, Tensorflow, Keras and OpenCV. Streamlit was used for the front-end, which was hosted on Pygrok. The services were separated based on the functionality of the methods, included under one of feature extraction, BiLSTM captioning and timestamping.

B. Analyzing Results of the Integrated Workflow

TABLE I: Metrics Computed

S. No	Metric	Value
1.	BiLSTM Accuracy	0.9854
2.	ZeroDCE total loss	1.6971
3.	IS Loss	0.1648
4.	Exposure Loss	1.4205
5.	CC Loss	0.0318
6.	SC Loss	0.08

This integrated workflow presents a very high accuracy on the test data (0.9854). Figure 3 helps us validate this as we can observe the decreasing loss over the training epochs. The Cumulative Explained Variance helps us validate our model more by indicating the variability in the videos/captions generated in the dataset.

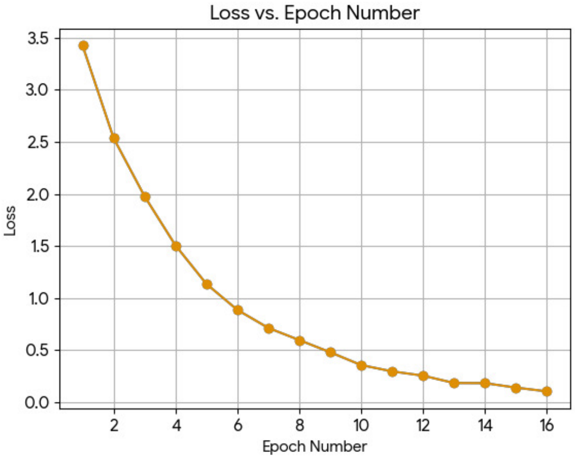


Fig. 3: Training Loss over 16 Epochs

The ZeroDCE model shows great enhancement metrics with a very low total loss (1.6971) over metrics Illumination Smoothness loss, Color Constancy loss, Spatial Constancy loss and Exposure loss.

VI. Conclusion and Future Works

A. Conclusion

In this work, we are working on a hybrid video captioning system that integrates spatial and temporal feature extraction, attention-based BiLSTM captioning, and a semantic timestamp forwarding mechanism. The system successfully shows

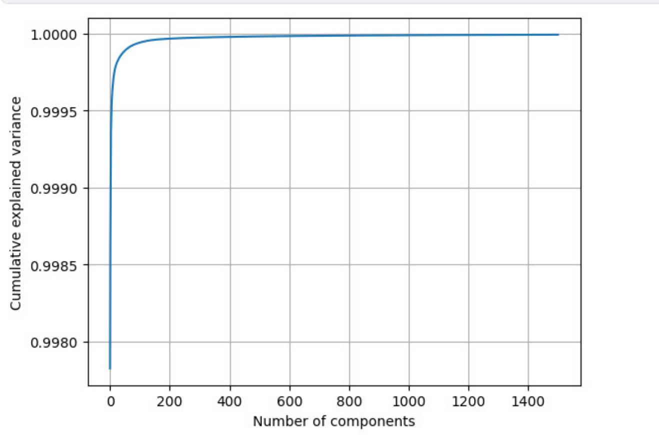


Fig. 4: Cumulative Explained Variance

the ability to generate captions for short CCTV videos and allows keyword- or semantically-driven navigation to relevant moments.

Despite these successes, several limitations were observed. The autoencoder-decoder used in the frame captioning module tends to generate captions with limited vocabulary, reducing the diversity and expressiveness of generated sentences. Additionally, the spatial-temporal feature extraction is dependent on the quality of pre-trained CNN features and FFT-based temporal patterns, which may not fully capture complex motions in certain video clips. In the timestamp forwarding module, while semantic search improves context-based navigation, the captions generated by the BiLSTM may not always align perfectly with the actual video events, leading to occasional discrepancies in retrieved timestamps.

B. Future Works

Future improvements can focus on the following areas:

1) Enhanced Caption Diversity: Incorporating richer language models or using transformer-based decoders could increase vocabulary and generate more descriptive and varied captions.
2) Improved Temporal Feature Modeling: Exploring advanced temporal feature extraction methods, such as 3D CNNs or optical flow-based representations, may capture motion patterns more effectively than FFT-based methods.
3) Tighter Integration of Subtasks: Currently, the semantic timestamp forwarding relies on captions from the BiLSTM but does not influence caption generation itself. A future version could implement a feedback loop where user queries or timestamp matches refine the captioning model in real time.
4) End-to-End Learning: Developing a fully end-to-end trainable system that jointly optimizes spatial-temporal features, caption generation, and semantic timestamp retrieval could enhance overall performance and reduce misalignments between video content and captions.
5) Dataset Expansion and Augmentation: Increasing the variety and length of video clips, as well as providing multiple human-generated captions per video, can help the model learn richer representations and generate more accurate captions.
6) Robustness and Error Handling: Enhancing the timestamp forwarding module to handle missing captions or poorly aligned text, possibly using multimodal cues, would improve user experience during video navigation.

Overall, the proposed system demonstrates the feasibility of integrating spatial-temporal feature extraction, attention-based captioning, and semantic video navigation. Future enhancements targeting vocabulary richness, temporal modeling, and tighter subsystem integration are expected to significantly improve performance and usability.

References

[1] K. Simonyan and A. Zisserman, “Two-Stream Convolutional Networks for Action Recognition in Videos,” in *Advances in Neural Information Processing Systems*, 2014.

[2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] J. Donahue et al., “Long-term Recurrent Convolutional Networks for Visual Recognition and Description,” in *Proc. IEEE CVPR*, 2015.

[4] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, “Sequence to Sequence – Video to Text,” in *Proc. IEEE ICCV*, 2015.

[5] K. Xu et al., “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention,” in *Proc. ICML*, 2015.

[6] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui, “Jointly Modeling Embedding and Translation to Bridge Video and Language,” in *Proc. IEEE CVPR*, 2016.

[7] Y. Wang et al., “Hybrid CNN–RNN Model for Video Captioning,” in *Proc. IEEE Transactions on Multimedia*, vol. 21, no. 2, 2019.

[8] J. Zhang, S. Xu, and Z. Zhang, “Fourier Transform-Based Motion Encoding for Action Recognition,” in *Proc. IEEE ICIP*, 2020.

[9] Z. Yang, Y. Yuan, Y. Wu, and Q. Chen, “Attention Enhanced BiLSTM for Video Caption Generation,” in *Proc. IEEE CVPR Workshops*, 2021.

[10] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” in *Proc. EMNLP*, 2019.