Dense Video Captioning

Ashwani Kumar
Information Technology
National Institute of Technology, Karnataka
Surathkal, India
ashwanikumar.211it013@nitk.edu.in

Arvind Prabhu
Information Technology
National Institute of Technology, Karnataka
Surathkal, India
arvindprabhu.211it010@nitk.edu.in

Jaheer Khan
Information Technology
National Institute of Technology, Karnataka
Surathkal, India
jaheerkhan.211it026@nitk.edu.in

***Abstract*—Dense video captioning aims to identify events in untrimmed videos and generate descriptive captions for each event. While multi-modal data plays a key role in enhancing task performance, traditional approaches focus on visual or dual audio-visual inputs, often neglecting textual input, such as subtitles. Given that textual data aligns closely with the structure of video caption words, incorporating text modalities can significantly boost captioning accuracy. In this paper, we introduce a novel framework, termed the Multi-Stage Fusion Transformer Network (MS-FTN), designed for multi-modal dense video captioning through the staged integration of text, audio, and visual features. The proposed multi-stage feature fusion encoder first combines audio and visual features at lower levels, subsequently integrating these with a globally shared textual representation at a higher level to produce multi-modal, complementary context features. Additionally, an anchor-free event proposal module is implemented to generate event proposals efficiently, circumventing the need for complex anchor calculations. Comprehensive evaluations on subsets of the ActivityNet Captions dataset highlight that MS-FTN achieves notable improvements in performance and computational efficiency. Furthermore, ablation studies confirm that the globally shared text representation is especially effective for multi-modal dense video captioning.**

Keywords: ***Anchor-free event proposal, dense video captioning, multi-modal fusion, multi-stage fusion transformer, text-audio-visual integration.***

I. INTRODUCTION

Dense video captioning is an field in video captioning that focus on identifying and creating detailed captions for specific events in a video . In traditional video captioning addresses shorter and trimmed clips with clear visual and audio cues , dense video captioning requires analyzing lengthy, continuous footage and detecting numerous events. Previous approaches based on visual and audio-visual inputs alone but research suggests that including textual data like subtitles and other textual information can greatly improve the quality and accuracy of generated captions. Textual data often aligns closely with the natural language structure used in captions and provide valuable context that is else absent in visual or audio inputs alone.

It is a novel framework that is called Multi-Stage Fusion Transformer Network (MS-FTN) that integrates text, audio, and visual features for more effective multi-modal dense video captioning. MS-FTN utilizes a multi-stage feature fusion encoder, which begins by combining audio and visual inputs at early stages, and later introduces a shared textual representation, fusing these complementary inputs to create context aware captions. MS-FTN includes an anchor-free event proposal module that generates event proposals without relying on fixed anchor points, which increases both accuracy and efficiency.

Evaluations on the ActivityNet Captions dataset reveal that MS-FTN improves both performance and computational efficiency compared to existing methods. Ablation studies further highlight the crucial role of globally shared textual representation, confirming its potential to boost multi modal dense video captioning accuracy. This multi stage and multi modal approach not only addresses limitations in traditional models but also sets a foundation for more nuanced and better automated video understanding in diverse applications.

II. LITERATURE SURVEY

Video captioning techniques that is emphasising deep learning based methods. It explores three primary approaches: template based, retrieval based, and deep learning methods. With a focus on encoder-decoder frameworks and attention mechanisms for generating natural language descriptions from video content. The survey also discusses various datasets like MSVD, MSR-VTT, and MPII-MD used for training video captioning models and presents popular evaluation metrics. This highlights the advancements in video captioning but also notes the challenges, like creating accurate, context aware descriptions, pointing towards future research directions to improve these models further. [1]

An video captioning model called a LSTMs. That integrates attention mechanisms with LSTM networks to generate natural language descriptions of videos. Existing methods that compress video frames into static representations, this model dynamically selects salient features using an attention mechanism and ensures semantic consistency between visual content and generated sentence. The framework utilises a two-dimensional Convolutional Neural Network for spatial features, an LSTM encoder for temporal sequences, and a multi modal embedding space for aligning video and language semantics. Experimental results on benchmark datasets demonstrate that aLSTMs achieve competitive or superior performance compared to state-of-the-art methods, as measured by BLEU and METEOR scores. This model effectively captures the complex temporal and semantic structures required for accurate video captioning.[2]

An innovative approach to video captioning by operating directly in the compressed video domain. Traditional methods involve multiple steps: decoding frames, extracting features, and then generating captions, which can be inefficient and prone to redundant information. The proposed method simplifies this process by leveraging compressed video data—consisting of I-frames, motion vectors, and residuals—to train an end-to-end transformer model. This approach eliminates the need for manual frame sampling and offline feature extraction, significantly improving both efficiency and speed. Experimental results show that this method achieves state-of-the-art performance while being nearly twice as fast as existing techniques. This approach highlights the benefits of processing compressed video directly, resulting in faster and more efficient video captioning.[3]

A new evaluation metric for image captioning called Positive-Augmented Contrastive learning Score (PAC-S), which combines contrastive visual-semantic learning with both real and synthetically generated data. The PAC-S metric improves upon existing methods by integrating positive samples from generated images and text, alongside curated data, to enhance the alignment with human judgement. Extensive experiments on various datasets, including images and videos, show that PAC-S surpasses traditional metrics like CIDEr and SPICE, as well as reference-free metrics such as CLIP-Score, in terms of correlation with human evaluations. The proposed metric also exhibits better performance in identifying object hallucinations, demonstrating its effectiveness across different evaluation settings. PAC-S provides a promising advancement in automatic image and video caption evaluation. [4]

This explores an encoder-decoder model for video captioning, transforming sequences of video frames into cohesive text captions. By employing a many-to-many mapping approach, the model maps temporal sequences of frames to word sequences, creating captions that describe entire videos rather than individual frames. Key processes include data preprocessing, model construction, and training, with performance measured through 2-gram BLEU scores across different dataset splits. This evaluation demonstrates the model's ability to generalize across temporal dimensions, handling scene transitions and action variations. Video captioning is notably more complex than image captioning due to the temporal nature of video and variability across frames, such as changes in brightness, camera angles, and actions. The model architecture leverages Long Short-Term Memory (LSTM) networks to handle sequential data, using a pre-trained 2D CNN to extract frame-level features, which the LSTM encoder processes to capture temporal sequences. The aim is to generate a coherent caption that summarizes the entire video, with architectural adjustments enhancing grammatical accuracy and overall caption coherence. [5]

This provides an in-depth review of deep learning methods used for Video Captioning (VC), along with a discussion of the datasets and evaluation metrics (BLEU, ROUGE, METEOR) commonly used in this field. After the review, the study compares how well these VC methods perform using different datasets and evaluation metrics. The survey also explores how VC can be applied to other areas, such as video tagging, content-based image retrieval, video recommendation systems, and assistance for visually impaired individuals. Additionally, the paper points out some areas where more research is needed, making it a helpful resource for researchers new to VC and looking to contribute to this rapidly developing field.[6]

This gives a detailed look at deep learning techniques used for video captioning, covering the basics, important datasets, evaluation methods, different approaches, and the challenges involved. The study explains the key parts of video captioning, like how videos are processed (encoder), how captions are generated (decoder), and the role of word embeddings. It also discusses popular datasets such as MSR-VTT, M-VAD, MSVD, and others, which are essential for training and testing video captioning models. Different methods used in video captioning are reviewed, including those based on attention mechanisms, reinforcement learning, graph structures, generative adversarial networks, and multi-modal approaches, each offering unique solutions to the challenges of generating accurate captions. The survey also highlights the difficulties in video captioning, such as handling the complexity of video data and applying deep learning techniques effectively. Overall, this survey provides valuable insights for researchers interested in deep learning-based video captioning. [7]

This proposed a novel approach using a multimodal attention-based transformer for dense video captioning. The RGB and flow features are processed separately in the audio-visual attention block of the encoder. The hierarchical attention block in the caption generation module uses semantic features to generate the descriptions for the events. The results show that the proposed model provides better evaluation metrics for the generated proposals than other state-of-the-art approaches.[8]

This provides a comprehensive overview of Dense Video Captioning (DVC) techniques, focusing on methods developed since the introduction of DVC in the ActivityNet challenge. Organised around the typical DVC pipeline—Video Feature Extraction, Temporal Event Localization, and Dense Caption Generation—the survey highlights the reuse of top-performing methods and transformers. It summarises key evaluation metrics and discusses datasets and challenges in DVC, noting a significant rise in research interest from 2018 to 2023. The survey emphasises the need for better integration of event detection and captioning, pointing out the common use of C3D, VGGish, and I3D for feature extraction, and the popularity of transformer decoders for caption generation. As the field evolves, future DVC research will need to address challenges like integrating with the medical field, improving accuracy, and developing better pre-trained models and evaluation criteria.[9]

This introduced a novel approach to dense video captioning, inspired by how humans understand scenes. By using cross-modal retrieval from an external memory (CM2), we significantly improved both event localization and caption generation. Our experiments on the ActivityNet Captions and YouCook2 datasets confirmed the effectiveness of this memory

retrieval method. Remarkably, CM2 delivered strong results without requiring pre-training on large video datasets, showcasing its efficiency. We believe this work paves the way for future research in dense video captioning and encourages the exploration of memory-augmented models to enhance video understanding and captioning. [10]

III. Proposed Methodology

In this project, our primary goal was to dense video captioning . We approached this challenge by breaking down our methodology into several key steps: dataset, , Feature Extraction Process, Caption Data Preprocessing, Model Architecture, Data Loading Strategy, Training Process, Inference Model, Caption Generation Workflow . Below is a detailed description of each step.

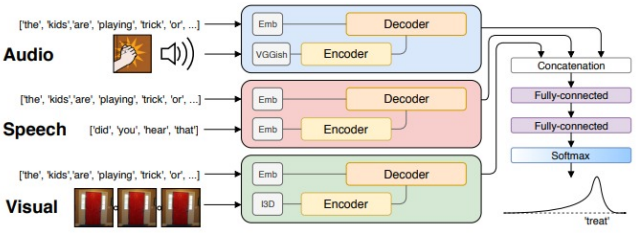

Figure 1. Architecture of the Project.

A. Dataset Overview

The dataset is organized into separate folders for training and testing data, each containing a similar structure to facilitate efficient model training and evaluation. Within each folder, there is a video sub folder holding the actual video files designated for either training or testing. Additionally, a feat sub folder contains preextracted features of the video frames, allowing the model to access frame-specific data without recalculating these features during each training cycle. To map video IDs to their corresponding captions, each folder includes a JSON file either training_label.json or testing_label.json. Which provides a structured reference for associating video content with descriptive text. This organization streamlines data access, ensuring that models can readily access both raw video files and precomputed features as they progress through training and evaluation.

B. Feature Extraction Process

Videos are treated as sequences of frames for feature extraction. Each video is broken down into frames, which can be considered as still images that represent different time points.

1) Uniform Frame Selection: To maintain consistency across videos of varying lengths, only 80 frames are sampled from each video, ensuring a fixed input size.

2) Feature Extraction Method: A pre-trained VGG16 convolutional neural network is employed to extract 4096-dimensional feature vectors from each video frame, chosen for its robust ability to capture essential visual information. The features from all frames in a video are then stacked into a matrix with a shape of (80, 4096), where 80 corresponds to the number of frames and 4096 represents the dimensionality of each frame's feature vector. Given that feature extraction is computationally intensive, the dataset provides these features as pre-extracted numpy arrays. This setup allows the project to focus on the core task of captioning model development without the need to repeatedly perform feature extraction.

C. Caption Data Preprocessing

1) Caption Loading and Pairing: Captions from the JSON files are loaded and paired with their respective video IDs to create training pairs.

2) Tokenization and Padding: Each caption is tokenized by splitting sentences into individual words and converting them into numerical tokens. To standardize input sizes, captions are padded to a uniform length of 10 words; captions longer than 10 words are truncated, while shorter ones are padded with zero values. Additionally, a beginning-of-sentence token is prepended to each caption to signal the model to start generating a sentence, and an end-of-sentence token is appended to indicate when the model should stop predicting.

3) Filtering Captions: Captions with word counts outside the range of 6-10 are excluded. This decision helps minimize excessive padding, which can lead to poor model performance due to the generation of padded tokens.

4) Vocabulary Limitation: The vocabulary is limited to the 1500 most frequent words in the training set. Rare words are excluded to prevent overfitting and reduce the complexity of the model.

D. Model Architecture

We employs an encoder-decoder architecture, a sequence-to-sequence model suited for generating text from sequential data like video features. In this setup, the encoder processes an 80-frame feature matrix using LSTM cells, sequentially extracting hidden states from each frame. The final hidden state of the encoder is preserved to initialize the decoder. The decoder, also based on LSTM cells, generates captions by starting with an initial input token and then iteratively feeding in the previously generated word in the sequence. It continues generating one word at a time until reaching an end-of-sequence token, signaling the completion of the caption.

To ensure consistency with the input data, the encoder processes 80 time steps corresponding to each video frame, while the decoder operates over 10 time steps, matching the padded caption length. Within the decoder, an embedding layer maps token indices to dense vectors, and a dense output layer projects LSTM outputs to a probability distribution across a 1500-word vocabulary, facilitating the word prediction process.

E. Data Loading Strategy

Given the dataset size of approximately 14,000 data points, loading all data at once can easily overwhelm system memory. To address this, a custom data generator is implemented in Python to manage data in manageable batches, each with a batch size of 320. This generator loads video features and their corresponding tokenized captions in chunks, enabling efficient data processing. It returns two primary inputs for model training: the encoder input (video features) and the decoder input (captions). Additionally, captions are converted into categorical data to align with the 1500-token vocabulary. To further optimize the process, video features are stored in dictionaries for quick and easy lookup, minimizing the need to repeatedly load the same data from disk and thus enhancing computational efficiency.

F. Training Process

The model is trained for a total of 150 epochs using the free version of Google Colab, which provides access to a Tesla T4 GPU, ensuring a relatively efficient training process. Each epoch takes approximately 40 seconds, allowing for an effective and streamlined training cycle. Categorical cross-entropy is used as the loss function, making it well-suited for multi-class classification tasks, while the Adam optimizer is employed to facilitate adaptive learning and optimize performance across epochs.

G. Inference Model

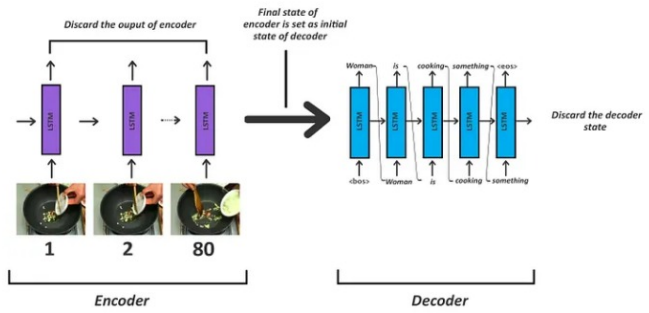

Figure 2. Inference Model

The inference process differs from the training phase, as the encoder and decoder are loaded separately for generation. The pre-trained encoder receives the 80-frame feature matrix and outputs its final hidden state, which serves as the initial state for the decoder. The decoder is then initialized with this final state, along with a starting token to prompt caption generation. For real-time inference, a greedy search approach is used: at each step, the model selects the most probable word and feeds it into the next LSTM cell, continuing until it predicts an end-of-sequence token or reaches the maximum caption length.

H. Caption Generation Workflow

The decoder iteratively predicts words to form a complete caption. At each step, it takes in the current state and the previously generated word, passing them as input to the decoder's LSTM. The LSTM then outputs the token for the next word, which is converted back into a word using a reverse lookup in the vocabulary. This process continues, with each predicted word feeding into the next step, until an end-of-sequence token is generated, indicating the completion of the caption.

IV. Results and Analysis

We wanted to understand how news sentiment is related to stock prices for companies listed in the Nifty 50 index. To do this, we used sentiment analysis based on the FinBERT model and extracted financial data from Yahoo Finance, with an aim to understand the impact of public sentiment reflected in news articles on stock price movements.

A. Feature Extraction and Representation

The VGG16 model was pretrained on the ImageNet dataset and used to extract high-level visual features from each frame in the video. These features capture essential objects and scenes, allowing the system to create a comprehensive representation of visual content. During the feature extraction stage, VGG16 provided rich feature maps, effectively identifying key objects and relationships in each frame. This preprocessing step enabled the LSTM model to focus on temporal dependencies and sequential patterns, improving caption generation.

Features are extracted from the first 80 frames of each video. VGG16 represents each set of features in the form of a vector of length 4096. The set of features for each vector are stacked to foem an array of shape (80, 4096).

B. Caption Generation

The LSTM encoder-decoder model was trained to capture temporal relationships in the video frames and generate captions describing the scene accurately.

C. Training

The words in the vocabulary are represented using one-hot encoding. Categorical cross entropy was used to calculate loss while training. The loss values are averaged over a batch to obtain a final score for that batch.

Accuracy was chosen as the metric used to evaluate the model. It measured the number of tokens correctly predicted at each time step in the sequence. At each step, the token predicted by the model was compared with the true token from the target sequence. Early stopping was used to stop the model from training if there was no improvement in the validation loss for four epochs.

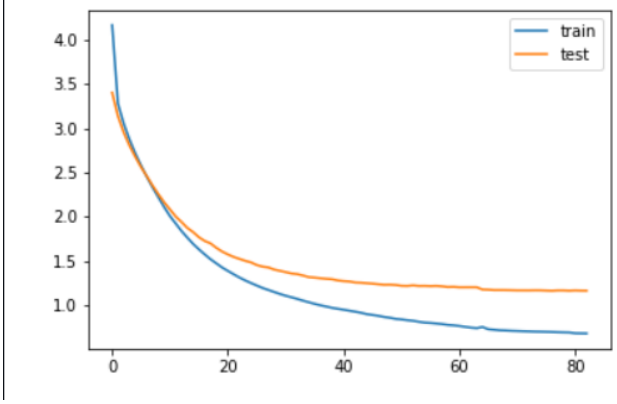

Figure 3. Loss Graph

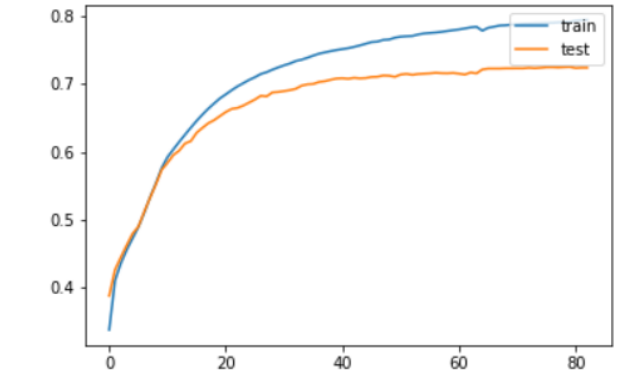

Figure 4. Accuracy Graph

D. Qualitative analysis

To illustrate the system's performance, we present several sample frames with their corresponding captions generated by the model. These examples show the system's ability to generate contextually accurate captions that reflect both the actions and objects present in each scene. For instance:

- Figure 5 shows the output of the model when given a sequence showing a person adding food to a pan. The model generated the caption, "Man is adding something" which accurately reflects the visual content.

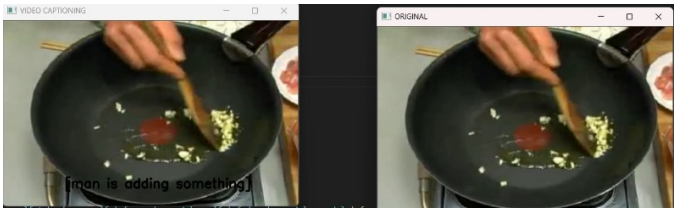

Figure 5. Adding something to pan

- Figure 6 shows the output of the model when given a sequence showing a person pouring rice into a pot. The model generated the caption, "Person is pouring water into a bowl" which does not accurately reflect the visual content.

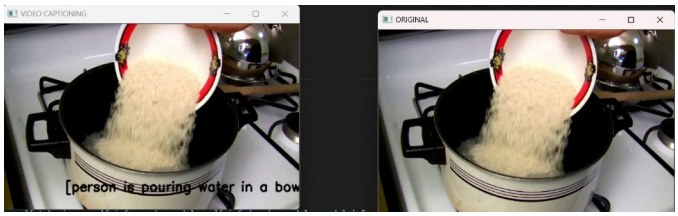

Figure 6. Adding rice to pot

E. Limitations and Error Analysis

Due to limited representation of certain activities such as sports, the model faces difficulty in recognizing those activities. The LSTM architecture also struggles with capturing long term dependencies in videos, as shown in figures 7 and 8. The video they were taken from shows a human face changing multiple characteristics over time, such as age, gender, skin tone, expression, ethnicity, etc.

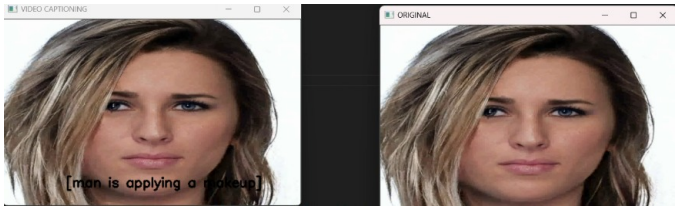

Figure 7. Timestamp: 0:00:01

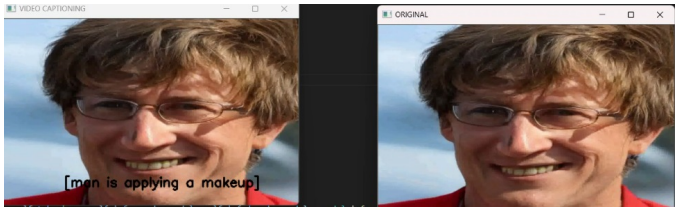

Figure 8. Timestamp: 0:00:04

BLEU-4 (Bilingual Evaluation Understudy) and METEOR (Metric for Evaluation of Translation with Explicit ORdering) are key evaluation metrics widely used in natural language processing to assess the quality of machine translation and text generation. BLEU-4 aims to measure the similarity between a generated sentence and one or more reference sentences. It calculates n-gram precision by comparing n-grams (sequences of n words) in the generated sentence with those in the reference sentences. BLEU-4 specifically focuses on 4-grams

(sequences of 4 words), using the geometric mean of 1-gram to 4-gram precisions. To discourage overly short sentences from scoring high merely due to matching a subset of the reference words, BLEU incorporates a brevity penalty. Its scores range from 0 to 1 (or 0 to 100 as a percentage), with higher scores indicating closer alignment to reference sentences.

Metric	Base Paper	Our findings
METEOR	0.096	0.043
BLEU-4	0.184	0.151

Table I
COMPARISON OF METEOR AND BLEU SCORES BETWEEN BASE PAPER AND OUR APPROACH

METEOR seeks to enhance BLEU by factoring in more nuanced text similarities, such as synonymy, stemming, and paraphrasing, making it more sensitive to variations in word choice. METEOR measures unigram recall, matching individual words, but extends beyond exact matches by incorporating synonyms and stemming (considering word inflections and synonyms), order (penalizing incorrect word sequences but allowing some flexibility), and paraphrase matching (accounting for variations in wording that convey similar meanings). Like BLEU, METEOR scores also range from 0 to 1, where higher scores suggest greater similarity to the reference sentence. In summary, while BLEU remains popular for its simplicity and efficiency, METEOR is often favored for capturing semantic similarity and accommodating a wider range of linguistic expressions.

V. CONCLUSION AND FUTURE WORK

The dense multimodal video captioning project successfully implements an encoder-decoder architecture to generate meaningful captions for video content by leveraging deep learning techniques. By processing video frames with pre-trained VGG16 and employing LSTM-based sequential models for both encoding and decoding, the system is capable of understanding visual and temporal patterns and generating concise, coherent descriptions. The results demonstrate that with appropriate data preprocessing, optimized training, and efficient model structures, video content can be effectively annotated, providing significant advancements in video comprehension and accessibility.

This approach has direct implications for improving search algorithms, enhancing recommendation systems, and enabling automated tagging for video libraries, thereby making video-based data more manageable and valuable. Despite challenges such as handling varying video lengths and large vocabulary sets, the methodology has proven robust with the use of padding and vocabulary limitations.

The scope for future enhancements in the video captioning domain is extensive. Key areas for improvement and further exploration include:

- Advanced Captioning Techniques: Integrating transformer-based architectures such as the Vision Transformer (ViT) or pre-trained models like GPT-4 Vision for more sophisticated video understanding and caption generation. Implementing beam search or other advanced decoding strategies to improve the quality of generated captions.
- Multi-language Support: Expanding the system to support multiple languages for captioning to make the model accessible to a global audience. Integrating natural language processing (NLP) models capable of multilingual caption generation.
- Contextual and Sentiment Analysis: Enhancing the model by adding sentiment analysis layers to provide richer, context-aware descriptions. Leveraging video metadata and audio analysis for more comprehensive captioning.
- Improved Data Augmentation: Using data augmentation techniques to handle limited datasets better, ensuring more diverse training scenarios. Incorporating synthetic data to expand the dataset for improved training coverage.
- Real-time Captioning: Optimizing the model for real-time applications, which could prove invaluable in live broadcasting and assistive technology for the hearing impaired. Exploring lightweight model architectures or model quantization for deployment on edge devices.
- Cross-modal Learning: Implementing cross-modal learning approaches that merge video, audio, and text data for even more nuanced captions. Enhancing the integration of audio cues (e.g., speech, sound effects) to complement visual features for a comprehensive video captioning system.
- Scalability and Deployment: Improving the model's scalability and transitioning from research to production environments, including deployment on cloud-based platforms or mobile applications. Developing user-friendly APIs for integrating the captioning system into existing platforms for easy adoption by content creators and media companies.

These future directions can lead to a more robust, adaptable, and widely applicable video captioning system, meeting the growing demand for intelligent, automated video analysis across different industries and user groups.

REFERENCES

[1] M. Amaresh and S. Chitrakala. Video captioning using deep learning: An overview of methods, datasets and metrics. In *2019 International Conference on Communication and Signal Processing (ICCSP)*, pages 0656–0661, 2019.

[2] Lianli Gao, Zhao Guo, Hanwang Zhang, Xing Xu, and Heng Tao Shen. Video captioning with attention-based lstm and semantic consistency. *IEEE Transactions on Multimedia*, 19(9):2045–2055, 2017.

[3] Yaojie Shen, Xin Gu, Kai Xu, Heng Fan, Longyin Wen, and Libo Zhang. Accurate and fast compressed video captioning, 2024.

[4] Sara Sarto, Manuele Barraco, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Positive-augmented contrastive learning for image and video captioning evaluation, 2023.

[5] Sikiru Adewale, Tosin Ige, and Bolanle Hafiz Matti. Encoder-decoder based long short-term memory (lstm) model for video captioning, 2023.

[6] Moloud Abdar, Meenakshi Kollati, Swaraja Kuraparthi, Farhad Pourpanah, Daniel McDuff, Mohammad Ghavamzadeh, Shuicheng Yan, Abduallah Mohamed, Abbas Khosravi, Erik Cambria, and Fatih Porikli. A review of deep learning for video captioning, 2023.

[7] Adel Jalal Yousif and Mohammed H. Al-Jammas. Exploring deep learning approaches for video captioning: A comprehensive review. *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, 6:100372, 2023.

[8] Hemalatha Munusamy and Chandra Sekhar C. Multi-modal hierarchical attention-based dense video captioning. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 475–479, 2023.

[9] Iqra Qasim, Alexander Horsch, and Dilip K. Prasad. Dense video captioning: A survey of techniques, datasets and evaluation protocols, 2023.

[10] Minkuk Kim, Hyeon Bae Kim, Jinyoung Moon, Jinwoo Choi, and Seong Tae Kim. Do you remember? dense video captioning with cross-modal memory retrieval, 2024.