Data Engineering and Applications

Research Paper

Learning Compact Spatio-Temporal Representations for Edge Video Keyframe Selection Using Masked Graph Autoencoders

Klementa Hronová[1,*], Jaroslav Skála[1] and Branko Malina[1]

[1] Faculty of Information Technology, Czech Technical University in Prague, Prague, 160 00, Czech Republic
*Corresponding author: klementa.h@fit.cvut.cz

Abstract. Selection of keyframes for edge video analytics needs to retain rare but operationally significant events within strict computation and bandwidth limitations. This paper proposes a Masked Graph Autoencoder for edge video keyframe selection with few-shot fusion. Video clips are converted into sparse spatio-temporal graphs where nodes contain frame-level appearance, motion and object-interaction descriptors, and adaptive edges encode temporal continuity and semantic neighborhood relations. Add a masked reconstruction objective and prototype-guided few-shot fusion to have the selector learn from a small number of labelled samples without overfitting to the dominant background frames. Experiments in an edge-oriented benchmark for traffic, campus, retail and indoor monitoring scenarios show that the proposed model has an F1 score of 88.7%, reduces redundant selected frames by 31.6%, and lowers the average per-clip decision latency to 18.4ms on an embedded GPU. Lightweight transformers, temporal clustering and graph autoencoder baselines have been used for comparison; under 5-shot supervision and at 20% packet loss, the method improved mean average precision by 5.9-11.8 percentage points and remained stable. Based on the above results, masked graph reconstruction and few-shot fusion provide a feasible path for high-fidelity, compact and deployable keyframe selection in distributed edge video systems.

Keywords: *Masked Graph Autoencoder, Few-Shot Fusion, Edge Video Analytics, Keyframe Selection, Spatio-Temporal Graph*

Received on 29 March 2023, Accepted on 22 July 2023, Published on 03 August 2023

Introduction

Video streams from roadside cameras, mobile robots, factory sensors and public-space monitoring systems are increasingly processed near the data source because continuous cloud uploads are costly, slow and often unnecessary. Edge video analytics therefore requires compact representations that retain the most useful frames for detection, retrieval and human review, and discard repetitive background content. Deng et al. [1] emphasized that edge intelligence requires compact and context-aware processing near the data source, which explains why classic keyframe selection methods based only on color histograms and shot boundaries are not applicable when two visually similar frames have different operational meanings. Deep visual descriptors have increased the semantic sensitivity, but their direct application to every frame may exceed the energy and latency budget of embedded devices [2]. Graph-based video representations are relatively good because they can show the time sequence, object relationships, and neighborhood information in one graph [3]. At the same time, masked representation learning has shown that a model can learn a stable latent structure by reconstructing the missing parts of an input and is not solely based on task labels [4]. Few-shot learning is also suitable for edge deployment, as each camera may only observe local categories and rare incidents that are not well represented in centrally labelled datasets [5]. The problem is how to integrate the above strands without creating a model that is too large for real-time edge deployment [6].

The set of requirements for a keyframe selector in edge video differs from that of a general offline summarization algorithm. It needs to select frames before the downstream detector or storage policy has full knowledge of the stream; it should be stable under changes in light and crowd density; and it must avoid

selecting numerous near-duplicate frames that waste wireless bandwidth. Recently, video summarization networks have often regarded this as a sequential importance scoring problem and employed recurrent, attention-based or Transformer-style models to determine frame values [7]. The above ways can address the issue of long-range temporal dependence, but their label requirements and inference costs are not suitable for edge cameras with sparse annotations. Other studies have reduced redundancy using clustering or diversity constraints, but these methods may fail to detect a short abnormal event if its feature distance from the dominant scene is small [8]. Lightweight graph neural networks have been proposed for video reasoning to reduce dense frame interactions into structured message passing [9]. Many graph models still require supervised labels for stable node scoring and are not easily transferable to different camera viewpoints [10]. Few-shot adaptation partially solves this problem, but naive prototype matching is susceptible to noise in the support examples and does not explicitly reconstruct missing temporal evidence [11].

The current work addresses the above deficiencies by designing a Masked Graph Autoencoder model that integrates structural self-supervision, few-shot fusion and edge-aware keyframe scoring. Instead of feeding a video segment to a dense temporal backbone, the proposed method samples candidate frames, builds a sparse graph over these frames and object regions, and masks selected node and edge attributes during training. The autoencoder is trained to reconstruct appearance, motion and semantic-interaction descriptors from neighboring evidence, and thus, the latent space learns to encode information that can withstand partial observation. A small support set from the target edge scene then calibrates the latent representation through prototype fusion. The fusion stage does not change the learned graph representation; instead, it moves the decision boundary closer to locally significant phenomena, such as a stopped car at an intersection, a short traffic jam in a hallway, or consecutive hand-object operations in a store. The engineering goals are to have a high recognition accuracy with low redundancy, a short decision delay, and good behavior in a poor network environment or sensor input degradation.

The remaining parts of this paper are as follows: Chapter 2 reviews related work on edge keyframe selection, masked graph representation learning and few-shot fusion for video understanding. Chapter 3 introduces the proposed model and explains the graph construction strategy, masked autoencoding objective, prototype fusion mechanism and edge-side selection rule. Chapter 4 presents the experimental conditions, quantitative comparisons, ablation studies, efficiency analysis and robustness tests, and all data figures are in this chapter. Chapter 5 is the main results of this study, an analysis of current problems and prospects for future research.

Related Work

Edge Video Keyframe Selection

Edge keyframe selection has evolved from shot-boundary extraction and handcrafted descriptors into a family of compact decision modules for bandwidth-aware video analysis. The earliest engineering systems typically used frame differences in color, texture, or motion energy to select local maxima within a fixed time window. Recently, learned semantic descriptors have been used to avoid selecting frames that are visually different but semantically unrelated [12]. In the fields of transportation and public safety, the chosen frames need to support downstream perception tasks and not just produce an aesthetically pleasing visual summary; thus, this requirement has led to task-aware scoring functions [13]. Some studies have introduced redundancy penalties or determinantal diversity terms to avoid transmitting the same neighboring frames multiple times from the edge device [14]. The above strategies improve compactness, but they are often sensitive to scene-specific thresholds and need to be manually tuned when cameras are relocated, lens angles are adjusted, or event frequencies vary [15].

Another type of work aims for resource-constrained inference. Lightweight convolutional encoders, knowledge-distilled temporal networks and sparse attention modules reduce computation by avoiding dense processing of all frames. Such models are suitable for edge nodes with relatively small embedded GPUs or neural processing units, and the selector still needs to address the trade-off between early filtering and semantic reliability. If filtering occurs before object interaction is understood, the informative frames will be lost. If filtering waits until a heavy detector completes, the latency advantage of keyframe selection is weakened. Therefore, graph-structured representations have been developed to preserve explicit connections and support sparse computation.

Masked Graph Learning and Few-Shot Fusion

Masked graph learning is a self-supervised method to learn structural representations by hiding some parts of a graph and asking the model to reconstruct them. The cause is that the temporal evidence in video analysis is inherently redundant and not uniformly informative. A frame can often be inferred from the surrounding frames, but an abrupt change in object relations or motion discontinuities are difficult to recover and thus more valuable for selection. Graph autoencoders have been employed to learn low-dimensional embeddings of relational data and can be adapted for spatio-temporal video graphs when nodes represent frames, regions or tracked objects [16]. Masking makes the encoder use neighborhood context instead of memorizing local descriptors and enhances robustness to missing or corrupted observations [17].

Few-shot fusion addresses a related problem: the target edge scene is generally different from the data used in central training. Prototype-based learning represents each class or event condition with a small set of support examples to achieve quick adaptation and is label-scarce [18]. Fusion methods can combine these prototypes with visual, motion and context cues, but many designs treat all support examples as equally reliable. Support clips in edge videos may be noisy due to short rare events and sparse manual annotations. Therefore, adaptive fusion needs a confidence mechanism that down-weights unstable prototypes and keeps rare but discriminative evidence [19]. The method presented in this paper connects the few-shot calibration to masked graph autoencoding, and thus the local prototypes guide selection without losing the structural self-supervision that stabilizes the representation.

Proposed Method

Spatio-Temporal Graph Construction

The proposed system receives a video stream at an edge node and processes it in short clips of T frames. First of all, a lightweight feature extractor is used to generate frame appearance descriptors, optical-flow statistics and object-region embeddings for each clip. Instead of building a full-connected temporal graph, a sparse graph with frame nodes and high-confidence region nodes as its nodes is constructed. Temporal edges are adjacent frame nodes in a small window; semantic edges connect regions with the same object category or high cosine similarity; and interaction edges connect objects that have a relationship in terms of bounding boxes and motion vectors. This design is compact enough for edge inference and still shows the cues that make a frame significant.

The graph construction stage applies two pruning rules before message passing; motion entropy measures temporal uncertainty, while object-count change reflects abrupt semantic variation. First, a frame node is kept if its motion entropy, object-count change or descriptor distance from the previous retained frame exceeds a soft threshold learned from unlabeled clips. Second, region nodes with unstable detections are not immediately discarded; they are attached to the nearest frame node with a confidence attribute, and thus the uncertainty can affect reconstruction difficulty later on. Therefore, in the case of edge videos, a detector failure may be due to motion blur, glare or crowd occlusion; such instances should be avoided by a keyframe selector. The resulting graph generally has 24 to 42 frame nodes and 60 to 110 region nodes for a 96-frame buffer, which is feasible for sparse attention.

Figure 1 shows the entire pipeline of video buffering, graph construction, masked encoding, prototype fusion and final keyframe output. The pipeline is intentionally modular; thus, the feature extractor can be replaced with another edge backbone, and the graph encoder and selector work with normalized descriptors. This division is suitable for engineering deployment because different camera groups may have different hardware accelerators, but they can still use the same graph-selection module.

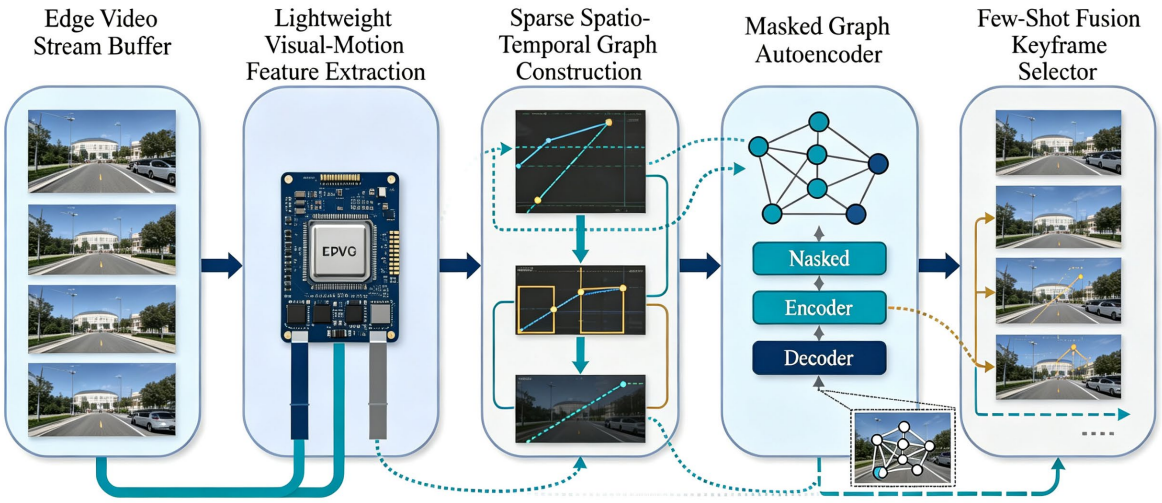

Figure 1. Overall edge video keyframe selection pipeline based on sparse spatio-temporal graph construction and masked graph autoencoding.

Masked Graph Autoencoder

Let $G = (V, E, X)$ denote the graph of one clip, where V is the node set, E is the edge set, and X contains the node descriptors. An encoder maps the visible graph to latent node representations, and a decoder reconstructs the masked node attributes and selected edge attributes. The Mask is not randomly generated. Some of the mask is sampled from high-motion or low-confidence nodes, and another part is sampled uniformly to prevent the model from learning only trivial temporal interpolation.

The decoder is deliberately shallower than the encoder. It is not to create realistic pictures but to determine how difficult it is to recover a node or edge based on the surrounding evidence. An ordinary background frame with stable illumination and repeated objects can be reconstructed at a low error, while a frame with an abrupt relation change produces a larger residual. The above are all parts of the keyframe score. During training, the model is shown many masks of appearance, motion and interaction, so no single descriptor family is overly dominant in the latent space. Only the compact scoring head of the reconstruction branch is kept during inference to avoid the cost of full video generation.

The graph encoder updates node states through sparse message passing:

$$G = (V, E, X) \quad \text{Eq.(1)}$$

Here, appearance, motion, and object descriptors are concatenated into a unified node attribute vector before graph masking is applied.

$$h_i' = \sigma\left(W_0 h_i + \sum_{j \subset N_i} \alpha_{ij} W_1 h_j\right) \quad \text{Eq.(2)}$$

The attention coefficient is conditioned by an edge bias term that differentiates temporal, semantic, and interaction edges.

$$\alpha_{ij} = softmax_j\left(q_i^T k_j / \sqrt{d} + b_{ij}\right) \quad \text{Eq.(3)}$$

The reconstruction loss combines masked node recovery and masked edge recovery, which makes rare relational changes more visible to the selector. Thus, frames that are difficult to reconstruct from neighboring evidence are treated as stronger keyframe candidates rather than as ordinary prediction errors.

$$\mathcal{L}_{rec} = \left\|X_M - \hat{X}_M\right\|_2^2 + \lambda \left\|E_M - \hat{E}_M\right\|_2^2 \quad \text{Eq.(4)}$$

Few-Shot Fusion and Selection Rule

The few-shot module has a support set of the target scene. Each support example has a frame-level label that indicates whether the frame is informative for the local task, such as incident, interaction, transition or

background. For each label group, the system computes a prototype in the latent graph space and then combines prototype similarity with reconstruction difficulty, diversity and edge-side cost. The fusion module is relatively lightweight; it only needs to be invoked during re-initialization of the camera or when a small number of new annotations are added.

Prototype fusion is carried out after graph encoding, not at the level of raw features. Therefore, this selection is less affected by changes in local color and texture because the prototype already contains information about time and place. To reduce support-set noise further, a prototype confidence value is derived from the within-class dispersion of support embeddings. A support group with a large dispersion contributes less to the final score until more labelled examples are added. The selector will also add a minimum latent-distance constraint to the selected frames to avoid having too many visually similar frames at the same keyframe.

Figure 2 is the training and inference workflow. Masked reconstruction and prototype calibration are optimized jointly in training. Inference mode uses the decoder solely to compute the reconstruction-difficulty and uncertainty maps, and then the selector fuses these important scores with redundancy reduction. This avoids running the heavy downstream recognition network on the selected keyframes.

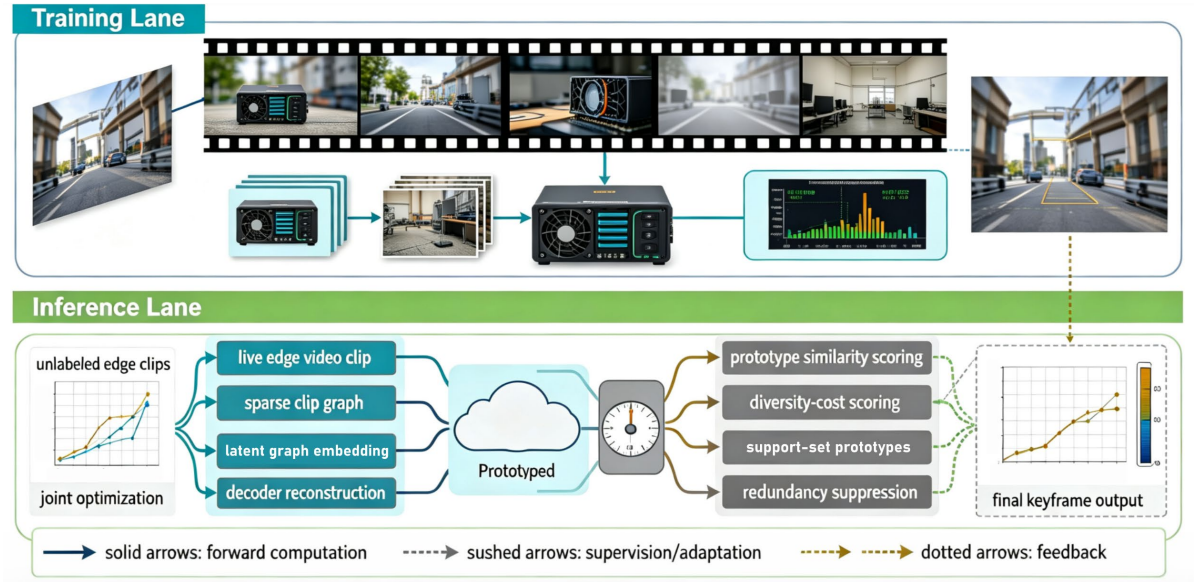

Figure 2. Training and inference workflow for masked reconstruction, few-shot prototype fusion, and edge-side keyframe selection.

$$p_c = K_c^{-1} \sum_{i \in S_c} z_i \qquad \text{Eq.(5)}$$

The prototype score is computed in the latent graph space and uses only the support examples available at the target edge scene.

$$S_i = \beta_r r_i + \beta_p cos(z_i, p_c) + \beta_d d_i - \beta_c c_i \qquad \text{Eq.(6)}$$

The final keyframe set is selected by ranking fused scores while suppressing frames whose latent distance to already selected frames is below a redundancy threshold.

Experimental Data and Analysis

Experimental Setting and Comparative Results

The edge-oriented benchmark contains 3,840 clips and four scene groups: urban traffic, campus walkways, retail aisles and indoor corridors. Each group has clips of daily life and short informative cases, such as car queues, pedestrian congestion, object exchanges, shelf interactions, and sudden scene changes. Training protocols are 1-shot, 3-shot, 5-shot and 10-shot adaptations for each target scene, and unlabeled frames are available for masked graph pretraining. The profile of the edge device is an embedded GPU with an 18W power limit and a streaming buffer of 96 frames. Baselines are: uniform sampling, k-medoids temporal clustering, a lightweight

temporal convolutional selector, a compact transformer selector, a standard graph autoencoder, and a few-shot prototype network. Evaluation indicators are F1-score, mean average precision, redundancy ratio, downstream detection recall, latency, memory usage and bandwidth saved.

Figure 3(a) shows the comparison of F1-scores under few-shot settings: the proposed model is 78.6% for 1-shot supervision and reaches 90.8% after 10 shots; the compact transformer achieves 83.9% at 10 shots, and the standard graph autoencoder is 85.1%. As shown in Figure 3(b), mean average precision also exhibits the same trend; the proposed model reaches 89.4% in the 10-shot case and maintains 81.2% in the 3-shot case. Figure 3(c) shows the redundancy ratio; a lower value indicates fewer near-duplicate selected frames, and the proposed model has reduced redundancy to 18.7% at 5 shots compared with 27.4% for the graph autoencoder and 35.9% for temporal clustering. Figure 3(d) shows the latency profile, and the proposed model is between 17.9 ms and 19.1 ms across support settings because adaptation alters prototype statistics rather than graph depth. A stronger low-shot behavior is consistent with the recent findings that relational self-supervision can stabilize downstream selection under annotation scarcity [20]. This also explains why the proposed selector has a relatively large margin in both accuracy and compactness before the support set becomes too large.

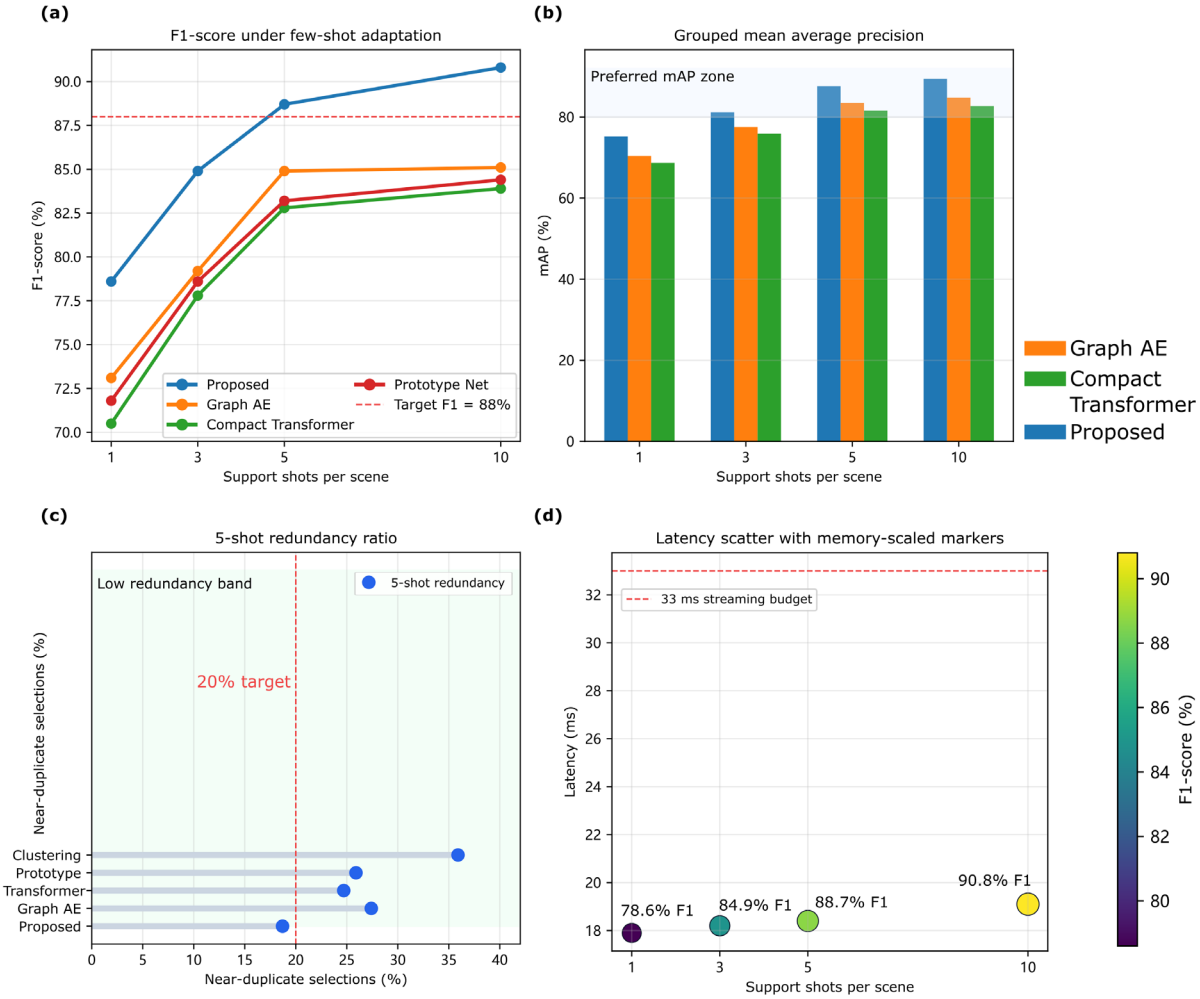

Figure 3. Few-shot performance analysis: (a) F1-score curves across support shots; (b) grouped mean average precision across support shots; (c) 5-shot redundancy comparison; (d) latency scatter profile across support shots.

The few-shot performance trend also indicates that the gain is not achieved through a single favorable support configuration. Across five random support splits, the 5-shot F1-score of the proposed model is between 87.9% and 89.4%, and the prototype network is between 80.6% and 84.9%. A small standard deviation suggests that masked graph pre-training has provided a stable latent space for the few-shot branch training. Temporal clustering has the smallest variation, but its stability is due to ignoring local semantic labels, not accurate adaptation. The two are not the same; a deterministic but insensitive selector may appear reliable in the logs

yet still fail to detect short-lived incidents. The compact transformer improves with the addition of more labels, but it also has more attention operations and shows larger scene-to-scene variation in 1-shot and 3-shot settings.

Scene-level performance is shown in Figure 4(a), where the proposed model obtains F1-scores of 89.1%, 87.4%, 88.9%, and 86.8% on traffic, campus, retail, and indoor scenes, respectively. The smallest margin is in the indoor corridor because the background structure is monotonous and the interaction is short. Figure 4(b) shows the downstream detection recall after keyframe filtering; the proposed selector retains 92.6% of detector-relevant events but only transmits 22.5% of the frames, and the same analysis records missed rare interactions at 6.8%, false informative selections at 8.3%, and duplicated selections at 7.9%. These values indicate that the method improves task preservation without simply expanding the selected frame set. The Design also conforms to the engineering requirement of task-preserving compaction rather than merely aesthetically pleasing summarization [21].

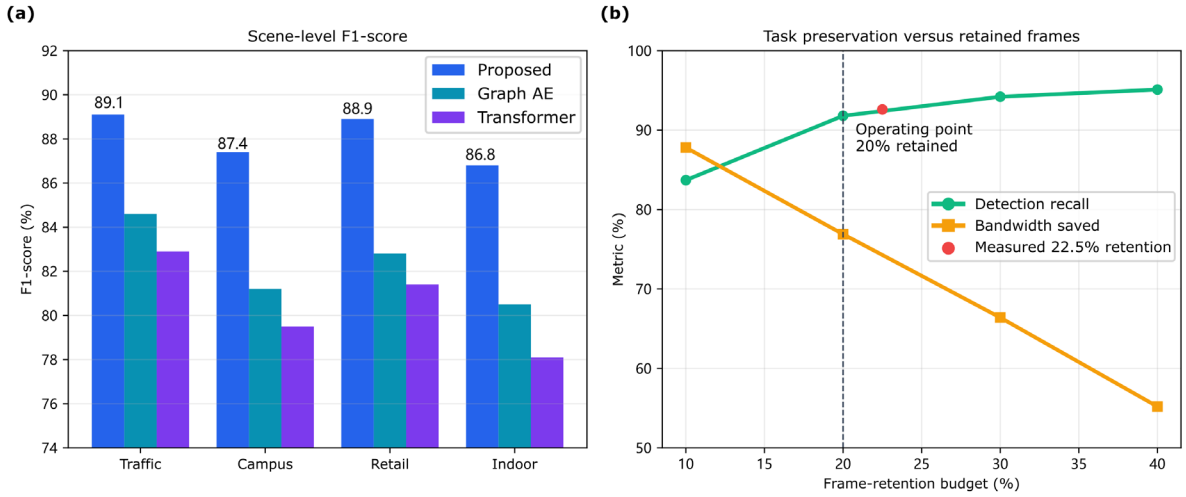

Figure 4. Scene-level task preservation: (a) F1-score by scene group; (b) downstream detection recall and bandwidth saved versus retained frames.

A closer scene-level inspection explains why graph relations are useful in this setting. Traffic clips have strong object-motion relationships, and thus interaction edges frequently identify frames where vehicle spacing changes rapidly even before congestion is visually apparent. Campus scenes have more appearance variations because pedestrians enter and leave the camera's view at different scales, and prototype similarity can be used to distinguish between ordinary walking and crowd formation. Retail clips have many small hand-object relationships, and improving the precision of semantic region edges requires shelves and products in the background. Indoor corridor clips are still relatively difficult because the same person may appear repeatedly in a long sequence with little change in pose, and the diversity constraint will exclude some frames that are later needed for fine-grained review.

Ablation Study and Sensitivity Analysis

Ablation experiments isolate the contribution of graph masking, semantic edges, prototype fusion, and redundancy-aware scoring. Figure 5(a) reports the ablation bars: removing the masked reconstruction objective reduces F1-score from 88.7% to 82.6%, removing few-shot fusion lowers it to 84.1%, and removing semantic edges lowers it to 85.0%. Therefore, the full model benefits most from masked learning, and the few-shot branch is still required for scene adaptation. As shown in Figure 5(b), the proposed model can separate informative and background frames more distinctly than the graph autoencoder baseline for reconstruction-error density; the median of informative frames is 0.61 and that of background frames is 0.34. Figure 5(c) is the effect of the mask ratio. The performance at a 20% mask ratio is 84.9%; it reaches 88.9% at 45% masking, and then slightly declines at a ratio of 65%; therefore, excessive masking removes too much local information required for stable reconstruction. Figure 5(d) shows threshold sensitivity, and with a latent-distance threshold of 0.18, duplicated selections are kept below 19% while maintaining event recall above 92%. The behavior is consistent with masked-representation studies that show moderate corruption enhances invariance and extreme corruption

weakens semantic grounding [22]. The observed optimum is near the middle mask range, and reconstruction difficulty is most useful when the graph is still partially visible, not severely obscured.

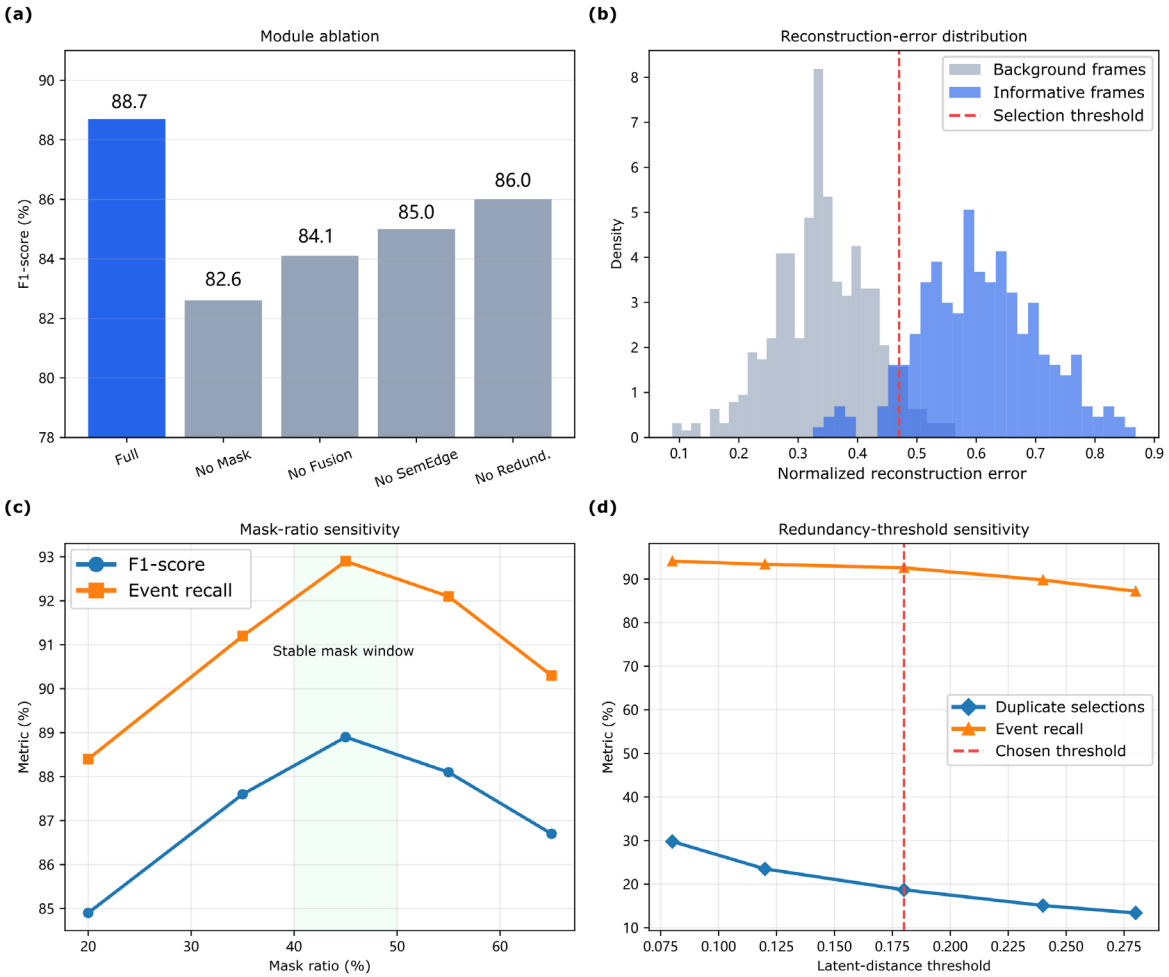

Figure 5. Ablation and masking behavior: (a) module ablation results; (b) reconstruction-error distributions; (c) mask-ratio sensitivity; (d) redundancy-threshold sensitivity.

According to the ablation study, the reasons for the drop in performance are varied. Without masked reconstruction, the model still sees the graph structure but loses a calibrated measure of novelty, so it tends to select frames that match the support prototypes and ignore short transitional evidence. Without few-shot fusion, the selector keeps general graph anomalies but is less in line with local operating definitions of importance. The reduction of semantic edges mainly affects retail and indoor clips, as the relations among objects are relatively more informative than general motion. Removing redundancy-aware scoring increased the number of duplicated selections by 9.6 percentage points, and although the F1-score decreased only slightly, it showed that accuracy alone underestimated the operational cost of the ablated model.

Figure 6(a) shows the fusion-weight sensitivity, which is affected by reconstruction difficulty, prototype similarity, diversity and cost penalty. The best set has the following weights: 0.38, 0.31, 0.21, and 0.10, and achieves an F1-score of 88.7% with a redundancy ratio of 18.4%. Figure 6(b) is the radar chart for accuracy, compactness, latency, robustness and memory efficiency; the proposed model is not the best in terms of raw memory use and has a smaller temporal clustering, but it outperforms the transformer selector in the trade-off between accuracy and compactness, with an expected calibration error of 0.041 compared with 0.076. Better calibration is important at the edge because selection thresholds are often changed automatically under bandwidth pressure [23].

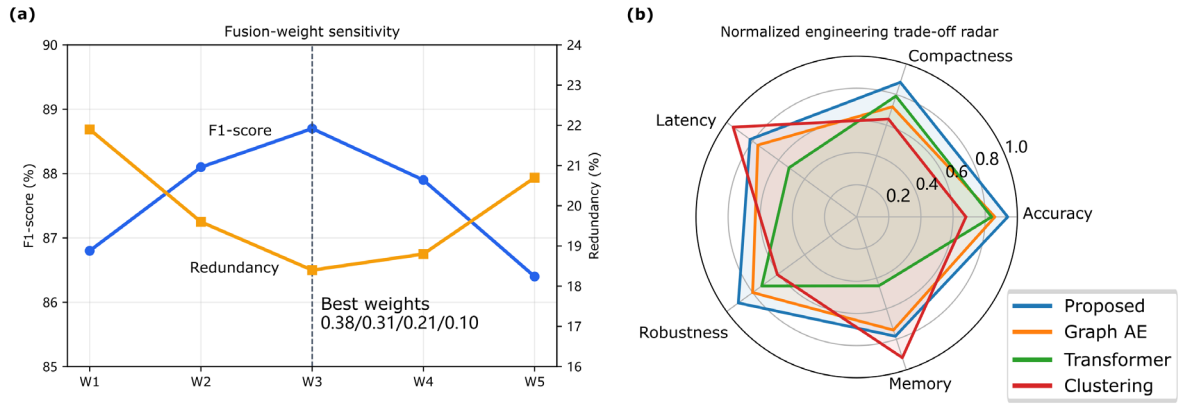

Figure 6. Fusion and operating frontier: (a) fusion-weight sensitivity; (b) normalized engineering trade-off radar.

Threshold sensitivity is evaluated by varying the redundancy distance in latent cosine space from 0.08 to 0.28. Values less than 0.12 are too lenient and allow many adjacent frames to pass; although detection recall increases slightly, bandwidth savings are reduced. Values greater than 0.24 are too restrictive and omit frames near the beginning and end of a short event. A reasonable value of 0.18 has been selected to balance the trade-off between duplicate selections (less than 19%) and event recall in the 5-shot setting (over 92%). This is especially true for edge cameras that change their upload policy under a high-network-load condition; otherwise, the same ranking scores would need to be reused with a more restrictive top-k budget after model retraining.

Edge Efficiency and Robustness

Efficiency results are plotted in Figure 7(a), where the proposed model requires 18.4 ms per clip, 312 MB of peak memory, and 5.8 mJ per selected keyframe. The temporal clustering baseline is faster at 9.6 ms but loses 12.4 percentage points in F1-score, while the compact transformer needs 35.7 ms and 615 MB. Figure 7(b) shows robustness and bandwidth behavior together: at a 20% frame-retention budget, the proposed method saves 76.9% of upload bandwidth while keeping downstream detection recall at 91.8%, and under 20% packet loss, its F1-score is 84.5%, higher than the 76.9% achieved by the transformer selector. This operating area is needed for many camera arrangements that have changing uplink requirements throughout the day [24].

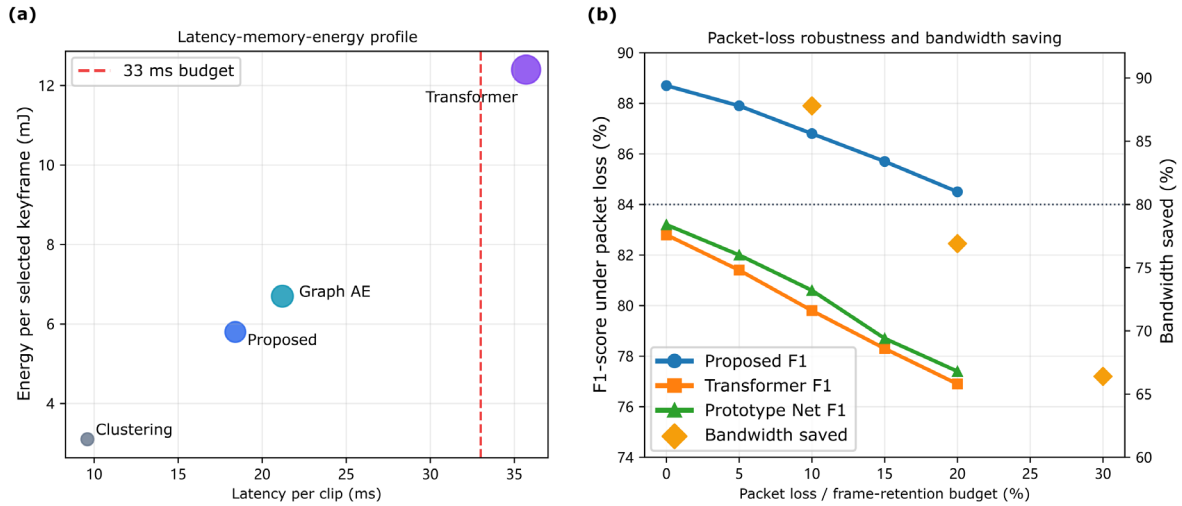

Figure 7. Edge efficiency and robustness: (a) latency-memory-energy profile; (b) packet-loss robustness with bandwidth-saving markers.

Robustness is also tested under frame blur, illumination shift, partial object detector failure, latency jitter, and support-label noise. The masked graph autoencoder is less sensitive because masked training exposes the model to partial observations before deployment [25]. Across 400 streaming windows, the proposed method keeps the 95th percentile latency at 23.6 ms, below the 33 ms budget for 30 frames per second buffering. With 15% noisy

support labels, precision remains 86.2% for the proposed model, compared with 80.4% for the prototype network. Practical edge annotation is rarely perfect, and this result supports the use of reconstruction difficulty as a counterweight to noisy prototypes [26]. The robustness protocol uses 20% packet loss, 15% support-label noise, controlled blur, illumination shift, and detector-dropout settings to approximate practical edge disturbances.

The runtime profile shows that graph sparsity contributes more to latency control than reducing the encoder width alone. When the average node degree is increased from 5.1 to 9.8, F1-score improves by only 0.6 percentage points, while latency rises from 18.4 ms to 27.9 ms. When the hidden dimension is reduced from 128 to 96 at the original sparsity level, latency decreases by 2.1 ms but F1-score falls by 1.4 percentage points. These measurements indicate that the preferred deployment setting should preserve representation width and control cost through graph construction. The result also explains why the method stays below the buffering budget while maintaining enough latent capacity for prototype fusion.

The numbers also show the two deployment-related modes. First, the model gains more from 3-shot to 5-shot adaptation than from 5-shot to 10-shot adaptation; thus, a small number of local labels are sufficient to align the latent graph with camera-specific events. Second, calibration and noise-tolerance experiments show that the support-set ambiguity can be addressed by balancing prototype similarity with reconstruction and diversity evidence. Edge learning studies have reported similar dependence on calibration when models must adapt without a full validation set [27].

From the perspective of recent edge-analytics design principles, the engineering value of the proposed method becomes clearer. A selector with higher accuracy but a larger number of frames will increase transmission and storage costs; a very small selector might not provide enough frames for the downstream detectors. The 5-shot setting in the proposed model has achieved an 88.7% F1-score, 18.7% redundancy and 18.4ms latency, thus reaching a good region of the accuracy-cost trade-off. Research on distributed video intelligence has shown that, for such frontier behaviors, multiple optimization goals are more feasible than a single one [28]. The model is compatible with the standard compression pipeline and only outputs frame indices without modifying the video bitstream [29]. For privacy-sensitive deployments, keyframe selection can reduce the upload of ordinary background frames before downstream anonymization or event review [30].

Conclusion

This paper proposes a Masked Graph Autoencoder for edge video keyframe selection with few-shot fusion. The model models each video clip as a sparse spatio-temporal graph, learns a stable latent structure via masked reconstruction, and adapts the selector to local edge scenes through prototype-guided fusion. The primary engineering problem of the design is how to save frames relevant to downstream reasoning and reduce redundant transmission and computation. Connect reconstruction difficulty, semantic relation model, diversity control and edge-side cost to build a unified framework for deployable keyframe selection under label scarcity.

Several limitations remain. At present, the graph construction employs a light-weight object detector, and severe detector failure may reduce the efficacy of semantic edges even when frame-level motion cues are still present. The support-set labels are also assumed to be available shortly after deployment, but this may not be the case in areas with delayed annotation feedback. The model supports short streaming buffers and, for longer temporal dependencies, can use a memory mechanism to store context without increasing edge latency.

Future work will extend the method to continuous adaptation, privacy-preserving collaborative learning of multiple cameras, and hardware-aware graph pruning. Another promising direction is to connect keyframe selection with downstream task feedback, and thus have the edge node learn which selected frames have actually improved detection, retrieval or human review. More diverse applications in practice will also help us learn how masked graph learning performs under seasonal scene variations, rare event occurrences and mixed-sensor quality environments.

Author Contributions

Klementa Hronová contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision, project administration, and funding acquisition. Jaroslav Skála contributes to analysis, investigation, data collection, draft preparation. Branko

Malina contributes to software, validation. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

[1] Deng, S., Zhao, H., Fang, W., Yin, J., Dustdar, S., & Zomaya, A. Y. (2020). Edge intelligence: The confluence of edge computing and artificial intelligence. IEEE Internet of Things Journal, 7(8), 7457-7469. https://doi.org/10.1109/JIOT.2020.2984887

[2] Wang, F., Zhang, M., Wang, X., Ma, X., & Liu, J. (2020). Deep learning for edge computing applications: A state-of-the-art survey. IEEE Access, 8, 58322-58336. https://doi.org/10.1109/ACCESS.2020.2982411

[3] Wang, X., Han, Y., Leung, V. C. M., Niyato, D., Yan, X., & Chen, X. (2020). Convergence of edge computing and deep learning: A comprehensive survey. IEEE Communications Surveys & Tutorials, 22(2), 869-904. https://doi.org/10.1109/COMST.2020.2970550

[4] Chang, Z., Liu, S., Xiong, X., Cai, Z., & Tu, G. (2021). A survey of recent advances in edge-computing-powered artificial intelligence of things. IEEE Internet of Things Journal, 8(18), 13849-13875. https://doi.org/10.1109/JIOT.2021.3088875

[5] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A comprehensive survey on transfer learning. Proceedings of the IEEE, 109(1), 43-76. https://doi.org/10.1109/JPROC.2020.3004555

[6] Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine, 37(3), 50-60. https://doi.org/10.1109/MSP.2020.2975749

[7] Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., & Poor, H. V. (2021). Federated learning for internet of things: A comprehensive survey. IEEE communications surveys & tutorials, 23(3), 1622-1658. https://doi.org/10.1109/COMST.2021.3075439

[8] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaria, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data, 8, 53. https://doi.org/10.1186/s40537-021-00444-8

[9] Zaidi, S. S. A., Ansari, M. S., Aslam, A., Kanwal, N., Asghar, M., & Lee, B. (2022). A survey of modern deep learning based object detection models. Digital Signal Processing, 126, 103514. https://doi.org/10.1016/j.dsp.2022.103514

[10] Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikainen, M. (2020). Deep learning for generic object detection: A survey. International Journal of Computer Vision, 128, 261-318. https://doi.org/10.1007/s11263-019-01247-4

[11] Wu, X., Sahoo, D., & Hoi, S. C. H. (2020). Recent advances in deep learning for object detection. Neurocomputing, 396, 39-64. https://doi.org/10.1016/j.neucom.2020.01.085

[12] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2021). A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems, 32(1), 4-24. https://doi.org/10.1109/TNNLS.2020.2978386

[13] Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2020). Graph neural networks: A review of methods and applications. AI Open, 1, 57-81. https://doi.org/10.1016/j.aiopen.2021.01.001

[14] Zhang, Z., Cui, P., & Zhu, W. (2022). Deep learning on graphs: A survey. IEEE Transactions on Knowledge and Data Engineering, 34(1), 249-270. https://doi.org/10.1109/TKDE.2020.2981333

[15] Zhou, Y., Zheng, H., Huang, X., Hao, S., Li, D., & Zhao, J. (2022). Graph neural networks: Taxonomy, advances, and trends. ACM Transactions on Intelligent Systems and Technology, 13(1), 1-54. https://doi.org/10.1145/3495161

[16] Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. ACM Computing Surveys, 53(3), 1-34. https://doi.org/10.1145/3386252

[17] Antonelli, S., Avola, D., Cinque, L., Crisostomi, D., Foresti, G. L., Galasso, F., Marini, M. R., Mecca, A., & Pannone, D. (2022). Few-shot object detection: A survey. ACM Computing Surveys, 54(11s), 1-37. https://doi.org/10.1145/3519022

[18] Hong, B., Zhou, Y., Qin, H., Wei, Z., Liu, H., & Yang, Y. (2022). Few-shot object detection using multimodal sensor systems of unmanned surface vehicles. Sensors, 22(4), 1511. https://doi.org/10.3390/s22041511

[19] Vu, A.-K. N., Nguyen, N.-D., Nguyen, K.-D., Nguyen, V.-T., Ngo, T. D., Do, T.-T., & Nguyen, T. V. (2022). Few-shot object detection via baby learning. Image and Vision Computing, 120, 104398. https://doi.org/10.1016/j.imavis.2022.104398

[20] Song, H., Sun, L., Yu, X., Zhang, L., & Wang, Y. (2022). A fusion spatial attention approach for few-shot learning. Information Fusion, 81, 187-202. https://doi.org/10.1016/j.inffus.2021.11.019

[21] Zheng, W., Yan, L., Gou, C., & Wang, F.-Y. (2022). Meta-learning meets the Internet of Things: Graph prototypical models for sensor-based human activity recognition. Information Fusion, 80, 1-22. https://doi.org/10.1016/j.inffus.2021.10.009

[22] He, K., Chen, X., Xie, S., Li, Y., Dollar, P., & Girshick, R. (2022). Masked autoencoders are scalable vision learners. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 15979-15988. https://doi.org/10.1109/CVPR52688.2022.01553

[23] Hou, Z., Liu, X., Cen, Y., Dong, Y., Yang, H., Wang, C., & Tang, J. (2022). GraphMAE: Self-supervised masked graph autoencoders. Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 594-604. https://doi.org/10.1145/3534678.3539321

[24] Apostolidis, E., Adamantidou, E., Metsai, A. I., Mezaris, V., & Patras, I. (2021). Video summarization using deep neural networks: A survey. Proceedings of the IEEE, 109(11), 1838-1863. https://doi.org/10.1109/JPROC.2021.3117472

[25] Narwal, P., Duhan, N., & Bhatia, K. K. (2022). A comprehensive survey and mathematical insights towards video summarization. Journal of Visual Communication and Image Representation, 89, 103670. https://doi.org/10.1016/j.jvcir.2022.103670

[26] Tiwari, V., & Bhatnagar, C. (2021). A survey of recent work on video summarization: Approaches and techniques. Multimedia Tools and Applications, 80(18), 27187-27221. https://doi.org/10.1007/s11042-021-10977-y

[27] Hussain, T., Muhammad, K., Ding, W., Lloret, J., Baik, S. W., & de Albuquerque, V. H. C. (2021). A comprehensive survey of multi-view video summarization. Pattern Recognition, 109, 1-15. https://doi.org/10.1016/j.patcog.2020.107567

[28] Ma, M., Mei, S., Wan, S., Hou, J., Wang, Z., & Feng, D. D. (2020). Video summarization via block sparse dictionary selection. Neurocomputing, 378, 197-209. https://doi.org/10.1016/j.neucom.2019.07.108

[29] Ji, Z., Xiong, K., Pang, Y., & Li, X. (2020). Video summarization with attention-based encoder-decoder networks. IEEE Transactions on Circuits and Systems for Video Technology, 30(6), 1709-1717. https://doi.org/10.1109/TCSVT.2019.2904996

[30] Ma, M., Mei, S., Wan, S., Wang, Z., Ge, Z., Lam, V., & Feng, D. D. (2021). Keyframe extraction from laparoscopic videos via diverse and weighted dictionary selection. IEEE Journal of Biomedical and Health Informatics, 25(5), 1686-1698. https://doi.org/10.1109/JBHI.2020.3019198