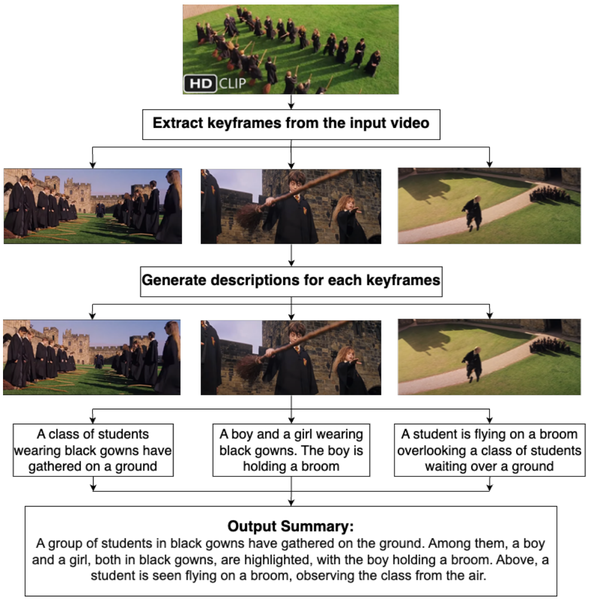
# Video Description and Semantic Retrieval via Keyframe Stitching

**Adwait Mahadar** mahadar@usc.edu
**Arpita Sahu** arpitasa@usc.edu
**Jeffrey Guo** jxguo@usc.edu
**Karthik Mayya** kmayya@usc.edu
**Shubham Rajrah** rajrah@usc.edu

## Abstract

Video description generation has been an important tool in modern day society, allowing information extraction from videos as well as providing a way for those with disabilities to access video content. Traditional approaches rely heavily on video transcripts and metadata to generate descriptive captions. In this work, we propose a novel methodology for dense video description generation that focuses solely on visual data, bypassing the need for textual transcripts. Using the information captured from keyframes, we generate coherent and temporally-aligned dense captions for a given video. Furthermore, we introduce a semantic retrieval mechanism that leverages the generated captions for efficient video search. Our proposed pipeline for video caption generation emphasizes rich and coherent textual summaries, but also facilitates efficient video processing and retrieval, making it a scalable solution for large video datasets.


HD CLIP
Extract keyframes from the input video
Generate descriptions for each keyframes

A class of students wearing black gowns have gathered on a ground

A boy and a girl wearing black gowns. The boy is holding a broom

A student is flying on a broom overlooking a class of students waiting over a ground

**Output Summary:**
A group of students in black gowns have gathered on the ground. Among them, a boy and a girl, both in black gowns, are highlighted, with the boy holding a broom. Above, a student is seen flying on a broom, observing the class from the air.

Figure 1: End-to-end sample of our system

## 1 Introduction

With the rapid growth of video content across various domains, there is an increasing need for automated tools to understand and describe video data effectively. Video description generation plays an important role in extracting meaningful information in a video content while providing improved accessibility for individuals with visual impairments. By transforming video data into concise, yet descriptive, textual summaries, these systems enable improved understanding and retrieval of information, particularly within large-scale datasets. As we are bridging the domains of vision and language, this can be classified as a video-to-text (VTT) problem (Perez-Martin et al., 2021).

Traditional video processing systems often rely on video transcripts, audio cues, and metadata in summarization (Otani et al., 2016). While effective, these methods face limitations in scenarios where such inputs are unavailable or incomplete. They also typically analyze a majority of frames in a video, which results in significant computational overhead as well as large memory consumption. This creates a need for efficient video analysis in caption generation tasks, which are capable of extracting visual content while minimizing redundancy.

Additionally, video captioning has centered around producing high-level, single-sentence descriptions for a given video. They often miss one key aspect – detail. Dense video captioning – describing events in the video with descriptive natural language – is gaining popularity in research as it enables video content to be better understood (Zhou et al., 2018a).

## 2 Hypothesis

To address the challenges mentioned above, we propose a methodology for dense video description generation that operates exclusively on visual data. Our approach leverages keyframe extraction to identify and summarize the most significant visual information from videos. By generating detailed, coherent, and temporally-aligned captions for these selected frames, we produce rich natural language summaries that effectively describe the contents of the video. We also introduce a semantic retrieval mechanism that utilizes these generated descriptions for video semantic search and retrieval tasks.

## 3 Related Works

### 3.1 Video Summarization and Captioning

Video summarization is a widely explored area in research, with efforts directed towards both video-to-video summarization as well as multi-modal video-to-text summarization. The latter, commonly known as video captioning, initially focused on producing one-sentence descriptions for localized events in a video (Xu et al., 2016; Papalampidi and Lapata, 2022). These methods typically combine convolutional neural networks (CNNs) for visual feature extraction with simple language models, such as LSTMs or RNNs, to generate text (Donahue et al., 2016).

In recent times, dense video captioning has gained popularity, where multiple events are detected and described in a single video rather than just the most significant one. Many researchers use temporal action proposal methods to localize sequences containing specific events or actions of interest, followed by a language model for generation (Krishna et al., 2017a). A well-known language model for dense video captioning is a self-attention transformer, which can effectively capture long-range dependencies (Zhou et al., 2018b). Additionally, Hu et al. (2023) propose a dual video summarization framework that integrates video summarization and captioning tasks to enhance video frame representation, demonstrating improved performance in generating accurate, descriptive captions (Hu et al., 2023)

### 3.2 Video Semantic Search

Existing methods for video-text retrieval can be broadly categorized into global and fine-grained approaches. Global video-sentence interaction methods align entire videos with sentences in a common feature space using separate text and video encoders. Models like ClipBERT (Lei et al., 2021) and CLIP4Clip (Luo et al., 2021) leverage pre-trained image-text models to efficiently map videos and sentences but fail to capture fine-grained relationships between individual video frames and words.

To address this, frame-word interaction methods focus on detailed alignment by comparing video frames and textual tokens. For instance, FILIP (Yao et al., 2021) uses token-wise maximum similarity for fine-grained alignment, while DRL (Wang et al., 2022) reduces feature redundancy with weighted token-wise interactions. However, these methods still lack a hierarchical understanding of video and text data, such as clip-phrase relationships. The HCMI (Jiang et al., 2022) approach bridges this gap by exploring multi-level interactions (video-sentence, clip-phrase, and frame-word) to achieve comprehensive video-text alignment.

Unlike previous methods that often rely on textual transcripts or metadata, our work generates dense and coherent video descriptions solely from visual data extracted from keyframes, eliminating dependency on external text. Additionally, we introduce a semantic retrieval mechanism that uses these descriptions for efficient video and keyframe search, providing a scalable end-to-end solution for large video datasets.

## 4 Methodology

### 4.1 Pipeline

Our proposed pipeline for this project consists of four main components:

1. **Keyframe Extractor**: Finds the most informative frames
2. **Caption generator**: Generates natural language captions for each selected frame
3. **Text Summarizer**: Stitches together the frame-level descriptions to produce the overall video text description
4. **Semantic Video Retrieval System**: Given a user query, returns relevant videos

Given a video as input, our system determines the most significant frames, i.e. keyframes, and generates a frame-level description for them. These captions are then passed through a text summarizer which stitches them together to produce a succinct, yet detailed, coherent natural language summary for the video. We can then take in a user search query, transform it into a vector, and utilize the

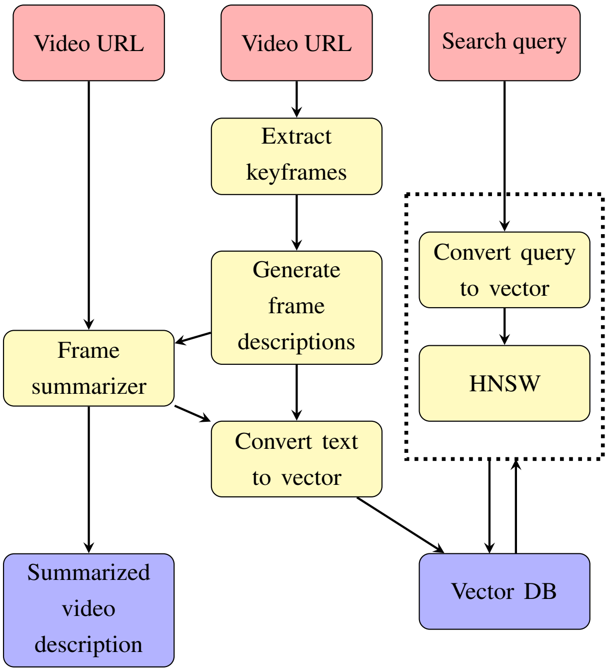


Figure 2: Video description & semantic search pipeline

Hierarchical Navigable Small World (HNSW) algorithm for rapid video retrieval from the video database that has created from training.

We have focused primarily on the natural language processing steps in the pipeline, i.e., the frame-level text description generation and the caption stitching, performing numerous experiments at both steps to find the techniques that produce the best results overall.

## 4.2 Datasets

### 4.2.1 VideoXum

The primary dataset we are using is VideoXum (Lin et al., 2024), a large-scale dataset for cross-modal video summarization. This dataset contains 14K long-duration videos, each associated with 10 text summaries. Of these 14K videos, 8K are used for training, 2K are used for validation and 4K are used for testing. This split is done in a manner such that the video length distribution is maintained. The VideoXum dataset is built on top of ActivityNet Captions (Krishna et al., 2017b), consisting of videos belonging to 200 distinct activity categories.

### 4.2.2 Wiki Movie Plots with Summaries

We used the Wiki Movie Plots with Summaries dataset (Priyavr, 2023) for finetuning in our text summarizer component. This dataset contains movie information for 35k movies such as release year, title, origin, directory, cast, genre, wiki page, plot, and plot summary.

### 4.2.3 Synthetic Dataset

We created a dataset to finetune our text summarizer component. This dataset contains 1000 videos, dense image captions for the videos, and an associated summary for the video taken from the VideoXum dataset.

## 4.3 Data Preparation

### 4.3.1 Data Cleaning

The VideoXUM dataset consists of the following features: video ID, duration, number of sampled frames, timestamps, text summary and visual frame summary. These video IDs correspond to YouTube videos, so we create a mapping between the ID's and their associated URLs. We determine the videos which are available with valid titles from the dataset with and use these in training and evaluating our system.

### 4.3.2 Keyframe Extraction

Keyframe extraction serves as an important data preparation step in our pipeline, enabling efficient processing and analysis of video content. Keyframe extraction focuses on identifying the most representative frames, which when put together can provide a holistic understanding of the full video.

We have experimented with three approaches to identify keyframes – Katna, Video-kf, and optical flow analysis. First, Katna (KeplerLab, 2019), performs K-means clustering on image histograms to tag video frames. It then selects the best image (with the least blur and highest Laplacian score) from each cluster (Liang et al., 2024). The user must explicitly pass the number of keyframes that are to be returned by this algorithm. Next is Video-kf (Averdones, 2019), a Python module which exploits Ffmpeg (Kiernan and Terzi, 2009) to extract the I-frames (intra-frame coding) of a video to act as the keyframes. Frames are selected that independently best represent the entire video content (Dibenedetto et al., 2024). The third approach is optical flow analysis, where we found the pixel difference between video frames and analyzed the motion information to identify the most important ones (Dong, 2023).

We found Katna to produce the best performance among the three methods. Based on human evaluation, it successfully captured the significant moments in the video. In comparison, Video-kf missed some key moments, and relying solely on pixel differences proved unreliable for identifying important events. We determined the number of keyframes for each video to be extracted by Katna using the following formula:

$$\text{\# keyframes} = \min\left(15, \frac{\text{duration of video (s)}}{10}\right) \tag{1}$$

### 4.4 Frame Captioning

In this section, we focus on generating descriptive captions for the selected keyframes, which form the foundation for the subsequent text summarization step. We experimented with multiple vision-language models to identify the most effective approach for dense, informative frame-level descriptions.

Our initial attempts utilized ViT-GPT2 and BLIP models for image captioning. ViT-GPT2 is a vision encoder-decoder model that combines the Vision Transformer (ViT) (Dosovitskiy et al., 2021) for visual feature extraction and GPT-2 (Radford et al., 2019) for text generation. It leverages ViT's capabilities for spatial feature extraction, paired with GPT-2's robust language modeling to generate captions. Similarly, we employed the pretrained BLIP (Bootstrapping Language-Image Pretraining) framework (Li et al., 2022), a vision-language model that integrates both image understanding and natural language generation. BLIP, trained on the COCO dataset (Lin et al., 2014), performed well in producing concise captions with accurate object detection and relationships.

However, despite their success in standard image captioning tasks, both ViT-GPT2 and BLIP generated captions that lacked the level of detail required for dense video description. The descriptions were often brief and failed to capture fine-grained contextual information essential for comprehensive video summarization.

To address these limitations, we explored more advanced models capable of generating detailed and dense captions. We first experimented with the BART model (Lewis et al., 2019), a denoising autoencoder for sequence-to-sequence tasks. BART's bidirectional encoder and autoregressive decoder enabled the generation of more coherent and contextually enriched descriptions. The output was notably more comprehensive compared to ViT-GPT2 and BLIP, providing improved coverage of visual details within keyframes.

For further refinement, we adopted the Florence model (Yuan et al., 2021), developed by Microsoft for dense vision-language tasks. Florence integrates a large-scale vision backbone optimized for understanding complex scenes. When applied to our keyframe dataset, it produced significantly richer captions, capturing intricate details and spatial relationships that aligned closely with our requirements for dense video descriptions.

Through these iterative improvements, the Florence model emerged as the most effective solution for dense frame captioning. It delivered captions that were descriptive, informative, and well-suited for subsequent summarization tasks, providing the level of detail necessary for our pipeline.

### 4.5 Description Stitching (Summarization)

After successfully generating descriptions of the selected frames, we combine them together into one naturally worded description. We tried out Pegasus (Zhang et al., 2020a) due to its abstractive summarization capabilities and LED (Beltagy et al., 2020) due to its capability of handling long sequences of keyframe captions. However, we found that these models tended to hallucinate when provided the dense image captions. For our task, T5 (Raffel et al., 2023) and Bart (Lewis et al., 2019) models performed the best for the task of video summarization. The encoder-decoder framework found in Bart and the text-to-text framework found in T5 made it ideal for the task of converting these captions to a summarized format. We tested the pre-trained Bart Large CNN model (Borgohain and Agarwal, 2023) and two finetuned T5 models for this task.

#### 4.5.1 Finetuning

By fine-tuning T5, we hoped be able to effectively adapt the existing parameters to suit our task without the time and computational costs of training a model from scratch. To do this, we used 28K datapoints from the Wiki Movie Plot with Summaries dataset (as mentioned in section 4.2.2) to finetune one t5 model by using the Plot as input and Plot Summary as target in hope that the dataset would be similarly suited to our task. We also created a Synthetic dataset (mentioned in section 4.2.3) for finetuning another T5 model on 1K datapoints, taking the dense image captions for our input and the VideoXUM summaries as the output. Implementing this section represents the bulk of our methodology — up till this point, we have used off the shelf solutions, while now we try use finetuning to produce models specifically suited for the task of video summarization in the context of keyframe extraction from videos.

### 4.6 Video Semantic Search

The video semantic search methodology comprises of two main sections :

1. **Video Data Loading to the Vector Database**: In our pipeline, the video description generation module ensures the creation of accurate, concise, and informative textual descriptions from videos. We extract keyframe descriptions and convert them into 384-dimensional embeddings using the SentenceTransformer library (Reimers and Gurevych, 2019) with the "all-MiniLM-L6-v2" pre-trained model. The generated embeddings, along with their associated metadata (e.g., keyframe caption, keyframe ID, timestamp, and video ID), are uploaded to a keyframe-index in a vector database (Pinecone). In addition to keyframe-level representations, we create video-level descriptions (summaries), convert them into 384-dimensional embeddings, and store them in a video-index along with relevant metadata (video summary and video ID).

   This process results in a curated vector database containing structured embeddings for both keyframe descriptions and video summaries, providing a robust foundation for efficient semantic search.

2. **Video Retrieval Using Semantic Search**: Once the video and keyframe data are stored in the vector database, we enable two core retrieval functionalities: **Video Search across the Entire Database** and **Keyframe Search within a Specific Video**. For retrieval, the user query is first converted into a 384-dimensional vector using the same Sentence-Transformer model. The query vector is then compared with the stored embeddings using the HNSW (Hierarchical Navigable Small World) algorithm (Malkov and Yashunin, 2018), which is natively implemented in Pinecone. The semantic search uses cosine similarity as the base metric to identify the most relevant results.

   For both video-level and keyframe-level retrieval, we display the top 5 search results based on their similarity scores. This semantic search mechanism efficiently retrieves relevant video content and keyframes, enabling a scalable and effective solution for large video datasets.

## 5 Experiments

### 5.1 Evaluation

To evaluate our frame descriptions, we used the Image-Paragraph Dataset by using a ROUGE score to compare the keyframe description against the descriptions in the dataset (Yeung et al., 2014). To evaluate our summary descriptions (descriptions generated from the keyframes), we will leverage multiple evaluation methods, including LLM-as-a-Judge (Zheng et al., 2023), ROUGE (Lin, 2004), BLEU (Papineni et al., 2002), and BERTScore (Zhang et al., 2020b).

For LLM-as-a-Judge, we provided the summary to Google's Gemini 1.0 Pro, and provided the following criteria: descriptiveness (is the description vivid and clear), coherence (is the summary easy to understand and does the paragraph structure connect well), completeness (are all points covered from the keyframe descriptions), fluency (is the description grammatically correct), and conciseness (does the description summarize well, or does it simply just put all the sentences together). Using Few-Shot Chain-of-Thought Prompting, we would request a score for each of these criteria from the LLM, which we would average together to be our evaluation score. The prompts were designed such that there were multiple examples for scores from 1 to 5 for each of the criteria, while the LLM was prompted with an explanation for each of the scores.

Our other metrics for our summary descriptions were ROUGE, BLEU, and BERTScore. In particular, we used ROUGE-1, which captures important keywords without being too reliant on word order like ROUGE-2 and ROUGE-L, and BERTScore-F1, which leverages contextual embeddings from the BERT model to capture semantic similarity while maintaining a harmonic balance between precision and recall.

By running the evaluations over the generated summaries, using the VideoXUM descriptions as reference for ROUGE, BLEU, and BERTScore, we were able to generate metrics for comparison between our description stitching models.

Lastly, to evaluate the effectiveness of our semantic search system, we use the SICK (Sentences Involving Compositional Knowledge) (Marelli et al., 2014) dataset. Query-answer pairs are constructed from sentence pairs with a similarity score of 5 (out of 5). For each query, we retrieve the top 5 results from the vector database and evaluate performance using the following metrics:

1. **Mean Reciprocal Rank (MRR):** MRR measures how early the correct answer appears in the ranked results. It is defined as:

$$\text{MRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\text{Rank}_i}$$

where $\text{Rank}_i$ is the position of the correct answer for query $i$, and $N$ is the total number of queries. A higher MRR indicates better ranking quality.

2. **Adjusted Precision@1 (P@1):** Precision@1 measures the proportion of queries for which the correct answer appears at the top logical rank, ignoring the query vector itself. It is given by:

$$\text{P@1} = \frac{\text{Number of Queries with Correct Answer at Rank 1}}{\text{Total Queries}}$$

3. **Mean Similarity Score:** This measures the average similarity between the query and the retrieved correct answer.

## 5.2 Baseline

As a baseline for our video captioning system, we adopted a simple yet effective approach by concatenating captions generated for individual keyframes to form a comprehensive description of the video. While this method ensures that all key elements are represented, it lacks coherency between frames, often resulting in disjointed descriptions.

For our baseline for semantic search, we used FAISS (Facebook AI Similarity Search) (Douze et al., 2024), an open-source library designed for fast and efficient similarity search in high-dimensional vector spaces. FAISS provides both exact and approximate nearest neighbor search, making it ideal for large-scale vector retrieval tasks.

## 5.3 Results

### 5.3.1 Video Captioning

| Method | BLEU | ROUGE-1 | BERTScore-F1 | LLM Judge |
|---|---|---|---|---|
| Baseline | 0.0164 | 0.208 | 0.0124 | 3.029 |
| BART | 0.0163 | 0.246 | 0.0774 | 4.147 |
| T5 (Movie) | 0.0169 | 0.256 | 0.0700 | 4.124 |
| T5 (Synthetic) | 0.0178 | 0.265 | 0.0801 | 4.114 |

Table 1: Evaluation of different caption stitching (summarization) approaches

Table 1 summarizes the evaluation metrics of the methods we used to create the video summaries, including the baseline. About 1000 video examples with human-written descriptions were used to evaluate each method. The baseline naturally scores the lowest on every metric except for BLEU, where it slightly outperforms BART. However, because BLEU focuses on comparing n-grams precision, not accounting for much flexibility or synonymy, it may vary in its consistency to score summarizations. Most notably, T5 fined tuned on the Synthetic dataset scores the highest on BLEU, ROUGE-1, and BERTScore-F1, while having a relatively strong score from the LLM Judge.

### 5.3.2 Video Retrieval Using Semantic Search

| Method | MRR | Precision@1 | Mean Similarity Score |
|---|---|---|---|
| Pinecone (HNSW) | 0.8746 | 0.805 | 0.9528 |
| FAISS (Baseline) | 0.8529 | 0.805 | 0.9047 |

Table 2: Evaluation of HNSW for video sematic search

Using HNSW (Hierarchical Navigable Small World) algorithm, we evaluated our semantic search system on the SICK dataset. About 200 sentence pairs with a similarity score of 5/5 were used to form query-answer pairs for evaluation. As a baseline, we implemented the same semantic search pipeline using (Facebook AI Similarity Search - FAISS (Flat L2 Index). While both systems achieve the same Precision@1, Pinecone outperforms FAISS with higher MRR and Mean Similarity Score, demonstrating better ranking quality and semantic relevance of the retrieved results.

### 5.3.3 Qualitative Results

Given below is an example of a series of extracted keyframes from a video and the corresponding dense captions generated by our proposed system:

> *A group of people in a small inflatable raft on a river appears to be enjoying the ride. The water is calm and there are no other people visible. The sky is blue and the sun is shining brightly, casting a warm glow over the scene. The overall mood of the image is peaceful and serene. The people in the raft are holding onto the raft with their hands and are looking up at the sky. It seems like they are participating in a rescue or rafting activity. The river is surrounded by a rocky cliff and trees on both sides. There are trees and hills in the distance, suggesting that the video was taken in a rural area. A group of four people, two men and two women, are standing next to a large inflatable. They are all wearing helmets and life jackets and are floating on the water.*

### 5.4 Dataset Availability

The VideoXum dataset (Lin et al., 2024) is a publicly available dataset and can be found here. The Wiki Movie Plots with Summaries dataset (Priyavr, 2023) can be obtained from Hugging Face here.

### 5.5 Code Availability

We have added all of our experiments that we have conducted, to a GitHub repository, NLPVideoDescription. The link to our repository can be found here: https://github.com/SRajrah/NLPVideoDescription. Video descriptions using our proposed pipeline can be generated using the code from the main branch.

## 6 Future Work

- **Dynamic keyframe selection**: Rather than explicitly passing the number of keyframes to be returned by extraction algorithm, we could dynamically decide the number of significant frames in a video. The number of keyframes should depend on the complexity of the video and not be based solely on duration (Chakraborty et al., 2015).
- **Utilizing more training samples**: A potential next step for our system is to enhance its robustness by training it on more samples. To improve training efficiency, we could prioritize using shorter-duration videos, which would reduce the computational overhead and improve throughput during the training process.
- **Additional evaluation metrics**: Though BLEU and ROUGE score provide a good insight into the performance of video captioning systems, they rely heavily on exact semantic matches (Schluter, 2017). We could evaluate our system using a more flexible metric while still ensuring certain keyword matches are found.
- **Output post-processing**: While the output consistently summarizes the captions in a concise yet detailed manner, it occasionally includes that the content it is summarizing is an image. We should include a methodology to post-process the output, such that we do not have any indication that the video was broken into keyframes in the summary. We could do this through adding another LLM to the pipeline which is in charge of post-processing the summary.

## 7 Conclusion

We proposed a methodology for dense video description generation geared towards creating detailed summaries that effectively describes the contents of the video. Additionally, we introduced a semantic retrieval mechanism that uses these generated descriptions for video semantic search and retrieval tasks. Our results show that we have successfully achieved the first objective. Our framework, which utilizes a T5 model fine-tuned on a synthetic dataset we created, outperforms the baseline approach of concatenating key frame captions.

The generated descriptions capture a significant amount of information, including the setting, actions of people and animals, positional details, and more. This is a significant improvement over typical video textual summaries, which tend to provide very simple summaries rather than a comprehensive play-by-play description. Using these descriptions as a basis for video comparison has also been shown to outperform the baseline methodology. This is due to the additional contextual and semantic details included in the generated descriptions, which provide more information for comparison than the simplified summaries of other video-to-text summarization models.

In conclusion, our methodology not only enhances the quality and granularity of video descriptions, but also provides an improvement in video semantic search and retrieval tasks, making it a valuable addition to the field of video content understanding.

## References

Averdones. 2019. Quick guide. https://github.com/averdones/video-kf.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *Preprint*, arXiv:2004.05150.

Rajdeep Borgohain and Nilesh Agarwal. 2023. Bart-large-cnn. https://github.com/inferless/Facebook-bart-cnn.

Shayok Chakraborty, Omesh Tickoo, and Ravi Iyer. 2015. Adaptive keyframe selection for video summarization. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 702–709.

Gaetano Dibenedetto, Marco Polignano, Pasquale Lops, and Giovanni Semeraro. 2024. Human pose estimation for explainable corrective feedbacks in office spaces. In *Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*, pages 264–275.

Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. 2016. Long-term recurrent convolutional networks for visual recognition and description. *Preprint*, arXiv:1411.4389.

Jinshan Dong. 2023. Study on video key frame extraction in different scenes based on optical flow. *Journal of Physics: Conference Series*, 2646(1):012035.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *Preprint*, arXiv:2010.11929.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library. *Preprint*, arXiv:2401.08281.

Zhenzhen Hu, Zhenshan Wang, Zijie Song, and Richang Hong. 2023. Dual video summarization: from frames to captions. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, IJCAI '23.

Jie Jiang, Shaobo Min, Weijie Kong, Dihong Gong, Hongfa Wang, Zhifeng Li, and Wei Liu. 2022. Tencent text-video retrieval: Hierarchical cross-modal interactions with multi-level representations. *Preprint*, arXiv:2204.03382.

KeplerLab. 2019. Katna: Tool for automating video keyframe extraction, video compression, image autocrop and smart image resize tasks. https://github.com/keplerlab/katna.

Jerry Kiernan and Evimaria Terzi. 2009. Constructing comprehensive summaries of large event sequences. *ACM Trans. Knowl. Discov. Data*, 3(4).

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017a. Dense-captioning events in videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017b. Dense-captioning events in videos. *Preprint*, arXiv:1705.00754.

Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. 2021. Less is more: Clipbert for video-and-language learning via sparse sampling. *Preprint*, arXiv:2102.06183.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *Preprint*, arXiv:1910.13461.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *Preprint*, arXiv:2201.12086.

Hao Liang, Jiapeng Li, Tianyi Bai, Xijie Huang, Linzhuang Sun, Zhengren Wang, Conghui He, Bin Cui, Chong Chen, and Wentao Zhang. 2024. Keyvideollm: Towards large-scale video keyframe selection. *Preprint*, arXiv:2407.03104.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Jingyang Lin, Hang Hua, Ming Chen, Yikang Li, Jenhao Hsiao, Chiuman Ho, and Jiebo Luo. 2024. Videoxum: Cross-modal visual and textural summarization of videos. *Preprint*, arXiv:2303.12060.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. *Preprint*, arXiv:1405.0312.

Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. 2021. Clip4clip: An empirical study of clip for end to end video clip retrieval. *Preprint*, arXiv:2104.08860.

Yu. A. Malkov and D. A. Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *Preprint*, arXiv:1603.09320.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Mayu Otani, Yuta Nakashima, Esa Rahtu, Janne Heikkilä, and Naokazu Yokoya. 2016. Video summarization using deep semantic features. *Preprint*, arXiv:1609.08758.

Pinelopi Papalampidi and Mirella Lapata. 2022. Hierarchical3d adapters for long video-to-text summarization. *Preprint*, arXiv:2210.04829.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Jesus Perez-Martin, Benjamin Bustos, Silvio Jamil F. Guimarães, Ivan Sipiran, Jorge Pérez, and Grethel Coello Said. 2021. A comprehensive review of the video-to-text problem. *Preprint*, arXiv:2103.14785.

Vishnu Priyavr. 2023. Wiki movie plots with summaries. `https://huggingface.co/datasets/vishnupriyavr/wiki-movie-plots-with-summaries`.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Natalie Schluter. 2017. The limits of automatic summarisation according to rouge. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 41–45. Association for Computational Linguistics.

Qiang Wang, Yanhao Zhang, Yun Zheng, Pan Pan, and Xian-Sheng Hua. 2022. Disentangled representation learning for text-video retrieval. *Preprint*, arXiv:2203.07111.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5288–5296.

Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. 2021. Filip: Fine-grained interactive language-image pre-training. *Preprint*, arXiv:2111.07783.

Serena Yeung, Alireza Fathi, and Li Fei-Fei. 2014. Videoset: Video summary evaluation through text. *Preprint*, arXiv:1406.5824.

Lu Yuan, Jianlong Chen, Tao Chen, Noel Codella, Xiyang Dai, , et al. 2021. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *Preprint*, arXiv:1912.08777.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. 2018a. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. 2018b. End-to-end dense video captioning with masked transformer. *Preprint*, arXiv:1804.00819.