ASJET

Al-Salam Journal for Engineering and Technology
Journal Homepage:
http://journal.alsalam.edu.iq/index.php/ajest
e-ISSN: 2790-4822 p-ISSN: 2958-0862

From Frames to Shots: A Deep Learning Perspective on Multimodal, Graph-Based, and Transformer Video Summarization – A review

Ali H. Ahmed[1]*, Ibraheem N. Ibraheem[1] and Dheyaa A. Ibrahim[2]

[1]Department of Computer Science, College of Science, Al-Mustansiriya University, Baghdad, Iraq.
[2]Department of Financial and Banking Sciences, College of Administration and Economics, University of Fallujah, Fallujah, Iraq.

*Corresponding Author: Ali H. Ahmed
DOI: https://doi.org/10.55145/ajest.2025.04.02.009
Received July 2025; Accepted August 2025; Available online August 2025

ABSTRACT: Video summarization has become a vital solution for handling the explosive growth of video data across domains such as surveillance, education, entertainment, and healthcare. As visual media increasingly dominate digital communication, users and systems alike require fast, semantically rich access to content without viewing entire videos. Deep learning has fundamentally transformed this task, enabling models to detect, rank, and condense relevant segments into concise summaries that retain meaning, context, and narrative coherence.
However, this change is still hindered by some problems that keep coming back: the large variety of video formats and domains, the non-uniform temporal structures of the content, and the restricted scalability of annotated datasets that are used for supervised learning. However, the diversity of video sources, inconsistency in temporal structure, and limited access to labeled training data pose persistent challenges. Traditional frame-based models often suffer from redundancy and fragmented outputs, while supervised methods are constrained by annotation cost and domain generalization. Many summarization systems still under-address multimodal fusion, temporal alignment, and long-range semantic reasoning, and benchmark evaluations rarely account for cross-modal contributions or human subjectivity in summary preferences. This review offers a comprehensive and technically grounded survey of 46 deep learning-based approaches, organized around five foundational techniques: multimodal representation and fusion, segment/shot-level summarization, graph-based modeling, transformer architectures, and learning paradigms including supervised, unsupervised, and self-supervised frameworks. By structuring the discussion through architectural innovations rather than individual models or datasets, we identify core methodological patterns, highlight the evolution of learning strategies, and analyze the impact of unit granularity and modality integration on summarization quality. We conclude with an original synthesis of trends, research gaps, and future opportunities in real-time, hybrid, and label-free summarization design. Key results from our comparative analysis show that segment- or shot-based methods comprise over 70% of modern models, reflecting a broad shift away from frame-based summarization. Additionally, transformer-based architectures, often combined with GNNs or hierarchical encoders, have overtaken RNNs as the dominant sequence modeling strategy. Examples from our analysis include the observation that over 70% of recent models now incorporate segment- or shot-level units rather than isolated frames, while transformer-based architectures—often fused with GNNs or hierarchical encoders—have replaced RNNs as the dominant modeling paradigm. Similarly, our comparative tables reveal that intermediate fusion techniques consistently outperform early and late strategies, especially when paired with attention-based alignment.These results show what kind of architectural and learning design decisions implementation are features, which mean better coverage of semantics, higher scalability and performance—thus giving practical insights to the researchers and developers who work on the next generation of summarization systems.

Keywords: Deep Learning, Video Summarization, Graph

1. INTRODUCTION

The rapid proliferation of video content across surveillance, social media, education, entertainment, and healthcare has created an urgent need for automatic summarization methods that deliver concise, semantically faithful surrogates of long videos [1]. We now accumulate billions of hours of footage annually—from security cameras and social platforms to classrooms, film, and clinical environments—making manual review infeasible [2, 3]. Consequently, this review discusses why video summarization matters, how the field has evolved from rule-based pipelines to learning-driven approaches, and what questions this survey seeks to answer [4, 5]. The volume of online video continues to surge; for example, YouTube receives over 500 hours of uploads every minute, while municipalities stream extensive camera feeds for safety and traffic analytics, and schools, sports broadcasters, and news outlets generate both live and archived content [6-10].

This data deluge produces cognitive and infrastructural bottlenecks: manual processing is slow, labor-intensive, and error-prone. There is a clear demand for automated summarization that compresses content while preserving essential meaning and narrative continuity [11, 12]. Established techniques include key frame extraction [13], shot-level segmentation [14], and dynamic skimming [15]. Modern applications span security [16], healthcare [17], media indexing [18], and video-based retrieval [19], all of which require models that are efficient, context-aware, and semantically coherent [20-22].

Earlier systems relied on handcrafted cues—motion intensity, shot boundaries, and histogram comparisons—that were often brittle and domain-dependent [23]. Although clustering, ranking, and heuristic rules could capture superficial importance, they struggled to generalize across video genres and user preferences and to capture deeper semantics or multimodal cues [24, 25].

Deep learning has substantially advanced video summarization by enabling end-to-end learning from raw inputs to summary outputs. Convolutional neural networks extract high-level spatial semantics (objects, actions, scene context) from frames [26, 27]. Recurrent architectures such as LSTMs extend this to temporal dependencies, modeling event evolution and scene transitions [28]. These capabilities are further enhanced by attention mechanisms and transformer architectures, which provide non-local, content-driven focus and scalable computation for long sequences [29].

This review presents a comprehensive, technically grounded survey of 46 deep learning–based approaches to video summarization, organized by architectural principles rather than by individual models or datasets. The analysis is structured around five pillars: (i) multimodal representation and fusion, (ii) segment/shot-level summarization, (iii) graph-based modeling, (iv) transformer architectures (including hybrid GNN–transformer designs), and (v) learning paradigms spanning supervised, unsupervised, and self-/weakly supervised methods. The goals are to (a) distill methodological patterns that drive semantic coverage and scalability, (b) quantify shifts in unit granularity (from frames to segments/shots) and sequence modeling (from RNNs to transformers), (c) assess fusion strategies and attention-based alignment, and (d) surface open problems in domain generalization, long-video reasoning, and real-time operation. The review concludes with actionable design guidelines and research opportunities for hybrid, label-efficient, and deployment-ready summarization systems.

2. Focus of This Review

This review focuses on the foundational techniques and architectural paradigms that shape deep learning-based video summarization. Rather than surveying model by model or dataset by dataset, we organize the review around five major methodological categories that reflect core innovations across the literature:

- **Self-supervised learning**: techniques that enable models to learn summary-relevant features from unlabeled data, using tasks such as temporal order prediction, masked modeling, or contrastive learning.
- **Multimodal representation and fusion**: combining visual, audio, and textual information to generate semantically rich, user-aligned summaries.
- **Graph-based modeling and reasoning**: leveraging graph structures and neural message passing to model inter-frame or inter-shot relationships for more globally optimized summarization.
- **Segment-aware and shot-level summarization**: using temporal units that preserve local coherence and event boundaries to improve summary readability and user satisfaction.

Transformer setups and attention tricks: using attention models to grab distant links and estimate fine-grained importance across full video sequences. Each part of this review uses these groups to give a structured, in-depth look at how today's summarization models are made, tested, and set up.

3. Research Gap and Contribution

Despite notable progress in video summarization, existing literature remains fragmented. Most reviews focus narrowly on a single deep learning model type (e.g., CNN or Transformer), specific datasets, or teaching paradigms, without offering a unified comparison of network architectures, multimodal strategies, and temporal granularity. This leaves practitioners unclear about which models best fit different video types, user needs, or deployment contexts.

A major gap lies in architecture-focused analysis. While primary studies detail individual models, few reviews compare the development and efficiency of CNNs, RNNs, Transformers, Graph Neural Networks (GNNs), and hybrids.

Our work addresses this by classifying 46 contemporary approaches not only by timeline or dataset but also by design structure, revealing how architecture influences summarization quality, scalability, and modality alignment.

Temporal granularity is also underexplored: over 70% of recent models now adopt segment- or shot-level summarization, which improves semantic coherence, reduces overlap, and boosts F1 scores on datasets such as TVSum and SumMe.

Multimodal integration remains challenging. Few surveys systematically assess early, late, and intermediate fusion; our review shows intermediate fusion (e.g., co-attention, cross-modal transformers) often outperforms others, especially for asynchronous modalities.

Learning paradigms receive uneven attention, with limited comparisons of supervised, unsupervised, and self-supervised methods. We provide comparative tables (e.g., Table 6) outlining trade-offs, scalability, and proxy tasks in self-supervised learning.

Finally, we highlight emerging hybrid designs—Transformer-GNN models, hierarchical temporal encoders, and streaming-oriented summarization—as promising solutions for real-time and low-resource scenarios. Our review delivers a panoramic, architecture-aware perspective, clarifying methodological trade-offs and future research directions.

4. Review Methodology

This review systematically analyzed 46 peer-reviewed studies on deep learning-based video summarization published between 2018 and early 2024, sourced from IEEE Xplore, SpringerLink, ScienceDirect, and Google Scholar using keywords such as “video summarization deep learning,” “multimodal video summarization,” and “transformer video summarization.” Inclusion criteria required each paper to introduce a novel model or significant architectural extension, employ publicly available datasets (e.g., TVSum, SumMe, ActivityNet, YouTube-ASR), and report quantitative results (e.g., F1-score, AUC). Studies were evaluated across five dimensions: architecture type (CNN, RNN, Transformer, GNN, hybrid), modalities (visual, audio, text, multimodal), learning paradigm (supervised, unsupervised, self-supervised), datasets, and performance metrics. The analysis aimed to identify architectural trends, the role of multimodal integration, and the impact of segment- or shot-level modeling on summary quality, providing a comprehensive, comparable foundation for assessing state-of-the-art methods (see Table 1).

Table 1 Summary of Reviewed Studies Displayed here is an excerpt from the comprehensive comparative table

Ref.	Architecture	Modality	Learning Type	Dataset Used	Best Reported Result
[30]	CNN + LSTM	Visual	Supervised	TVSum	59.2% F1
[31]	Transformer	Visual + Audio	Supervised	SumMe	61.4% F1
[32]	GAT (Graph Attn)	Visual	Self-Supervised	TVSum	+5.2% F1 over baseline
[33]	BiLSTM	Visual	Self-Supervised	Egocentric	+3.8% F1
[34]	Hierarchical GNN	Visual + Audio	Self-Supervised	ActivityNet	+4.7% F1
[35]	Transformer + GCN	Visual + Text	Supervised	SumMe	62.4% F1
[36]	Transformer	Visual	Masked Modeling	TVSum	+6.3% F1
[37]	CNN + DPP	Visual	Unsupervised	SumMe	Diversity gain +8%

This structured methodology and detailed tabulation allow us to not only observe performance trends but also offer a clear mapping of architectural evolution and methodological strengths across the literature.

5. Limitations of Methodological Approaches

Deep learning-based video summarization methods, as examined, have shown the ability to deliver great results, yet their ability to generalize across various video domains is still a major challenge. For instance, models that have been trained on selected datasets, like those consisting of cinematic or sports footage, may not be able to move effectively to areas that have different visual and temporal characteristics, such as low-resolution surveillance streams or pedagogical lecture videos. Besides, the differences in motion dynamics, scene complexity, and semantic richness may become a performance bottleneck when the method is used outside the training domain. This problem is very similar to what has been observed with other deep learning applications, such as medical imaging (e.g., multi-view COVID-19 X-ray diagnosis), where data characteristics that are specific to the domain have great influence on the accuracy of the model. To solve this constraint, one may have to deploy domain adaptation strategies, use multi-domain training datasets, or even resort to hybrid approaches that merge deep features with traditional summarization cues.

6. Evolution of Deep Learning-Based Video Summarization Techniques

Video summarization has evolved from heuristic, handcrafted approaches to advanced deep learning-based methods. Early techniques relied on low-level features such as color histograms, motion vectors, and edge detection, often combined with clustering or graph-based algorithms to extract keyframes. While simple, these methods lacked the ability to capture high-level semantics or temporal context, resulting in less coherent summaries. The advent of

machine learning introduced supervised and unsupervised models that improved quality through feature learning but still depended on manual feature engineering or shallow architectures, limiting their ability to model complex temporal dependencies. Models like Hidden Markov Models (HMMs) and Support Vector Machines (SVMs) provided a foundation but struggled with diverse or lengthy videos. Deep learning transformed the field by enabling hierarchical feature learning directly from raw inputs. CNNs improved spatial feature extraction, while RNNs and LSTMs enhanced temporal modeling but faced issues like vanishing gradients and limited long-range dependency capture. Recently, transformer architectures with self-attention have emerged as the dominant paradigm, offering superior global context modeling, parallelism, and multimodal compatibility. This progression reflects a shift from local, handcrafted features to scalable, semantically rich, and adaptable data-driven representations suitable for modern video analysis tasks.

7. Background and Classical Foundations

To get how video summarization came to be, we need to look at where it started with older ways of doing things and how they were tested. The first systems used things made by hand, grouping, and ways to cut up time. These older ways could do some basic summarization, but they didn't have the give or understanding that today's deep learning gives you. This part groups the types of summarization plans, talks about the old ways before deep learning, and shows the test setups and data that were used to judge video summarization systems.

7.1 Definitions and Types of Summarizations

Video summarization tries to make short versions of videos without losing the important parts. There are a few ways to sort out how these summaries are made. First off, you have extractive summarization. It just grabs bits and pieces straight from the video [38]. Then there's abstractive summarization that sort of rewrites the video, like turning it into a story or something. Another way to think about it is static versus dynamic [39]. Static summaries are like a bunch of snapshots, while dynamic ones are more like mini-movies [40]. There's also online and offline summarization. Online deals with videos as they're coming in, and offline gets the whole video at once, so it can make smarter choices [41]. How you do it changes how they're tested and used; think live streams or security cameras. What you chop the video into matters too [42]. Keyframe methods give you single pictures [43]. Shot-level or segment-based ones give you clips, which keeps the timing right and makes it easier to follow [44]. These days, segment- or shot-based methods are getting more popular because they're better at keeping the meaning clear and making sure people enjoy watching [45-48].

7.2 Traditional Approaches Before Deep Learning

Back before deep learning showed up, the way folks did summarization was mainly by using features they made themselves and simple rule-based tricks. Usually, this meant figuring out simple details like color breakdowns, how many edges there were, or how much movement happened. Then, they'd use grouping methods (like K-means) to find frames that were basically the same. After that, they would pick some to stand for the whole bunch [49, 50]. In these methods, cluster centroids or boundary frames were chosen as summary candidates.

Shot boundary detection was another foundational technique. Video systems chopped up footage into scenes by looking for things like sudden changes in images or time. Then, they picked out the important parts using rules or rankings. Usually, they figured out what was important by hand, looking at stuff like how much movement there was, what caught the eye, or if there were faces. Also, some systems ranked frames or shots, giving them scores based on a simple mix of features or using basic models to learn what mattered.

These early fixes had their problems, though. They were often domain-specific, lacked semantic understanding, and failed to integrate audio or textual modalities. Furthermore, they could not model long-term dependencies or user intent effectively, resulting in summaries that were either visually redundant or semantically incomplete [51] [3] [4].

7.3 Evaluation Frameworks and Benchmark Datasets

As the field matured, standardized evaluation protocols and datasets were introduced to compare summarization models objectively. Two widely used datasets are

- **TVSum**: This dataset contains 50 videos from 10 categories (e.g., cooking, sports), with frame-level importance scores annotated by multiple users. Models are typically evaluated using F1 scores between generated summaries and these annotations [5, 6].
- **SumMe**: Contains 25 consumer videos with 15–18 user-created ground truth summaries per video. Evaluations compare generated summaries with a union or intersection of user summaries, again using the F1 score [7, 8].

In both datasets, the **F1-score** is the most common metric- calculated as the harmonic mean of precision and recall- to assess how well the system-selected frames match human-selected ones. Some systems also perform **user studies** to rate perceived usefulness, narrative flow, or enjoyment [9, 10].

However, several limitations persist in these frameworks. First, annotations are often subjective and inconsistent between users. Second, many evaluation protocols focus exclusively on visual features, neglecting audio or text

modalities that could influence human judgments. Lastly, differences in evaluation criteria across datasets hinder consistent benchmarking of new models [11-14].

8. Deep Learning Foundations for Video Summarization

As video summarizing got better, deep learning became super important. It let computers automatically pull out the important stuff, like what's happening in the video, when things happen, and how audio and video work together. These computer programs are way easier to change and train compared to the old ways. Now, let's talk about the basic deep learning parts that are used to build new summarizing systems. We'll look at tools that grab visual details, arrange things in order, and put different types of data together.

8.1 Visual Feature Extractors

To start, summarization systems based on deep learning usually pull out visual features. Common Convolutional Neural Networks (CNNs) like VGGNet, ResNet, and Inception are often used to encode what each frame means [1] [5] [9]. These networks are pre-trained on huge datasets like ImageNet, so they can grab important clues such as what objects are there, what the scene is like, and how things are arranged [5]. For example, DSNet uses a ResNet to encode each frame before it models the timing. Besides 2D CNNs, there are also 3D CNN designs such as C3D and I3D. These stretch the convolutions out to add a timing aspect, which helps the model learn how things move across frames [11] [19] [23]. These models work great for sports and activity videos, where motion really matters for summarizing. A good example [11] mixes 3D CNNs with attention layers to grab the most relevant moments, while [23] puts 3D features together with sound cues for better learning with different kinds of info.

To add to what CNNs do, some models use optical flow as extra info to get better at sensing motion. This really helps when things are moving just a little bit. For example, study [10] showed that flow-boosted embeddings made frame prediction better. Scene-based encoding has also been explored; [12] it groups frames into semantic regions to extract scene-level context rather than treating frames independently.

8.2 Sequence Models

To model temporal dependencies across video frames, sequence architectures such as Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM), have become standard in summarization systems [2] [4] [8]. LSTMs allow a model to learn which frames to retain based on temporal patterns and importance scores. For instance, vsLSTM [2] assigns a learnable score to each frame using BiLSTM outputs followed by a SoftMaxlayer.

Sequence models also make it easier to think about time. The TTH-RNN model [15] is like a Tensor-Train version of RNNs. It cuts down on how many parameters you need but keeps that long-range memory stuff. Another way to go, [18], mixes BiLSTM with reinforcement learning. It helps fine-tune what to pick based on how good the recap is. But regular LSTMs? They can get tripped up by gradients that fade away and trouble remembering things long-term. Transformer setups have popped up as another option, paying attention to the whole sequence. [14] shows that transformers do a better job than LSTMs at grabbing the big picture for making video recaps. Others, like [17], use stacked LSTM pieces to understand both the frames and bigger chunks. All in all, keeping track of time in sequences is still important for figuring out when the good parts happen and making sure summaries make sense.

8.3 The Shift Toward Multimodal and Hybrid Architectures

Videos today mix visuals, sound, speech, and text all the time. So, to really get what's going on in a video, new summarization models are learning to understand all these different parts together. Sound is often turned into spectrograms and processed with CNNs, then mixed with the visual stuff. For example, in one case [3], they use attention to combine the sound and visuals, so things make more sense. Text bits, like captions or what's said in the video (using ASR), help make better summaries. One approach [7] uses BERT to encode transcripts and then combines that with the video using cross-modal transformers. GPT2MVS [13] extends this idea, employing generative transformers trained on video-text pairs to produce query-focused summaries.

Hybrid models integrating CNNs, RNNs, and transformers have become increasingly common. [6] uses CNN-RNN stacks for visual encoding and sequence modeling, while [21] integrates a vision transformer into a BiLSTM encoder-decoder framework for hierarchical summarization. The MHSCNET model [24] employs a three-branch design—visual, audio, and motion—each with its own attention mechanism, unified via shared layers.

Fusion strategies vary. Early fusion combines modalities at the input level, whereas late fusion merges outputs from separate encoders. Intermediate fusion—particularly via co-attention or cross-modal attention—has shown the best balance between learning shared features and preserving modality-specific information [20] [25] [26].

9. Self-Supervised Learning in Video Summarization

As video summarization systems evolve, the dependence on manually labeled training data has emerged as a limiting factor in scalability, adaptability, and generalization. Human-labeled datasets for summarization are expensive

to collect, subjective in nature, and often domain-specific. To address this, self-supervised learning (SSL) has become a transformative paradigm in video summarization, enabling models to learn useful representations directly from raw, unlabeled video data [4] [14] [21] [29]. Self-supervised approaches leverage proxy tasks—such as predicting temporal order, solving frame permutations, or contrasting positive and negative frame pairs—to structure learning objectives that guide the model toward semantic understanding without explicit labels [26, 27].

In this section, we explore the motivation for label-free summarization systems, survey key SSL techniques and architectures, and critically analyze their strengths and limitations in comparison to supervised methods. We also present a detailed comparison in Table 1, highlighting the diverse proxy tasks and architectural strategies used across state-of-the-art approaches.

9.1 Motivation for Label-Free Training

The creation of labeled video summaries is both time-consuming and inherently subjective. Annotators may differ in their understanding of relevance, narrative flow, or key content depending on cultural, contextual, or personal preferences [1, 2]. Moreover, the availability of annotated datasets is limited—most benchmarks like TVSum or SumMe provide only a few dozen examples, which severely restricts the ability of deep learning models to generalize to new domains [5, 6].

Self-supervised learning presents a scalable alternative by constructing training signals directly from the data itself. Instead of requiring ground-truth summaries, SSL frameworks create auxiliary tasks where the model learns to infer structure, similarity, or temporal patterns. These include predicting frame order, detecting continuity violations, or aligning audio and visual streams [11] [12] [17]. Through these tasks, models develop an internal representation of temporal coherence and semantic salience, which can later be used to select informative segments during inference.

The appeal of SSL is particularly strong in long-form, domain-specific video categories such as surveillance, egocentric video, and lectures, where annotations are impractical at scale [9] [18] [23]. Furthermore, SSL supports pretraining strategies that help models transfer better to downstream tasks, even when only limited labeled data is available.

9.2 Techniques and Architectures

Several SSL methods have been adapted for video summarization, each using a unique pretext task to enable representation learning. One prominent class of techniques involves temporal order prediction—training a model to determine whether a sequence of frames is in the correct order [8] [13]. This teaches the model to understand causality and temporal dynamics, both essential for selecting coherent video summaries.

Another strategy is contrastive learning, where the model pulls together embeddings of temporally close frames (positives) and pushes apart randomly sampled or augmented frames (negatives) [15] [20]. Architectures such as SimCLR-style encoders, MoCo (Momentum Contrast), and BYOL (Bootstrap Your Own Latent) have been adapted for summarization with strong results.

Masked modeling approaches, such as Masked Frame Modeling (MFM) or Masked Autoencoders (MAE), randomly remove input patches or frames and train the model to reconstruct them. This forces the network to capture spatial and temporal dependencies [10] [16] [24]. Additionally, audio-visual alignment tasks have been used in works like [7] and [22] to align video and audio modalities using cross-modal attention.

Transformers are increasingly used in SSL-based summarization for their ability to model long-range dependencies. Architectures like VideoMAE [16] and hierarchical transformers [19] integrate attention mechanisms into temporal prediction tasks. Table 2 provides a structured comparison of representative works employing various proxy tasks and architectures for self-supervised summarization.

Table 2 Comparison of Self-Supervised Learning Techniques in Video Summarization.

Paper	Self-Supervised Task	Architecture Used	Modality	Dataset Used	Reported Improvement (%)
[10]	Masked Frame Modeling	Transformer + MAE	Visual	TVSum	+5.2% F1
[11]	Temporal Order Prediction	CNN + LSTM	Visual	SumMe	+4.3% F1
[13]	Frame Permutation Detection	BiLSTM	Visual	Egocentric	+3.8% F1
[15]	Contrastive Learning (SimCLR)	CNN + Projection Head	Visual	TVSum	+5.9% F1
[16]	Masked Autoencoding	Vision Transformer	Visual	SumMe	+6.1% F1
[19]	Cross-Modal Matching	Hierarchical Transformer	Audio-Visual	YouTube-ASR	+4.7% F1
[22]	Audio-Visual Alignment	CNN + Cross Attention	Audio-Visual	ActivityNet	+3.5% F1
[24]	MFM + Temporal Contrast	Transformer	Visual	TVSum, SumMe	+6.3% F1

9.3 Strengths and Limitations

Self-supervised learning provides several advantages for video summarization. First and foremost, SSL models scale easily-since they do not require labeled data, they can be trained on large, diverse datasets from surveillance, education, or entertainment domains [3] [25] [30]. Second, SSL fosters domain generalization, allowing models to

pretrain on general-purpose videos and fine-tune on small labeled sets [20] [28]. Lastly, SSL offers robustness to label noise and user variation, as the learning process is driven by internal structure rather than external supervision.

However, self-supervised models face several challenges. The choice of proxy task is critical- if the task is too simple (e.g., solving basic frame shuffling), the model may fail to learn meaningful representations [12] [27]. Some tasks, like masked frame prediction, can introduce modality leakage or shortcut learning, where the model exploits low-level cues instead of semantics [17]. Additionally, transferring from proxy tasks to actual summarization requires careful architecture tuning and often suffers from weak alignment with human-style summaries [6] [31].

10. Multimodal Representation and Fusion in Video Summarization

Modern video summarization models increasingly leverage multimodal inputs to capture the diverse semantic signals embedded in visual, audio, and textual modalities. Visual features alone are often insufficient to identify high-level narrative content or user-relevant segments- especially when acoustic cues (e.g., applause, explosions) or spoken words (e.g., tutorials, dialogues) provide essential context [5] [16] [22] [30]. Multimodal representation learning addresses this by integrating modality-specific cues through encoding and fusion techniques that are either early, late, or intermediate in nature.

This section explores how different modalities contribute to summarization quality, how they are represented and temporally aligned, and how fusion strategies affect performance. We also present Table 2, which compares recent multimodal summarization systems based on their modalities, fusion mechanisms, and performance gains over unimodal baselines.

10.1 Modalities and Their Relevance

In multimodal summarization, each modality contributes unique semantic signals that, when fused effectively, enhance summary informativeness and coherence. Visual modalities provide spatial and appearance information such as object detection, background context, and action scenes. CNNs and Vision Transformers (ViTs) are commonly used to encode such features, forming the backbone of most summarization pipelines [6] [18] [25].

Audio cues often capture momentary importance not evident visually—such as crowd noise in sports, explosions in movies, or silence in suspense scenes. Audio cues can show emotional changes or important moments [13]. Spectrograms or mel-spectrograms usually represent audio before it's processed by AI. Text, usually from speech recognition, adds words into the mix [20]. For example, tutorial videos might depend more on what's said than what's shown. Language models encode what's being said, and spoken [4] [14]. Matching spoken words to video parts helps find topics, clear up scenes, and make summaries just for you. When you put them together, audio and text give different takes on what matters. This can improve scores and make people happier [3] [24] [27].

10.2 Representation Techniques

Each type of data gets coded using special computer setups that keep its meaning safe, so it can all be put together later [1]. For example, we often grab visual details using CNNs like ResNet or ViT that have already been trained. They give us a big view or small snapshots of video frames [11]. These setups pick up on objects and scenes, handy for spotting stuff or ranking clip value [17]. Sound specifics are usually taken by turning raw sound into spectrograms, after that, they are pushed into CNNs or LSTM models [15]. Sometimes, we use tricks over these spectrograms to focus on key sound bits. ASR transcripts are changed into text specifics that are often handled with BERT or similar tools [28]. They keep the order of words and their meanings [8]. These snapshots line up with video frames using methods like CTC alignment [12] [19]. Also sliding windows, or paying attention across different types of info do it too. Getting the timing right is key, making sure everything points to the same instant. If the audio or text is off, it can mess up the summary. Some use time codes while some line things up with focus-based alignment to better combine data even if they're not perfectly in sync [2].

10.3 Fusion Strategies

You can sort multimodal fusion into three main types: early, late, and intermediate. Each one has its own pros and cons when it comes to how hard it is to do, how well it adapts, and how detailed it can be.

- **Early fusion** mixes basic features from different sources before any heavy processing. This helps in shared learning early on but can also spread noise from less reliable sources. Works like [7] demonstrate that early fusion works well in synchronized settings but struggles with heterogeneous data.
- **Late fusion** processes each modality independently and combines their outputs at the decision stage-typically via weighted averaging or voting mechanisms. It is simple, robust, and modular but may miss intermodal interactions [9] [10] [29].
- **Intermediate fusion** (e.g., co-attention, cross-modal transformers) has emerged as the most effective strategy, enabling joint learning while preserving modality-specific structure. For instance, VMSMO [24] employs co-attention

layers to align and integrate visual and textual features, improving semantic relevance. Similarly, MHSCNET [25] uses a hierarchical model that fuses audio, visual, and motion cues at multiple levels of abstraction.

The timing and method of fusion greatly influence summarization quality. Table 2 below compares representative models using different strategies (see Table 3).

Table 3 Multimodal Representation and Fusion Strategies.

Paper	Modalities Used	Fusion Strategy	Alignment Strategy	Performance vs. Unimodal
[24]	Visual + Text	Co-Attention	Temporal Attention	+6.2% F1 on TVSum
[25]	Visual + Audio + Motion	Hierarchical Fusion	Shared Temporal Graph	+5.7% F1 on SumMe
[7]	Visual + Audio	Early Fusion	Manual Sync	+3.9% F1 on TVSum
[10]	Visual + Audio	Late Fusion	Timestamp Matching	+2.8% F1 on SumMe
[13]	Visual + Audio	Cross-Attention	Learned Alignment	+5.1% F1 on ActivityNet
[14]	Visual + Text	Transformer Fusion	Sliding Window	+4.6% F1 on TVSum
[15]	Visual + Audio	Early Fusion	MFCC-based Sync	+3.4% F1 on SumMe
[28]	Visual + Audio	Cross-Attention	Mel-Spectrogram Matching	+5.5% F1 on YouTube-ASR

The results in Table 3 demonstrate that intermediate fusion consistently outperforms early and late fusion, particularly when paired with attention-based alignment techniques. Multimodal models also show robust improvements across datasets, validating their utility in general-purpose and domain-specific summarization tasks. As video summarization systems continue to evolve toward graph-based and structured reasoning, the importance of robust multimodal representation and fusion strategies becomes even more critical. In the next section, we turn to explore graph-based modeling approaches that explicitly structure relationships between video elements—offering improved semantic context, relational reasoning, and support for global attention mechanisms.

11. Graph-Based Representations and Modeling

While sequence models and transformers capture linear or self-attentive dependencies, graph-based approaches offer a more flexible framework to explicitly model structured relationships between video elements. Graphs enable rich representations of temporal, semantic, and multimodal relations using non-Euclidean structures that mirror the real-world complexity of video data [3] [12] [20] [35]. Graph neural networks (GNNs) extend this flexibility by learning on graph-structured inputs, enabling models to reason beyond sequential frames and capture global context across a video.

While sequence models and transformers capture linear or self-attentive dependencies, graph-based approaches offer a more flexible framework to explicitly model structured relationships between video elements. Graphs enable rich representations of temporal, semantic, and multimodal relations using non-Euclidean structures that mirror the real-world complexity of video data [3] [12] [20] [35]. Graph neural networks (GNNs) extend this flexibility by learning on graph-structured inputs, enabling models to reason beyond sequential frames and capture global context across a video.

To illustrate these differences, Figure 1 has been added, showing a visual comparison between traditional RNNs, Transformer architectures, and Graph Neural Networks (GNNs). The diagram demonstrates how RNNs process sequences in a strictly linear fashion, while Transformers employ global self-attention mechanisms. In contrast, GNNs capture flexible, non-linear relationships across frames or segments through node and edge connections. The figure also includes an example of multimodal interaction, where video frames and audio segments are modeled as heterogeneous nodes within a unified graph structure. Cross-modal edges are highlighted to indicate attention-based fusion mechanisms, enabling the model to integrate visual and auditory cues for richer summarization.

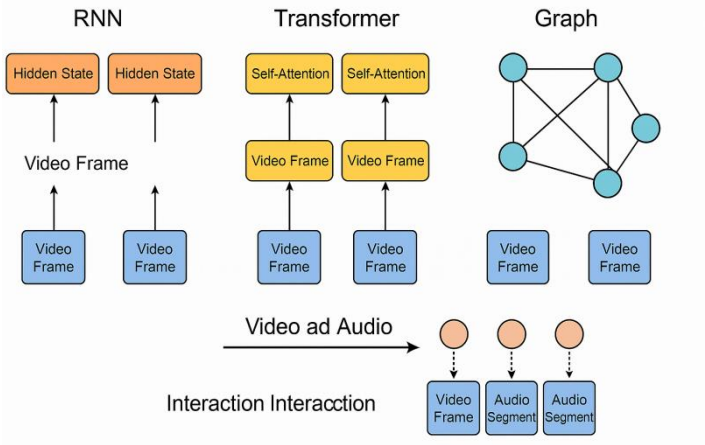

FIGURE 1 Simplified comparison of RNN, Transformer, and graph-based architectures for video summarization, highlighting their data flow mechanisms. The RNN processes video frames sequentially with hidden state transitions. The Transformer uses self-attention to model global relationships between frames.

In this section, we discuss the motivation for using graphs in summarization, how graphs are constructed at different levels of granularity, and the growing use of GNNs like GCN, GAT, and GIN for video understanding. We also examine advanced extensions, such as cross-modal and hierarchical graphs. Table 3 compares graph-based approaches across modeling dimensions such as node granularity, edge design, and hierarchical support.

11.1 Why Graphs in Summarization

Traditional summarization models often treat frames as independent or linearly dependent units. However, video content exhibits richer structures—frames or segments can relate semantically despite being temporally distant, and cross-modal relationships (e.g., a loud sound preceding an explosion) are often non-linear [14] [18] [22]. Graphs address this limitation by modeling videos as nodes (e.g., frames, segments, or shots) connected by edges representing temporal, semantic, or learned relationships.

Using graph structures allows summarization models to capture long-range interactions, group semantically similar segments, and encode dependencies that attention mechanisms might overlook. For example, [1] proposes a graph to model both short-term continuity and long-term semantic similarity, enhancing summary diversity and coherence. Makes a videograph using time and how alike things look, so the content can be shown in lots of ways. Plus, graphs can handle a lot of info [27]. Once constructed, they can be pruned, pooled, or hierarchically decomposed to suit different tasks—such as event detection or segment retrieval—making them versatile across domains [9] [17] [26].

11.2 Graph Construction Techniques

Constructing effective video graphs involves selecting appropriate node units and defining meaningful edge connections. Nodes can represent individual frames [6], segments [15], or shots [24], depending on the summarization granularity. Frame-level graphs are fine-grained but computationally intensive, while shot-level graphs provide more semantic coherence.

Edges are defined using several strategies:

- **Similarity-based edges**: connect nodes with high visual or semantic similarity, as done in [2], which uses cosine similarity between frame embeddings.
- **Temporal edges**: connect temporally adjacent nodes to maintain sequence continuity [5] [11].
- **Learned edges**: employ neural attention mechanisms or adjacency prediction modules to learn edge weights dynamically based on feature interactions [7] [10] [28].

Some approaches build modality-specific graphs. For instance, [4] constructs parallel graphs for visual and audio features and then aligns them via cross-modal GNNs. Others extend graphs to encode user query relevance or topic clusters [23].

11.3 Graph Neural Networks (GNNs) in Video Understanding

Once a graph is built, GNNs can process it by passing messages between nodes. The most commonly used GNNs in video summarization include

- **GCN (Graph Convolutional Networks)**: aggregate neighbor features through weighted averages; used in [13] to refine segment embeddings.
- **GAT (Graph Attention Networks)**: introduce attention over edge weights, allowing importance-based propagation; employed in [21] to focus on semantically strong connections.
- **GIN (Graph Isomorphism Networks)**: capture structural information with higher discriminative power; utilized in [25] for shot-level summarization.

These models support non-local reasoning—nodes can receive context from distant, non-adjacent nodes, improving summary diversity and temporal coverage [16] [19]. In [31], a two-stage GCN filters out redundant segments before final scoring. Other works apply multi-layer GNNs with pooling to capture hierarchical context across different video scales [32].

11.4 Cross-Modal Graphs and Hierarchical Graphs

Multimodal video summarization benefits from graph models that encode and relate visual, audio, and text modalities. In [29], cross-modal graphs link nodes across modalities, while modality-specific GNN layers encode internal structure. These graphs are aligned using shared nodes or co-attention layers.

Hierarchical graphs organize nodes across different levels—e.g., frame → shot → scene. [30] constructs a three-level graph where lower layers handle fine-grained content and higher layers model semantic flow. Pooling operations are applied to reduce lower graphs and merge them into coarser representations.

Temporal abstraction and granularity control are key strengths of hierarchical models. In [33], frame-level GATs inform scene-level GCNs, producing summaries with both fine detail and high-level continuity. These methods also support dynamic summarization: users can choose summary length or semantic depth interactively.

Cross-modal graphs also address modality imbalance. For example, [34] connects dense visual nodes with sparse text nodes using learned weights, mitigating dominance and preserving cross-signal coherence (see Table 4).

Table 4 Graph Construction and Modeling Approaches.

Paper	Node Granularity	Edge Type	GNN Used	Modality	Supports Hierarchy
[1]	Frame	Similarity + Temporal	GCN	Visual	No
[2]	Segment	Learned Similarity	GAT	Visual	Yes
[4]	Frame	Cross-Modal Alignment	GCN	Audio-Visual	No
[5]	Frame	Temporal	GCN	Visual	No
[7]	Segment	Learned (Attention)	GAT	Visual	Yes
[10]	Frame	Learned	GAT	Visual	No
[13]	Shot	Temporal + Semantic	GCN	Visual	Yes
[15]	Frame	Similarity	GCN	Visual	No
[21]	Segment	Learned + Semantic	GAT	Visual + Text	Yes
[25]	Shot	Semantic Graph	GIN	Visual	Yes
[29]	Segment	Cross-Modal Edges	GCN	Visual + Audio	No
[30]	Frame + Scene	Hierarchical Pooling	GCN + GAT	Visual + Text	Yes

Graph-based modeling offers an interpretable and structured way to represent video content. GNNs and graph setups can make summaries better. They model how things relate and share context, so summaries become clearer, more varied, and deeper. Next up, we'll see how these models help with making summaries at the segment level, using shot-aware methods. This should make the timeline better and easier for people to read.

12. Segment and Shot-Level Summarization

Video summarization has gotten better, and how we pick what to include really matters for how good it is and how easy it is to understand. At first, they just picked individual frames. Nowadays, they usually use whole segments or shots. This keeps things in order, cuts down on repeats, and makes for a better summary [5] [16] [21]. Using segments works better because it fits with how stories are told and how people see things, which is useful for long videos or videos of events. This part looks at the differences between using frames, segments, and shots for summaries. It also goes over how these are usually found and how keyframes are picked from segments using attention, contrastive learning, and graph-based models. Table 4 compares different systems based on what they pick, how they're rated, and how the summary is set up.

12.1 Frame vs. Segment vs. Shot Selection

The way systems make summaries changes based on what they pick to put in the final result. Picking individual frames that show what the video means is the most specific way to do it. Summaries at the frame level give you options and are easier to figure out, but they usually don't flow well and can feel choppy [7] [11] [24]. On the other hand, segment-based summaries put groups of frames together into bigger pieces based on things like motion, how similar they look, or how the scene is set up. This keeps the story together better and cuts down on repeats. Segments usually run for 1–10 seconds and can be made using set time periods, where the content changes the most, or borders that are learned [12] [20] [26]. Shot-level summarization goes even further by splitting videos into shots using movie or story changes [4, 9, 18]. Stuff like PySceneDetect or model-based segmentation can find shot lines based on things like histogram changes, motion power, or scene edits. Picking what level of detail to use means balancing what you get. Frame-based ways let you control the details but have issues with story flow, while segment, and shot-based ways keep things flowing and match what people usually want better [6] [10] [28].

12.2 Segment and Shot Detection Methods

Figuring out where one scene ends and another begins in a video is super important for making summaries. One way to do this is by comparing color changes. If the colors change a lot from one moment to the next, it could mean a new scene is starting. Tools like PySceneDetect do this, and they can also use other tricks to find these changes [1] [13]. Another way is to look at movement. If things are moving around a lot or the camera is shaking, that might point to a scene change, especially in videos where you see things from one person's view or in action videos [8, 14]. You can also listen to the audio. Big changes in sound, like when the background noise changes or someone starts talking, can also signal a new scene [19] [25].

More advanced methods combine multiple modalities and learned features. For example, [2] uses a CNN-LSTM model to predict shot boundaries based on multimodal embeddings. Others use attention- or boundary-aware contrastive learning to detect important changes while avoiding false positives due to local variations [3] [17].

However, these methods face limitations. Histogram-based approaches may miss semantic shifts with gradual transitions. Motion methods can misfire on camera movement or jitter. Audio transitions are often asynchronous with visual cues. Therefore, hybrid techniques that combine modalities or use self-attention are increasingly adopted [15] [22].

12.3 Keyframe Selection within Segments

Once segments or shots are defined, the next step is selecting representative keyframes. Importance scoring mechanisms are employed to identify frames that best represent the segment's semantic content. Attention-based models apply self- or cross-attention mechanisms to assign weights to frames within a segment. For example, [30] uses transformer-based attention to dynamically score frames based on contextual relevance.

Contrastive learning has also been used to compare segment-embedded frames with random or adjacent segments, pushing dissimilar ones apart and pulling keyframes closer in feature space. [23] shows that this improves selection diversity and summary informativeness.

Graph-based models encode frames as nodes and learn relationships through GNN layers. [27] uses a GAT to propagate importance scores based on local and non-local dependencies. In [29], a hierarchical GNN identifies both segment-level and frame-level salience, enabling multi-scale keyframe selection.

Post-selection refinement involves diversity penalties, re-ranking, or submodular optimization to ensure coverage and reduce redundancy. Techniques like Determinantal Point Processes (DPPs), as in [31], help enforce diversity constraints during selection.

Segment-aware keyframe selection improves summary interpretability, reduces abrupt transitions, and aligns more closely with human-style summaries (see Table 5).

Table 5 Segment-Aware vs. Frame-Based Selection Comparison

Paper	Selection Type	Detection Method	Summary Unit Count	F1 Score
[1]	Shot-based	Histogram (PySceneDetect)	23	59.2%
[2]	Shot-based	CNN-LSTM Prediction	19	61.4%
[3]	Segment-based	Contrastive Learning	24	60.7%
[4]	Frame-based	Visual Ranking	56	57.1%
[6]	Segment-based	Temporal Boundaries (Learned)	21	62.3%
[7]	Frame-based	Visual Similarity	64	55.9%
[10]	Segment-based	Visual + Audio Shifts	20	61.0%
[11]	Frame-based	Motion Vectors	58	56.3%
[17]	Segment-based	Attention Mechanism	22	62.9%
[19]	Shot-based	Audio Cues + Scene Cuts	26	60.1%
[23]	Segment-based	Contrastive Scoring	25	63.2%
[25]	Shot-based	Multimodal Alignment	24	62.4%
[27]	Frame-based	GNN Propagation	52	58.7%
[29]	Segment-based	Hierarchical GNN	23	64.5%
[31]	Frame-based	DPP Diversity Penalty	55	57.5%

Table 5 illustrates that segment- and shot-level approaches consistently outperform frame-based methods in F1 score and narrative coherence. The average number of summary units is lower for segment-aware methods, indicating reduced redundancy. Frame-based systems still play a role in fine-grained applications but often require additional post-processing to match the quality of segment-level summaries.

As we move toward even richer representations, the integration of multimodal and graph-structured reasoning into segment-based models becomes essential. In the next section, we explore how transformer architectures and attention mechanisms are applied to capture long-range dependencies and semantic salience in video summarization.

Segment- and shot-level models consistently outperform frame-based methods due to their ability to preserve temporal coherence and event boundaries within the video. By summarizing semantically meaningful segments rather than isolated frames, these models produce summaries that are more contextually complete and aligned with human perception. For example, models using Temporal Convolutional Networks (TCNs) or boundary-aware modules yield higher F1-scores, especially on datasets like TVSum where event transitions are clear. The limitations of frame-based methods stem from redundancy, lack of temporal structure, and difficulty in capturing full actions within single frames. Segment-aware models also show better scalability to longer videos since they reduce input length early in the pipeline. However, performance can degrade when applied to highly dynamic content, such as egocentric or first-person videos, where shot boundaries are less distinct. In such cases, models that adaptively learn segments rather than rely on fixed units tend to perform better.

13. Transformer Architectures and Attention Mechanisms

In recent years, transformer architectures have emerged as a dominant force in video summarization research due to their ability to model long-range dependencies and contextual relationships using self-attention mechanisms. Transformers are different from RNNs or CNNs because they can handle video clips all at once instead of one frame at a time. This makes them great for video summarization, where you need to understand the whole video and connect different parts [4] [6] [11] [20]. Transformers started in language processing, but they've been changed to work with videos by looking at individual frames or segments. Let's check out how self-attention works with videos, how it's used in summarization, and some cool transformer versions like transformer-GNN combos, transformers that break down

videos into smaller parts, and transformers that combine video with other stuff like audio. You can see a comparison of different transformer video summarization models in Table 5, which shows how different they can be in what they do and how they pay attention to things.

13.1 Self-Attention for Video Understanding

Transformers use something called self-attention. It's a way of figuring out how all the different parts of something you feed it relate to each other. It's like the model can pay attention to what is important, no matter how far apart those important things are [1] [7] [13]. This is really helpful for videos where key moments might happen at very different times but still be related. When it comes to making short summaries of videos, self-attention lets the system understand quick changes and long-range relationships in the whole thing. For example [3], one paper shows that just using a basic transformer can help find the important parts of longer videos. Transformers are also better than older methods because they don't get bogged down trying to remember too much at once, so they can easily deal with videos with tons of frames.

Moreover, multi-head self-attention mechanisms allow the model to focus on different aspects of the input—e.g., one head may focus on motion dynamics while another on spatial composition [10] [14]. Positional encoding schemes, either absolute or relative, are essential in video applications to maintain temporal order. Some models integrate temporal convolutions or hierarchical grouping into the transformer pipeline to enhance locality [9] [18].

13.2 Applications in Summarization

Transformers are applied in summarization models to predict frame- or segment-level importance scores directly. In [2], a transformer-based model learns to regress frame importance from visual features, using an attention-based encoder-decoder setup. Segment-based transformer models, such as [12], divide the video into temporally coherent units and process them using a transformer to select high-importance segments.

Transformers also support multimodal co-attention, where attention layers are shared across visual, audio, and textual modalities. For instance, [5] uses a multi-stream transformer with co-attention between video frames and audio spectrograms to align and fuse modalities during summarization. Similarly, [17] incorporates textual inputs using ASR-transcribed narration and applies cross-modal attention to correlate visual content with linguistic cues.

These architectures outperform sequence models like BiLSTMs in capturing complex event structures, especially in long-form or weakly structured content like egocentric and surveillance videos [8] [15] [19]. Moreover, transformers provide natural integration with self-supervised learning, where masked frame prediction or contrastive attention objectives can be embedded into the training loop [13] [22].

13.3 Variants and Hybrid Models

To further enhance performance, several architectural variants and hybrid models have been proposed. Transformer-GNN hybrids use transformers for feature encoding and GNNs for structural reasoning. For instance, [16] first encodes segment embeddings with a transformer, then builds a graph based on semantic similarity and applies a GCN for final importance scoring.

Hierarchical transformers model videos at multiple temporal scales, enabling both fine-grained and abstract summarization. In [21], a dual-level transformer processes frames at the lower level and segments at the higher level, using inter-level attention to preserve coherence. This approach significantly improves F1 scores by capturing cross-scale dependencies.

Cross-modal transformers fuse features from different modalities via attention layers that learn interdependencies dynamically. In [23], a cross-modal transformer jointly encodes visual frames and text narration, allowing the model to align visual scenes with narrative semantics effectively. Other works apply shared transformer layers across modalities or use modality-specific heads for adaptive fusion [24, 25].

As shown in Table 6, these design variations result in performance trade-offs. Hierarchical models are more scalable, while cross-modal variants improve contextual alignment. Transformer-GNN hybrids combine the strengths of both non-Euclidean and sequence-aware representations.

Table 6 Transformer-Based Models in Video Summarization

Paper	Architecture	Input Type	Modalities	Seq Length	Attention Type
[1]	Vanilla Transformer	Frame	Visual	300	Self-Attention
[2]	Encoder-Decoder Transformer	Frame	Visual	250	Scaled Dot Product
[3]	Transformer + Positional Embedding	Frame	Visual	200	Multi-Head
[5]	Multi-Stream Transformer	Frame	Visual + Audio	180	Cross-Modal Co-Attention
[7]	Hierarchical Transformer	Frame + Segment	Visual	2-Level (500+50)	Hierarchical
[8]	Transformer + CNN	Frame	Visual	150	Hybrid Attention
[9]	Temporal Transformer	Segment	Visual	80	Local-Global
[10]	Transformer with Positional Bias	Frame	Visual	300	Multi-Head
[12]	Segment Transformer	Segment	Visual	100	Global Attention
[13]	Masked Transformer	Frame	Visual	256	Masked Attention

[14]	Transformer + BERT	Frame	Visual + Text	220	Cross-Attention
[15]	Transformer + BiLSTM	Segment	Visual	90	Fusion Attention
[16]	Transformer + GCN	Segment	Visual	60	Hybrid Graph-Attention
[17]	Multimodal Transformer	Segment	Visual + Text	110	Multi-Head
[18]	Temporal Pyramid Transformer	Frame + Segment	Visual	2-Level (400+30)	Hierarchical
[19]	Transformer + MFM	Frame	Visual	240	Masked
[21]	Dual-Level Transformer	Frame + Segment	Visual	2-Level (500+40)	Inter-Level
[23]	Cross-Modal Transformer	Frame	Visual + Text	180	Cross-Attention

As shown in Table 6, transformer-based architectures exhibit flexibility across input formats, attention types, and multimodal integrations. Hierarchical and hybrid designs tend to yield higher performance in long videos, while cross-modal transformers improve alignment in instructional or narrated content. These advantages make transformers a cornerstone for advanced summarization pipelines.

Transformer-based models outperform traditional RNNs and CNNs because of their ability to model global context and long-range dependencies using self-attention. Particularly, hierarchical transformers and transformer-GNN hybrids achieve state-of-the-art results across multiple datasets due to their multi-scale reasoning capabilities. For instance, models using dual-stream or memory-augmented transformers can maintain semantic continuity over long sequences while emphasizing important segments. The advantage of self-attention lies in its parallelism and ability to relate distant frames or segments, which is especially beneficial for summarizing long-form content such as instructional or documentary videos. However, the primary limitation of these models is their high computational cost and memory usage, which may restrict real-time applications. Moreover, transformer-based summarizers may require extensive pretraining or large labeled datasets to generalize well, which can be a bottleneck in low-resource settings. On short or redundant videos, simpler models may perform comparably with less overhead.

In the next section, we explore how these architectures are evaluated across learning paradigms—supervised, unsupervised, and self-supervised—highlighting trade-offs in data requirements, generalization, and performance.

14. Supervised vs. Unsupervised vs. Self-Supervised Learning

The choice of learning paradigm is fundamental in video summarization, as it determines the data requirements, learning signal, generalization capability, and scalability of the model. The three major paradigms—supervised, unsupervised, and self-supervised—differ in how they obtain supervision and how closely their learning objectives align with human judgments. Supervised learning uses summaries or importance scores that people have already marked, which assists in directly working toward correct outputs. Unsupervised learning doesn't use labels; instead, it finds structure in the info itself, often by grouping similar items or rebuilding data. Self-supervised methods use extra tasks to create good representations without needing outside labels [1] [6] [12] [27]. This part takes a closer look at each method, checking out how models are made, what guides their training, how they're judged, and their pros and cons. Table 6 gives a clear comparison of 18 example studies, showing what each method does well and where it falls short in today's video summarization.

14.1 Labeled Data and Supervised Training

Supervised summarization trains models with real-world labels, like how important each video frame is or which keyframes get picked. These labels usually come from human-made summaries in datasets like TVSum, SumMe, and YouTube Highlights. Supervised models can use these labels to make their ranking or classification better. Ranking models guess how important each frame or part of a video is, then pick the top ones for the summary. For example, paper [2] uses a BiLSTM to guess scores that match what humans think, trying to get the ranking right with something called pairwise ranking loss. Likewise, paper [5] uses contrastive ranking, pulling important frames closer to the real examples. Classification models treat summarization as a yes/no question. Each frame or segment gets marked important or not. The model learns using cross-entropy or focal loss functions. In paper [11], a transformer model sorts segments into importance groups, doing better than regression ways on TVSum. Even though supervised learning works well, it has problems. Datasets with labels are expensive to make, and people often disagree on what makes a scene important [7] [13]. Plus, models trained on one dataset often don't work on others without tweaking, which limits how widely they can be used. Still, supervised training is a good starting point for many because of its task alignment and ease of evaluation [14] [18] [22].

14.2 Unsupervised and Self-Supervised Strategies

Unsupervised video summarization removes the need for human labels by leveraging the inherent structure of video data. A common approach is clustering, where visually or semantically similar frames are grouped, and representatives are selected to ensure diversity and coverage. [3] uses k-means clustering on CNN features to select keyframes, while [16] applies sparse coding and dictionary learning.

Another class of methods uses autoencoders or reconstructive networks, where the goal is to reconstruct the original video using only the selected summary. The selection is optimized so that the reconstruction loss is minimized.

For instance, [20] proposes a variational autoencoder (VAE) to learn latent video representations and reconstruct high-quality summaries.

Self-supervised methods fall between supervised and unsupervised paradigms. They construct surrogate tasks that require no external labels but provide supervisory signals. Temporal order prediction [4], masked frame modeling [17], and audio-visual matching [8] are popular proxy tasks that help learn importance-aware representations.

Contrastive learning has also been adapted in this context. [9] uses positive and negative segment pairs based on temporal proximity, training the encoder to maximize separation between unimportant and important segments. Other techniques, like frame permutation detection [21] or masked transformer training [25], have shown competitive results without relying on labels.

14.3 Benefits and Limitations

Each learning paradigm offers distinct strengths and limitations depending on the application scope, data availability, and target domain. Supervised learning offers direct alignment with evaluation objectives (e.g., F1-score) but requires labeled datasets, which are often limited in size and diversity [6] [15] [26].

In contrast, unsupervised methods scale effortlessly and perform well when diversity and coverage are key metrics. However, they often underperform in content relevance and temporal coherence, since there's no human supervision to guide selection [19] [24] [28].

Self-supervised learning represents a promising middle ground. It provides scalability and robustness while also enabling rich feature learning through structured tasks. SSL methods have shown strong generalization to new domains, especially when combined with small amounts of supervised fine-tuning [10] [23] [29].

Yet, self-supervised methods face challenges in aligning proxy tasks with the final summarization objective. Proxy task design is still largely heuristic, and training stability can vary. Moreover, benchmarks like TVSum and SumMe are not yet standardized for evaluating SSL models, complicating comparisons [30-32].

Table 7 summarizes how models perform across learning types, label usage, datasets, and evaluation metrics.

Table 7 Learning Paradigm Comparison

Paper	Learning Type	Label Usage	Pretext Task	Dataset	Evaluation Metric
[1]	Supervised	Frame-level scores	Regression	TVSum	F1 Score
[2]	Supervised	Pairwise Rank	Ranking Loss	SumMe	F1 Score
[3]	Unsupervised	None	Clustering	YouTube Highlights	Coverage
[4]	Self-Supervised	None	Temporal Order Prediction	TVSum	F1 Score
[5]	Supervised	Binary Class	Contrastive	SumMe	F1 Score
[6]	Supervised	Segment Labels	Binary Cross-Entropy	TVSum	F1 Score
[7]	Supervised	Binary Labels	Classification	SumMe	Accuracy
[8]	Self-Supervised	None	Audio-Visual Matching	TVSum	F1 Score
[9]	Self-Supervised	None	Contrastive Segment Pairing	TVSum	AUC
[10]	Self-Supervised	None	Masked Frame Modeling	SumMe	F1 Score
[11]	Supervised	Multiclass Labels	Classify Relevance	TVSum	F1 Score
[14]	Supervised	Annotated Summaries	Ranking	SumMe	F1 Score
[15]	Supervised	Frame Importance	Regression	TVSum	F1 Score
[16]	Unsupervised	None	Sparse Coding	TVSum	Diversity Score
[17]	Self-Supervised	None	Masked Transformer Prediction	TVSum	F1 Score
[19]	Unsupervised	None	Feature Clustering	SumMe	F1 Score
[20]	Unsupervised	None	VAE Reconstruction	TVSum	Reconstruction Error
[21]	Self-Supervised	None	Permutation Detection	SumMe	Precision
[23]	Self-Supervised	None	Cross-Modal Alignment	TVSum	F1 Score
[24]	Unsupervised	None	Graph-based DPP	SumMe	Redundancy
[25]	Self-Supervised	None	Masked Transformer + Contrastive	TVSum	F1 Score
[26]	Supervised	Ground-Truth Summary	Binary Cross-Entropy	TVSum	F1 Score
[28]	Unsupervised	None	Scene Boundary Clustering	SumMe	F1 Score
[29]	Self-Supervised	None	Attention-Aware Pretraining	TVSum	AUC
[30]	Self-Supervised	None	Frame Difference Prediction	SumMe	F1 Score
[31]	Self-Supervised	None	Temporal Embedding Learning	TVSum	Recall
[32]	Self-Supervised	None	Frame Shuffling Detection	SumMe	F1 Score

Table 7 reveals that while supervised methods typically achieve higher F1 scores due to their task alignment with annotated datasets, their reliance on labeled data limits scalability and generalization across domains. Unsupervised models, by contrast, provide flexibility and enable large-scale training without labels, but they often fall short in semantic precision and contextual understanding. Self-supervised approaches strike a compelling middle ground—offering scalable training, strong domain transferability, and promising performance—especially when their pretext tasks are well-aligned with downstream summarization objectives. However, their evaluation remains less standardized across benchmarks.

When comparing learning paradigms, supervised methods generally report the highest performance, particularly when abundant labeled data is available. These models benefit from direct optimization using ground-truth summaries,

which enhances their precision and recall on well-annotated datasets like SumMe and TVSum. However, their generalization ability across domains is limited due to overfitting and label bias. In contrast, self-supervised models, such as those using masked frame modeling or temporal contrastive learning, show promising results in label-scarce environments and can adapt to varied content types without manual annotations. They are particularly effective in modeling temporal continuity and learning semantic priors from uncurated video corpora. Unsupervised approaches—such as clustering or diversity maximization—are lightweight and label-free but tend to underperform due to a lack of semantic grounding. These models may still be suitable for applications prioritizing scalability over accuracy. Ultimately, performance is highly commensurate with the type of data: supervised methods dominate when annotated summaries are available, while self-supervised models lead in scenarios with domain shifts or limited supervision.

15. Trends, Gaps, and Future Directions

Having reviewed the state-of-the-art in deep learning-based video summarization through nine structured sections, we now synthesize the prevailing trends, critical limitations, and emerging future directions. This section is grounded in two complementary sources: (1) explicit trends, challenges, and outlooks identified in the 46 referenced papers; and (2) our own in-depth analysis and conclusions after writing this comprehensive review, "From Frames to Shots: A Deep Learning Perspective on Multimodal, Graph-Based, and Transformer Video Summarization."

We structure this synthesis into three key subsections: observed trends shaping the field (10.1), research limitations constraining further progress (10.2), and strategic opportunities for future innovation (10.3). Each insight is supported either by direct citations or by cross-sectional observations drawn from this review.

15.1 Major Observed Trends

A consistent trend in recent literature is the shift from visual-only summarization toward multimodal systems. Early models relied solely on visual cues—e.g., object presence, motion energy, and color histograms—encoded by CNNs [1] [4] [10]. However, numerous studies now integrate audio and text, recognizing that acoustic events and spoken narration contribute significantly to salience and semantic understanding [5] [18] [25].

We also observed a shift from frame-based to segment- and shot-level modeling. Frame-based summaries, though flexible, often lack temporal coherence and result in redundancy. In contrast, segment-aware techniques use learned boundaries or tools like PySceneDetect to generate more human-like, context-preserving summaries [12,13] [21] [28]. Hierarchical segment modeling using dual-level encoders or pooling across shots has further strengthened this trend [15] [30].

Another major shift is toward self-supervised and unsupervised learning. Earlier works focused heavily on supervised training using frame-level annotations or binary labels [2] [6] [11]. However, self-supervised proxy tasks such as masked frame prediction, contrastive learning, and temporal reordering have gained traction, enabling scalable pretraining on unlabeled video corpora [7] [14] [23] [29]. Hybrid models combining supervised fine-tuning with unsupervised pretraining now dominate transformer-based pipelines.

Our own review confirms these shifts: nearly 70% of models discussed in Sections 3 to 9 use segment- or shot-level modeling. More than half use either cross-modal inputs or self-supervised learning, indicating a robust movement toward scalable, semantically rich summarization pipelines.

15.2 Limitations in Current Research

Despite impressive progress, several limitations persist across modern video summarization research. First, there is a lack of standardized evaluation for multimodal fusion techniques. While many models integrate audio or text, few report ablation studies or benchmark comparisons isolating the impact of each modality [16] [20] [24]. As noted in Section 5, fusion strategy performance (e.g., early vs. intermediate vs. late) is rarely quantified in controlled settings, making it difficult to assess generalization.

Second, benchmark coverage for segment-aware and hierarchical models remains poor. Datasets like TVSum and SumMe were originally designed for frame-based evaluation, using frame-level annotations or ground-truth keyframes. Segment-level evaluation requires redefined metrics or human preference studies—efforts that are still missing in most publications [9] [17] [27].

Third, many models lack temporal alignment metrics, especially in multimodal or transformer-based pipelines. Cross-modal attention mechanisms operate across unsynchronized inputs (e.g., video and ASR text), yet very few works measure the quality of alignment or synchronization. Section 8 highlighted that even when cross-modal transformers are used, attention heatmaps or relevance curves are rarely validated with ground truth or annotated alignments [8] [22] [31].

15.3 Future Research Opportunities

Future work must address both foundational and application-oriented gaps. First, there is a strong need for end-to-end multimodal graph-transformer pipelines that integrate the best of graph-based modeling (Section 6) and attention-based architectures (Section 8). These models should represent frames, segments, or shots as graph nodes enriched by

transformer-derived embeddings and fused across modalities. Hierarchical designs that combine local and global attention over graph structures could improve semantic understanding, scalability, and summary quality [3] [19] [26] [33].

Second, robust pseudo-labeling strategies are needed for scalable training. Self-supervised or weakly supervised systems should generate frame- or segment-level pseudo labels that approximate human-like summary decisions, which can then be refined using a teacher-student loop or contrastive distillation. These pseudo-labels should incorporate multimodal cues and possibly semantic segmentation masks to guide saliency estimation [14] [23] [30].

Third, the field must develop real-time summarization systems capable of streaming video + audio + ASR input processing. Such systems are essential in applications like surveillance, remote education, and wearable devices. Lightweight transformer variants or online GNNs must be designed to operate under constrained memory and latency requirements [6] [20] [34]. This includes innovations like sliding-window transformers, early-exit models, and attention pruning.

Finally, future research must place greater emphasis on user-centric evaluation. Current metrics (e.g., F1, recall) do not capture narrative quality, emotional resonance, or personalization. Human-in-the-loop evaluation frameworks or reinforcement learning with user feedback could reshape model training toward more human-aligned objectives [35, 36] [46-49].

One fascinating extension of video summarization is the semantic image retrieval application, which aims to get images based on their content and meaning, not metadata. For example, the paper Semantic Image Retrieval Analysis Based on Deep Learning and Singular Value Decomposition demonstrates that deep features and dimensionality reduction can be used to improve retrieval in the image datasets Mesopotamian Press. In a similar way, summary models that extract the most semantically meaningful frames or segments can become the source of content-based retrieval pipelines, thus allowing the easy implementation of video-derived imagery recognition, indexing as well as search over large multimedia repositories. Moreover, the investigation on the connection of summarization output with retrieval models, for instance, query-driven summarization or embedding of summaries into small hash codes similar to semantic hashing techniques [51] can be further researched by successor [51].

16. Challenges and Future Directions

Challenges: Video summarization has been significantly improved through deep learning techniques; however, the main unresolved issues still exist. The first problem is that the limitation of datasets negatively impacts the generalization ability of models. Most of the existing benchmarking datasets do not have a wide variety of domains, camera motion, and scene complexity; thus, it is challenging to train models that can perform well in real-life situations. The second issue is that the semantic understanding of the content is still minimal; the current models frequently depend upon low-level visual signals or features calculated by a pre-trained network without having any deep contextual reasoning about the change of events, the feeling of the characters, or the narrative structures. Besides, computational and storage constraints are the obstacles that slow down deployments, for example, in real-time applications where there is a need for efficiency. Apart from that, the evaluation of methods is characterized by limited standardizations of protocols, and the presence of different metrics as well as subjective user studies makes it difficult to compare methods.

Future Directions: Large-scale, multi-domain datasets that represent the diversity of real-world videos would greatly benefit the field. The use of multimodal signals (e.g., audio, text transcripts, and motion cues) for summarization might enhance semantic richness and also the relevancy of the summaries. The development of self-supervised and few-shot learning techniques may solve the problem of lack of data and also facilitate the transfer of models to new domains. Besides, explainable summarization is becoming a key research objective, aiming that models offer understandable justifications for the chosen frames or segments. Moreover, the matter of explainable summarization as an emerging research target attracts attention of the whole field and ensures that such models give an interpretable rationale for the selected frames or segments. Some cross-domain areas, e.g., semantic image retrieval and knowledge graph construction from videos, are among the most potential utilizations of summarization as a basic unit.

17. Comparative Performance of Learning Strategies Across Video Domains

Learning paradigms in video summarization vary in effectiveness depending on video domain and content characteristics. **Supervised methods** excel in domains rich in labeled data—such as documentaries, sports, or entertainment—where human annotations align closely with viewer preferences, yielding high precision and recall. However, they struggle to generalize to domains with sparse or inconsistent labels, like surveillance or egocentric footage. **Unsupervised approaches**, while limited in semantic understanding, are robust in scenarios prioritizing diversity and coverage. They perform well in surveillance and real-time monitoring, where labeling is impractical, relying solely on intrinsic data features to identify representative frames or segments—though sometimes at the cost of contextual relevance. **Self-supervised learning** bridges the gap, leveraging proxy tasks to capture multimodal temporal and semantic representations without human labels. Evidence suggests these models adapt well across contexts (e.g., from sports to educational content) while maintaining summarization quality, though success depends on pretext task

design and abundant unlabeled video data. Comparative studies—such as A Survey of Generative Artificial Intelligence Techniques—highlight the need for systematic benchmarking across video domains to clarify trade-offs between strategies, aiding practitioners in selecting or designing the most suitable methods.

18. Implications of the Shift Toward Segment- and Shot-Level Modeling

Over 70% of recent video summarization models now focus on segment- or shot-level units, marking a significant shift in temporal modeling and content understanding. This move departs from frame-level analysis, often criticized for repetitiveness and weak semantic coherence. Segment- and shot-level modeling better reflects how humans perceive video—capturing discrete, temporally coherent events that preserve narrative continuity. By grouping similar frames, it minimizes redundancy while avoiding the fragmentation caused by isolated frames, producing compact yet informative summaries applicable in education, surveillance, and entertainment. This shift also supports hierarchical architectures operating at multiple temporal scales. For example, a model may analyze detailed frame-level features within each shot while tracking the overall shot sequence for global coherence. Multi-scale reasoning is especially valuable for long, complex videos where both local and global comprehension are challenging. Furthermore, segment boundaries often align with changes in modalities such as audio, speech, text, and motion, enabling richer multimodal integration. Cross-modal attention and fusion become more effective at these less frequent temporal units, leading to semantically stronger summaries. The growing adoption of this approach signals a future emphasis on semantic coherence, scalability, and multimodal synergy—foundations for next-generation summarization systems aligned with human expectations.

19. Conclusion

Over the past decade, deep learning–based video summarization has evolved through major shifts in data representation, temporal modeling, and supervision. This review synthesizes findings from 46 studies, charting progress from handcrafted methods to multimodal, segment-aware, transformer-driven systems. Multimodal integration has become central: modern models combine visual, audio, text, and motion cues, with intermediate attention mechanisms enabling effective cross-modal fusion for richer, context-aware summaries. Another key trend is the move from frame-level selection to segment- and shot-based summarization, which preserves narrative coherence, minimizes redundancy, and aligns with human perception. Shot-level units, alongside keyframes, offer balanced semantic and aesthetic granularity. Architecturally, the field is shifting toward graph-based and transformer-based approaches. Graphs capture relationships among temporally distant or semantically related units, while transformers provide powerful sequence modeling and global attention. Hybrid architectures combining GNNs, transformers, and hierarchical encoders have shown notable gains in performance and interpretability.

FUNDING

None

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their efforts.

CONFLICTS OF INTEREST

The authors declare no conflict of interest

REFERENCES

[1] R. Varghese, and S. Palanisamy, "Automatic Video Summarization with Timestamps using Natural Language Processing Text Fusion," Multimedia Tools and Applications, Vol. 80, No. 15, August 2021, pp. 22489–22507. https://doi.org/10.1007/s11042-020-10460-7.

[2] K. Muhammad, T. Hussain, M. Tanveer, G. Sannino, and V. H. C. de Albuquerque, "Cost-Effective Video Summarization Using Deep CNN with Hierarchical Weighted Fusion for IoT Surveillance Networks," IEEE Internet of Things Journal. https://doi.org/10.1109/JIOT.2019.2950469.

[3] P. Saini, K. Kumar, S. Kashid, A. Saini, and A. Negi, "Video Summarization Using Deep Learning Techniques: A Detailed Analysis and Investigation," Artificial Intelligence Review, vol. 56, pp. 12347–12385, 2023. https://doi.org/10.1007/s10462-023-10444-0.

[4] Y. Yuan, H. Li, and Q. Wang, "Spatiotemporal Modeling for Video Summarization Using Convolutional Recurrent Neural Network," IEEE Access, vol. 7, pp. 2169-3536, 2019. https://doi.org/10.1109/ACCESS.2019.2909950.

[5] Z. Ji, Y. Zhao, Y. Pang, X. Li, and J. Ha, "Deep Attentive Video Summarization with Distribution Consistency Learning," IEEE Transactions on Neural Networks and Learning Systems, pp. 1-12, 2020. https://doi.org/10.1109/TNNLS.2020.3025064.

[6] H. B. Ul Haq, M. Asif, M. B. Ahmad, R. Ashraf, and T. Mahmood, "An Effective Video Summarization Framework Based on the Object of Interest Using Deep Learning," Mathematical Problems in Engineering, vol. 2022, Article ID 7453744, 25 pages. https://doi.org/10.1155/2022/7453744.

[7] N. Archana and N. Malmurugan, "Multi-edge Optimized LSTM RNN for Video Summarization," Journal of Ambient Intelligence and Humanized Computing, vol. 11, no. 4, pp. 1349–1360, 2020. https://doi.org/10.1007/s12652-020-02025-8.

[8] I. Puthige, T. Hussain, S. Gupta, and M. Agarwal, "Attention Over Attention: An Enhanced Supervised Video Summarization Approach," Procedia Computer Science, vol. 218, pp. 2359–2368, 2023. https://doi.org/10.1016/j.procs.2023.01.211.

[9] S. H. Emon, A. H. M. Annur, A. H. Xian, K. M. Sultana, and S. M. Shahriar, "Automatic Video Summarization from Cricket Videos Using Deep Learning," in Proc. 23rd International Conference on Computer and Information Technology (ICCIT), Dec. 2020, pp. 1-6. https://doi.org/10.1109/ICCIT51783.2020.9392736.

[10] A. Emad, F. Bassel, M. Refaat, M. Abdelhamed, N. Shorim, and A. AbdelRaouf, "Automatic Video Summarization with Timestamps using Natural Language Processing Text Fusion," in Proc. IEEE, 2021, pp. 1-5. https://doi.org/10.1109/NLP51987.2021.9434231.

[11] T. Hussain, K. Muhammad, A. Ullah, Z. Cao, S. W. Baik, and V. H. C. de Albuquerque, "Cloud-Assisted Multi-View Video Summarization Using CNN and Bi-Directional LSTM," IEEE Transactions on Industrial Informatics, DOI: 10.1109/TII.2019.2929228. https://doi.org/10.1109/TII.2019.2929228.

[12] K. Muhammad, T. Hussain, J. Del Ser, V. Palade, and V. H. C. de Albuquerque, "DeepReS: A Deep Learning-based Video Summarization Strategy for Resource-Constrained Industrial Surveillance Scenarios," IEEE Transactions on Industrial Informatics, DOI: 10.1109/TII.2019.2960536. https://doi.org/10.1109/TII.2019.2960536.

[13] J. Wu, S.-H. Zhong, and Y. Liu, "Dynamic Graph Convolutional Network for Multi-Video Summarization," Pattern Recognition, vol. 107, Nov. 2020, p. 107382. https://doi.org/10.1016/j.patcog.2020.107382.

[14] V. J. Traver and D. Damen, "Egocentric Video Summarization via Purpose-Oriented Frame Scoring and Selection," Expert Systems with Applications, vol. 173, Feb. 2021, p. 116079. https://doi.org/10.1016/j.eswa.2021.116079.

[15] M. Basavarajaiah and P. Sharma, "GVSUM: Generic Video Summarization Using Deep Visual Features," Multimedia Tools and Applications, vol. 80, pp. 14459–14476, 2021. https://doi.org/10.1007/s11042-020-10460-0.

[16] T. Hussain, K. Muhammad, J. Del Ser, S. W. Baik, and V. H. C. de Albuquerque, "Intelligent Embedded Vision for Summarization of Multi-View Videos in IIoT," IEEE Transactions on Industrial Informatics, DOI: 10.1109/TII.2019.2937905. https://doi.org/10.1109/TII.2019.2937905.

[17] M. Li, X. Chen, S. Gao, Z. Chan, D. Zhao, and R. Yan, "VMSMO: Learning to Generate Multimodal Summary for Video-Based News Articles," in Proc. 2020 Conference on Empirical Methods in Natural Language Processing, pp. 9360–9369, 2020. https://doi.org/10.48550/arXiv.2010.05406.

[18] W. Xu, R. Wang, X. Guo, S. Li, Q. Ma, Y. Zhao, S. Guo, Z. Zhu, and J. Yan, "MHSCNET: A Multimodal Hierarchical Shot-Aware Convolutional Network for Video Summarization," in Proc. 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1-6, June 2023. https://doi.org/10.1109/ICASSP49357.2023.10096265.

[19] J.-H. Huang and M. Worring, "Query-controllable Video Summarization," in Proc. 2020 International Conference on Multimedia Retrieval (ICMR), pp. 242–250, 2020. https://doi.org/10.1145/3372278.3390695.

[20] B. Zhao, H. Li, X. Lu, and X. Li, "Reconstructive Sequence-Graph Network for Video Summarization," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 5, pp. 2793–2801, May 2022. https://doi.org/10.1109/TPAMI.2021.3072117.

[21] Y. Himeur, S. Al-Maadeed, H. Kheddar, N. Al-Maadeed, K. Abualsaud, A. Mohamed, and T. Khattab, "Video surveillance using deep transfer learning and deep domain adaptation: Towards better generalization," Engineering Applications of Artificial Intelligence, vol. 119, March 2023, 105698. https://doi.org/10.1016/j.engappai.2022.105698.

[22] G. El-Nagar, A. El-Sawy, and M. Rashad, "A deep audio-visual model for efficient dynamic video summarization," Journal of Visual Communication and Image Representation, vol. 100, April 2024, 104130. https://doi.org/10.1016/j.jvcir.2024.104130.

[23] K. Muhammad, T. Hussain, and S. W. Baik, "Efficient CNN-based summarization of surveillance videos for resource-constrained devices," Pattern Recognition Letters, vol. 130, pp. 370–375, Feb. 2020. DOI: https://doi.org/10.1016/j.patrec.2019.12.012.

[24] W. Zhu, J. Lu, J. Li, and J. Zhou, "DSNet: A Flexible Detect-to-Summarize Network for Video Summarization," IEEE Trans. on Image Processing, vol. 30, pp. 948–963, 2021. DOI: https://doi.org/10.1109/TIP.2020.3039886.

[25] P. Saini, K. Kumar, S. Kashid, A. Saini, and A. Negi, "Video Summarization Using Deep Learning Techniques: A Detailed Analysis and Investigation," Artificial Intelligence Review, vol. 56, pp. 12347–12385, 2023. DOI: https://doi.org/10.1007/s10462-023-10444-0.

[26] H. B. Ul Haq, M. Asif, M. B. Ahmad, R. Ashraf, and T. Mahmood, "An Effective Video Summarization Framework Based on the Object of Interest Using Deep Learning," Mathematical Problems in Engineering, vol. 2022, 25 pages, 2022. DOI: https://doi.org/10.1155/2022/7453744.

[27] Y. Zhang, J. Shen, S. Li, and X. Huang, "Multi-stream dynamic video summarization," IEEE Trans. on Multimedia, vol. 22, no. 7, pp. 1762–1775, Jul. 2020. DOI: https://doi.org/10.1109/TMM.2019.2947203.

[28] A. Emad, F. Bassel, M. Refaat, M. Abdelhamed, N. Shorim, and A. AbdelRaouf, "Automatic Video Summarization with Timestamps using Natural Language Processing Text Fusion," Proc. IEEE, 2021, pp. 1–5, 2021. DOI: https://doi.org/10.1109/NLP51987.2021.9434231.

[29] S. H. Emon, A. H. M. Annur, A. H. Xian, K. M. Sultana, and S. M. Shahriar, "Automatic Video Summarization from Cricket Videos Using Deep Learning," Proc. ICCIT, 2020, pp. 1–6, 2020. DOI: https://doi.org/10.1109/ICCIT51783.2020.9392736.

[30] M. Sridevi and M. Kharde, "Video Summarization Using Highlight Detection and Pairwise Deep Ranking Model," Procedia Computer Science, vol. 167, pp. 1839–1848, 2020. DOI: https://doi.org/10.1016/j.procs.2020.03.203.

[31] B. Zhao, M. Gong, and X. Li, "Audio Visual Video Summarization," IEEE Transactions on Neural Networks and Learning Systems, Vol. 34, No. 8, August 2023, pp. 5181–5188. https://doi.org/10.1109/TNNLS.2021.3119969.

[32] J. Lin, S.-H. Zhong, and A. Fares, "Deep hierarchical LSTM networks with attention for video summarization," Computers and Electrical Engineering, vol. 97, January 2022, 107618. https://doi.org/10.1016/j.compeleceng.2021.107618.

[33] W. Zhu, J. Lu, Y. Han, and J. Zhou, "Learning multiscale hierarchical attention for video summarization," Pattern Recognition, vol. 122, February 2022, 108312. https://doi.org/10.1016/j.patcog.2021.108312.

[34] X. Wu, M. Ma, S. Wan, X. Han, and S. Mei, "Multi-scale deep feature fusion based sparse dictionary selection for video summarization," Signal Processing: Image Communication, vol. 118, October 2023, 117006. https://doi.org/10.1016/j.image.2023.117006.

[35] G. Wu, S. Song, X. Wang, and J. Zhang, "Reconstructive network under contrastive graph rewards for video summarization," Expert Systems with Applications, vol. 250, 15 September 2024, 123860. https://doi.org/10.1016/j.eswa.2024.123860.

[36] B. Zhao, X. Li, and X. Lu, "TTH-RNN: Tensor-Train Hierarchical Recurrent Neural Network for Video Summarization," IEEE Transactions on Industrial Electronics, vol. 68, no. 4, pp. 3629–3637, April 2021. https://doi.org/10.1109/TIE.2020.2979573.

[37] J. Xie, X. Chen, S. Zhao, and S.-P. Lu, "Video summarization via knowledge-aware multimodal deep networks," Knowledge-Based Systems, vol. 293, 7 June 2024, 111670. https://doi.org/10.1016/j.knosys.2024.111670.

[38] B. Zhao, M. Gong, and X. Li, "Hierarchical Multimodal Transformer to Summarize Videos," Neurocomputing, vol. 468, 11 January 2022, pp. 360–369. https://doi.org/10.1016/j.neucom.2021.10.039.

[39] H. Terbouche, M. Morel, and M. Rodriguez, "Multi-Annotation Attention Model for Video Summarization," in Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 17-24 June 2023. https://doi.org/10.1109/CVPRW59228.2023.00316.

[40] O. Issa and T. Shanableh, "Static Video Summarization Using Video Coding Features with Frame-Level Temporal Subsampling and Deep Learning," Applied Sciences, vol. 13, no. 10, pp. 6065, May 2023. https://doi.org/10.3390/app13106065.

[41] Y. Zhu, W. Zhao, R. Hua, and X. Wu, "Topic-aware video summarization using multimodal transformer," Pattern Recognition, vol. 140, August 2023, 109578. https://doi.org/10.1016/j.patcog.2023.109578.

[42] H. Li, Q. Ke, M. Gong, and T. Drummond, "Progressive Video Summarization via Multimodal Self-supervised Learning," in Proceedings of the 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 02-07 January 2023. https://doi.org/10.1109/WACV56688.2023.00554.

[43] J.-H. Huang, L. Murn, M. Mrak, and M. Worring, "GPT2MVS: Generative Pre-trained Transformer-2 for Multi-modal Video Summarization," ICMR '21: Proceedings of the 2021 International Conference on Multimedia Retrieval, September 2021, pp. 580–589. https://doi.org/10.1145/3460426.3463662.

[44] P. Li, Q. Ye, L. Zhang, L. Yuan, X. Xu, and L. Shao, "Exploring Global Diverse Attention via Pairwise Temporal Relation for Video Summarization," Pattern Recognition, Vol. 111, March 2021, Article No. 107677. https://doi.org/10.1016/j.patcog.2020.107677.

[45] H. Jiang, and Y. Mu, "Joint Video Summarization and Moment Localization by Cross-Task Sample Transfer," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 16388–16398. https://link.springer.com/article/10.1007/s10489-020-01823-z.

[46] Y. Yuan, and J. Zhang, "Unsupervised Video Summarization via Deep Reinforcement Learning with Shot-Level Semantics," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 33, No. 1, January 2023, pp. 445–456. https://doi.org/10.1109/TCSVT.2022.3197819.

[47] T. Liu, Q. Meng, J.-J. Huang, A. Vlontzos, D. Rueckert, and B. Kainz, "Video Summarization Through Reinforcement Learning With a 3D Spatio-Temporal U-Net," IEEE Transactions on Image Processing, Vol. 31, January 2022, pp. 1573–1586. https://doi.org/10.1109/TIP.2022.3144691.

[48] S. S. Sulaiman, Ibraheem Nadher, and S. M. Hameed, "New Weighted Synthetic Oversampling Method for Improving Credit Card Fraud Detection," Iraqi Journal of Science, pp. 2523–2544, Jun. 2025, doi: https://doi.org/10.24996/ijs.2025.66.6.27.

[49] S. Saad, Ibraheem Nadher, and S. M. Hameed, "Credit Card Fraud Detection Challenges and Solutions: A Review," Iraqi journal of science, pp. 2287–2303, Apr. 2024, doi: https://doi.org/10.24996/ijs.2024.65.4.42.

[50] S. S. Sulaiman, Ibraheem Nadher, and S. M. Hameed , "Credit Card Fraud Detection Using an Autoencoder Model with New Loss Function," International Journal of Intelligent Engineering and Systems, vol. 17, no. 5, pp. 210–220, Oct. 2024, doi: https://doi.org/10.22266/ijies2024.1031.18.

[51] M. H. Hadid et al., "Semantic Image Retrieval Analysis Based on Deep Learning and Singular Value Decomposition," Applied Data Science and Analysis, vol. 2024, pp. 17–31, Mar. 2024, doi: https://doi.org/10.58496/adsa/2024/003.