DOI: 10.24996/ijs.2026.67.5.37

ISSN: 0067-2904

Deep Learning Techniques for Video Summarization Based on Object Detection

Shadan Abdul Haleem*, Eman Hato
Department of Computer Science, College of Science, Mustansiriyah University, Baghdad, Iraq

Received: 8/2/2025 Accepted: 1/6/2025 Published: xx

Abstract

With the rapid growth of video content, effective video summarization methods are essential. This paper introduces a new framework using deep learning for object detection. YOLOv8 first identifies objects in each frame from every 15-frame sequence. These objects are cropped and resized for feature extraction with Residual Neural Network (ResNet 50). A clustering process using Hierarchical Density-Based Spatial Clustering (HDBSCAN) classifies each object. Finally, keyframes are randomly selected from each object cluster to create a concise summary. This paper primarily contributes to the identification of video objects, such as people and vehicles, to retain the most informative content. Additionally, it generates a video summary that significantly reduces the original length while preserving a diverse range of video content. The framework's performance was tested on the SumMe dataset, with accuracy and F1-score as key metrics. Results show an overall detection accuracy of 0.8988 and an F-score of 0.9451. The method produced very short video summaries, saving an average of 95% of the time compared to the original videos, demonstrating a significant reduction in video length while maintaining summary reliability.

Keywords: Deep Learning, Video Summarization, Keyframe Selection, Object Detection, Clustering Algorithm.

تقنيات التعلم العميق لتلخيص الفيديو اعتمادًا على اكتشاف الأجسام

شدن عبدالحليم*, ايمان هاتو

قسم علوم الحاسوب, كلية العلوم, الجامعة المستنصرية, بغداد, العراق

الخلاصة

مع النمو السريع للمحتوى الفيديوي، تصبح طرق تلخيص الفيديو الفعالة ضرورية. تقدم هذه الورقة نموذج جديدًا يعتمد على التعلم العميق للكشف عن الكائنات. يبدأ النظام باستخدام YOLOv8 لتحديد الكائنات في كل إطار من كل تسلسل مكون من 15 إطارًا. تُقص الكائنات المكتشفة وتُغير أبعادها لاستخراج الميزات باستخدام تقنية ResNet 50. يتم تصنيف كل كائن عبر عملية تجميع باستخدام طريقة التجميع الهرمي الكثافي المعتمد على الكثافة (HDBSCAN). وأخيرًا، تُختار إطارات رئيسية بشكل عشوائي من كل مجموعة كائنات لإنشاء ملخص مختصر. يُسهم هذا البحث بشكل رئيسي في تحديد عناصر الفيديو، مثل الأشخاص والمركبات، للحفاظ على المحتوى الأكثر فائدة. كما يُنتج ملخصًا للفيديو يُقلل بشكل كبير من طوله الأصلي مع الحفاظ على تنوع محتوى الفيديو. تم تقييم أداء النموذج باستخدام مجموعة بياناتSumMe ، مع اعتبار الدقة ومعدل-F1

*Email: shedanaljumaily26@uomustansiriyah.edu.iq

Scoreكمؤشرات رئيسية. تظهر النتائج دقة كشف إجمالية بنسبة 0.8988 ومعدل F1–Score بنسبة 0.9451. كما أن المنهج أنتج ملخصات فيديو قصيرة جدًا، مما وفر في المتوسط 95% من الوقت مقارنة بالفيديوهات الأصلية، مما يدل على تقليل ملحوظ في طول الفيديو مع الحفاظ على موثوقية الملخص.

1. Introduction

In recent years, the rise of social media and video-sharing platforms has led to an exponential growth in user-generated audiovisual content. Individuals frequently capture and share various aspects of their daily lives, including personal videos, such as moments spent with friends, and family, or engaging in hobbies, activity videos, like sports or similar events, review videos, where users express opinions on products, services, or movies, and how-to videos, designed to teach others how to complete specific tasks [1]. Alongside content creators, millions of users consume vast amounts of this material daily. For instance, YouTube users collectively watch over one billion hours of video content every day, while more than 500 hours of new content are uploaded every minute. The rapid expansion of video content each year has made it increasingly difficult to manage and process the vast amount of visual data, as illustrated in Figure 1. As video production continues to grow across various industries, the demand for effective video summarization has become more critical. Video Summarization (VS) helps minimize the time and effort needed to analyse lengthy videos, simplifying the process of extracting valuable information [2].

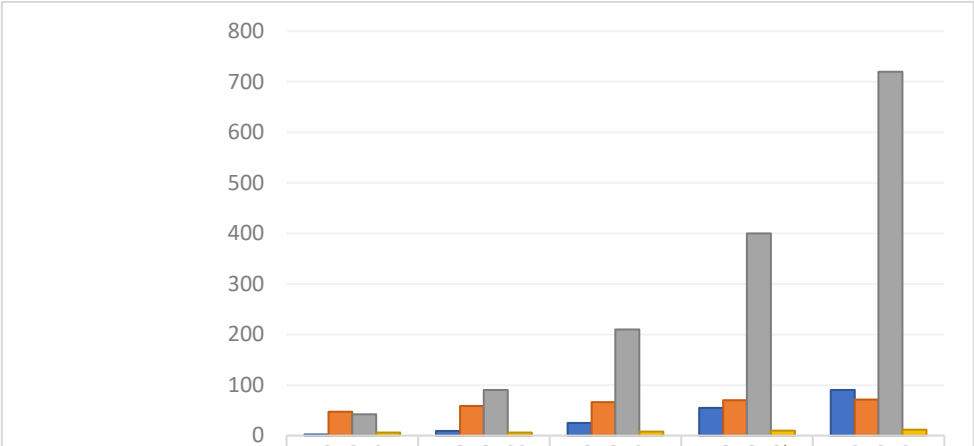
VS is a popular approach for creating efficient video archiving systems. It involves generating a summary of a video, which can consist of still or moving images. The main goal of VS is to minimize the amount of data that needs to be analysed to retrieve specific information, making it a crucial task in video analysis and indexing applications [3].

There are various methods to perform video summarization depending on what information is required to capture; they may be classified into four significant categories: Feature-based summarization, Event-based summarization, Attention-based summarization, and Object-based summarization. In Object-based summarization, the objects in the extracted frames are detected, and if frames that had the required objects are saved in a summarized video, otherwise, they are discarded [4, 5].

Deep learning, a subset of machine learning, has advanced significantly due to the rapid growth of data and improvements in hardware technologies. Its "deep" nature refers to multiple network layers for non-linear processing, enabling the learning of complex, hierarchical data representations. Early deep-learning-based methods for video summarization framed the task as a structured prediction problem, estimating the importance of video frames by modelling their temporal dependencies. Deep learning models require a massive amount of data for training, which is a significant challenge due to the scarcity of publicly available real-world datasets [6,7].

YOLO (You Only Look Once) models are real-time object detection systems that identify and classify objects in a single pass of the image. YOLO frames object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. It looks at the whole image at test time, so its predictions are informed by global context in the image. YOLO has several advantages over previous methods. It is extremely fast and can therefore be applied to use cases such as self-driving cars and video surveillance. Additionally, it has high accuracy, especially with natural images, and fewer false positives since it analyses an entire image simultaneously, providing greater contextual accuracy. The YOLO algorithm employs a single Convolutional Neural Network (CNN) that divides the image into a grid. Each cell in the grid predicts a certain number of bounding boxes. Along with each bounding box, the cell also predicts a class probability, which indicates the likelihood of a specific object being present in the box [8, 9].

YOLOv8 introduced a new architecture, which is an advanced version of the YOLOv series, known for its efficiency and performance in object detection tasks. One key technique YOLOv8 improves upon previous YOLO versions through several significant enhancements, including a modified backbone network, an anchor-free detection head, and a new loss function that boosts speed and accuracy. Additionally, YOLOv8 employs multi-scale object detection, allowing it to effectively identify objects of various sizes within an image. Using the SiLU (Sigmoid Linear Unit) activation function further accelerates learning by mitigating the vanishing gradient problem, leading to faster convergence. These improvements contribute to YOLOv8's efficiency in accurately identifying diverse objects in videos, such as persons, cars, buses, trucks, bicycles, airplanes, trains, and motorcycles [10,11]. The superiority of the YOLOv8 model is demonstrated by comparing its performance accuracy and completion speed with other YOLO versions, as illustrated in Figure 1 [10].



	YOLOv8n	YOLOv8S	YOLOv8m	YOLOv8l	YOLOv8x
Params (Million)	2	9	25	55	90
Accuracy (mAP@0.5)	47.2	58.5	66.3	69.8	71.5
CPU Time (ms)	42	90	210	400	720
GPUTime(ms)	5.8	6	7.8	9.8	11.5

Figure 1: The comparison of YOLO models [10]

This paper proposes an object detection-based model for effective video summarization. This model relies on identifying and utilizing key objects within videos, such as people, cars, bicycles, and buses, to generate concise summaries that retain the most meaningful content. The proposed method aims to deliver concise and contextually rich video summaries, focusing on the detected objects as the primary elements of interest. In addition, the proposed method presented an optimized step to reduce redundancy and make video summaries a more robust solution for complex and diverse video content. The key contributions of this paper are summarized as follows:

- The proposed framework identifies the video objects (e.g., people, vehicles) to preserve the most informative video content.
- The proposed framework integrates YOLOv8 for object detection, ResNet-50 for deep feature extraction, and HDBSCAN for clustering of object instances.
- The proposed framework produces a video summary with a length reduction of up to 95% while maintaining diverse video content.

The rest of the paper is organized as follows: Section 2 describes the Related work for existing techniques. Section 3 presents the methodology that explains the video summarization method.

Section 4 discusses the Experimental Setup. Section 5 presents the Results and Analysis. and finally, the last section is the conclusion.

2. Related Works

Many researchers have tried to summarize the video depending on the object's detection. First, the video summarization framework inputs the video and the object of interest (OoI). After that, the frames having OoI are detected using an object detection module. Finally, only the detected frames are combined to produce a video summary as an output.

M. S. Nair and J. Mohan [12] presented a method to detect key-frames for static summarization. The proposed method detects keyframes based on feature vectors extracted from multiple pre-trained Convolutional Neural Network models (Multi-CNN). The features are extracted using four pre-trained models of CNN. These vectors are fed to the Sparse Autoencoder, which outputs a combined representation of the input feature vectors. The key frames of the input video are extracted based on combined feature vectors using a Random Forest Classifier. The method is evaluated using two datasets: VSUMM and OVP, based on user summaries present in the ground truth. The method achieved an average F-score of 0.83 on the VSUMM dataset and 0.82 on the OVP dataset, respectively.

X. Zhu et al. [13] improve object detection in drone images by introducing an additional prediction head for small objects, employing Transformer Prediction Heads (TPH) to enhance detection in dense scenes, and incorporating CBAM (convolutional block attention model) to prioritize key areas. It also applies data augmentation, multi-scale testing, and a self-trained classifier to boost accuracy, particularly for similar objects. While the model outperforms YOLOv5 by 7% and performs well on the VisDrone2021 dataset, it comes with increased computational costs due to the extra prediction head. The method achieved an average precision of 39.18% on the VisDrone2021 DET test challenge.

The enhanced YOLO model was presented by H.-K. Jung and G.-S. Choi [14]. They altered the convolutional layers and optimized the activation and loss functions of the original YOLOv5, leading to a 0.9% increase in mAP (mean average precision) scores. It was trained on a dataset of 3360 images captured under diverse environmental conditions, including varying weather and lighting, to ensure robustness. The model's performance was validated against the original YOLOv5, showing faster convergence and superior accuracy in object detection tasks, especially in complex environments.

H. B. Ul Haq et al. [15] proposed a video summarization framework that addresses video summarization challenges by focusing on objects of interest (OoI) chosen by the user. It works by collecting frames where the selected object appears. The framework has three main steps: selecting the OoI, detecting or locating the object using YOLOv3, and summarizing the video. It was tested on VSUMM, TVSum, and a custom dataset. The Overall accuracy of the VSUMM dataset is 99.6, and the accuracy of the TVSum dataset is 99.9 according to Accuracy. However, it has limitations in detecting small objects and relies heavily on specific objects of interest, which may reduce its effectiveness in wider applications.

Negi et al. in [16], The frames are processed locally, and only the relevant ones containing objects of interest are sent to the cloud, reducing bandwidth and unnecessary data transfer. It relies on YOLOv5 to detect objects of interest and extract frames, achieving high performance. The proposed model achieved a best mAP of 0.98899 and a best precision of 0.96926 with a recall of 0.93643. However, it has some limitations; first, the model was trained on a single computer, which could make it less effective with larger datasets, and second, its focus on specific objects may overlook other important details in the video.

M. Tahir et al. [17] presented a study that aims to quickly detect road traffic accidents in surveillance videos to reduce human and financial losses in smart cities. The proposed method

focuses on two key areas: identifying traffic accidents and creating privacy-protected video summaries of these events. YOLO, a deep learning model, is trained on synthetic and annotated data to summarize accident and non-accident events. The proposed method was tested on real-time CCTV (Closed-Circuit Television) footage, and the result shows that the method achieved an accuracy ranging from 55% to 85% in event accident detection. In addition, the summarized videos reduced video duration to 42.97% on average and were stored in an encrypted format to avoid untrusted access to sensitive event-based data.

H. B. Haq et al. [18] proposed a method that extracts frames based on the user's object of interest (OoI). Initially, the selection of OoI is done. The proposed method chooses the object from the repository and automatically throws out any unnecessary objects; the YOLOv3 is then utilized to discover the needed object. It achieves 98.7% accuracy and a 93.5% reduction in time when processing the entire video on the SumME dataset, and 97.5% accuracy with a 67.3% reduction in time on the self-created dataset. It generates summaries based on user-specific needs, providing better context around the objects. However, the model is limited to pre-recorded videos and struggles with low-resolution or distorted frames.

F. Alharbi et al. [19] produced an approach that uses the InceptionV3 model to extract features, ensuring the best representation after thorough analysis. A custom encoder-decoder setup with convolutional and transposed blocks captures complex patterns in video frames. The addition of a channel attention mechanism emphasizes important details, helping the model focus on key features for better predictions. Weights enhance critical features and improve performance, as shown in extensive testing. It attains an average F-Score of 51.8 on SumMe and 61.5 on TVSum. However, using InceptionV3 and channel attention may demand high computational resources, limiting its use on lightweight devices.

P. Kadam et al. [20] presented a method containing two phases. Phase one creates a general summary by selecting keyframes from the entire video, forming the basic structure. Phase Two takes the output from Phase One (the frames and detected objects) and customizes it based on user queries to make the summary more personalized and relevant. The strength of the proposed model is its ability to generate personalized video summaries according to user preferences, using techniques like object detection, tracking, and semantic analysis. It handles large video data efficiently, provides relevant summaries, and adapts to various content types. The result of the F-score is 58.6. Table 1 provides a comparative analysis of video summarization methods based on the object of interest described in the previous section.

Table 1: A comparative analysis of video summarization methods based on the object of interest

Re.	Method name	Year	Evaluation Metric	Datasets	Performance results
[12]	Multi-CNN	2021	F-score	VSUMM OVP	0.83 0.82
[16]	VS framework	2022	mAP	Lobby	0.98
[17]	YOLOv5-based VS framework	2023	mAP	real-time CCTV footage	0.95
[18]	DL method for VS based on (OoI)	2023	F1 score	SumMe Self-Created Dataset	0.94 0.97
[19]	SAVS-Net	2024	F1 score	SumMe TVSum	51.8 61.5
[20]	KF Extraction Based Single View Query Dependent VS	2024	F1 score	SumMe TVSum	0.77 0.72

Although Table 1 offers a comparative overview of similar works, a direct experimental comparison using the same dataset and metrics was not conducted due to variations in implementation availability and differing experimental settings. Future work will involve implementing benchmark models under a unified setting for empirical comparison.

3. Proposed Framework

The general structure of the proposed framework consists of several phases, as shown in

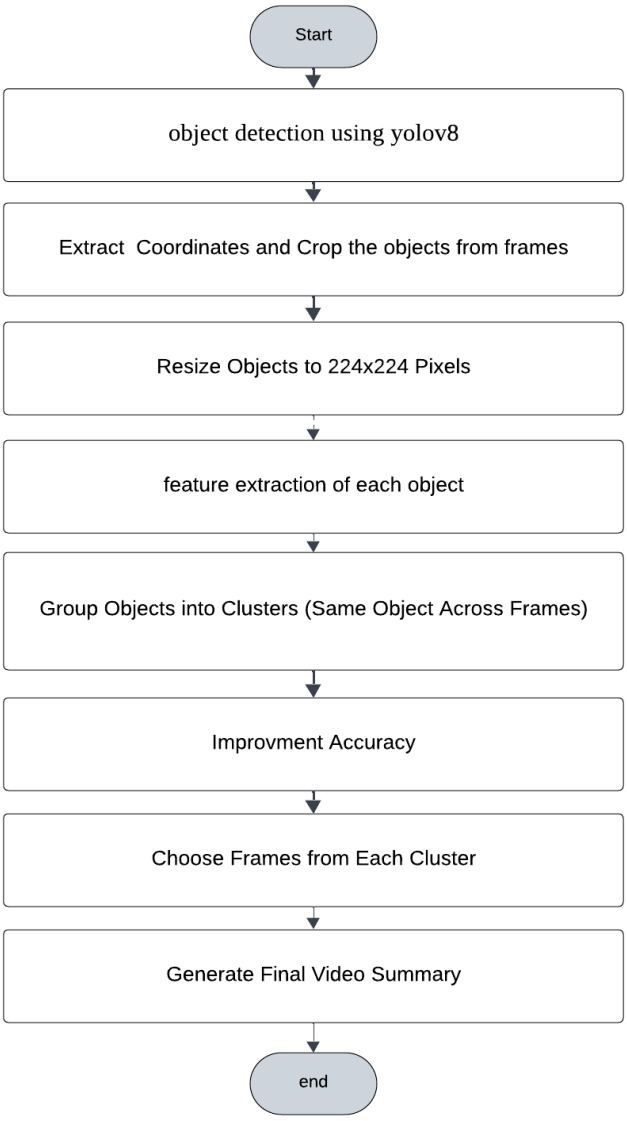


Figure 2: The general structure of the proposed framework

3.1 Object Detection

The used YOLOv8 learns to detect objects in the video frames based on the COCO dataset. The frames are extracted from each video and passed to the trained YOLOv8 model. The objects are identified in each frame by bounding boxes around detected objects. YOLOv8 was chosen for its improved accuracy, speed, and robustness over earlier versions like YOLOv5 and YOLOv3. It provides a good balance between speed and accuracy, and is widely used in recent

studies for its effectiveness in handling diverse video frames. An example of frame object detection is shown in Figure 3.

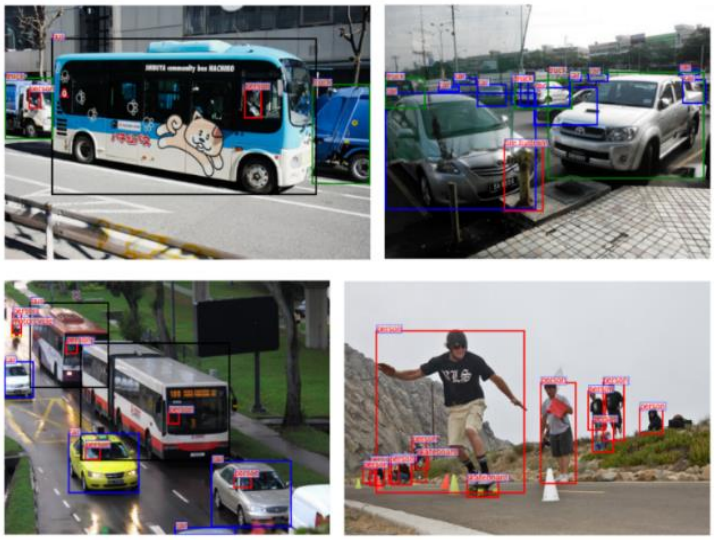


Figure 3 : Object Detection Using YOLOv8 on some images.

3.2 Objects Crop

After detecting the objects for each frame, the objects from individual video frames are cropped based on the coordinates of the bounding boxes for each identified object and saved as separate images. This step ensures that feature extraction focuses solely on the relevant parts of a video frame, eliminating distractions from unnecessary background details. This isolation enhances the accuracy and quality of the extracted features, resulting in more meaningful and consistent clusters that rely on object-specific information. Without cropping, background elements can interfere with the object's clustering, leading to incorrect grouping based on shared background features rather than the objects themselves. Moreover, processing entire frames demands more computational resources due to the larger and more complex data, making cropping essential for efficient and accurate video summarization.

3.3 Resize Cropped Images

The cropped images are resized to a consistent size to ensure that the extracted features have equal dimensions across all frames. Images of different sizes can create inconsistencies, making it harder for the model to detect patterns and group similar objects together. The standard size used for object images is 224×224 pixels, which matches the input layer format of the ResNet-50 network used for feature extraction. This standard input size ensures consistent and reliable feature representation. Additionally, using the same size across all feature extractors helps maintain uniformity in the output feature dimensions, facilitating accurate clustering and comparison.

3.4 Feature Extraction

The ResNet-50 model is a variant of the ResNet convolutional neural network consisting of four main parts: the convolutional layers, the identity block, the convolutional block, and the fully connected layers. The convolutional layers are responsible for extracting features from the input image. ResNet50 has been trained on the ImageNet dataset, which contains over 14 million images and 1000 classes. The ResNet50 is utilized in the proposed framework as a deep

feature extractor. This means that only convolutional layers are used from the ResNet-50 model, followed by max-pooling layers to reduce the spatial dimensions of the feature maps while preserving the most important features.
Although ResNet-50 may capture robust general features, it may not be optimized for specific tasks requiring finer or contextually particular representations.

The proposed framework also employs Histograms of Oriented Gradients (HOG) to make features robust in the face of variable conditions or noisy data. HOG is an effective feature extraction method for texture features and can successfully capture details even under challenging conditions such as deformation, rotation, or changes in lighting. Combining deep features and HOG features increases the feature's robustness and clustering accuracy.

3.5 Objects Clustering

In this step, the cropped images related to the same object across video frames are grouped. Each cluster represents a distinct object in the video frames. This eliminates object redundancy and ensures each object appears once in the final video summary.

The proposed framework utilizes the HDBSCAN for object clustering. One of the most significant advantages of HDBSCAN over DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is its robustness. It's especially remarkable in heterogeneous mixtures of data. Like DBSCAN, it can model arbitrary shapes and distributions; however, unlike DBSCAN, it does not require the specification of an arbitrary and sensitive hyperparameter. An example of applying the HDBSCAN for object clustering is shown in Figure 4.

(a) **(b)**

Figure 4: Example of object clustering shows (a) Person wearing a white shirt clustering category from the Car rail crossing video and (b) Airplane clustering category from the St Maarten Landing video. those are objects cropped from different frames grouped together in the same cluster

3.6 Accuracy Improvement

This step aims to reduce the true negative samples resulting from the over-segmentation clustering. The clusters are retested by checking the similarity between the different clusters.
The comparison is made by selecting an image from each cluster and calculating the similarity between the Gabor features of the selected image. When the similarity ratio exceeds a specified threshold, which is determined experimentally, the two clusters are merged because they contain images of the same object. The threshold value of 3 was selected after empirical testing

with values from 1 to 5 and was found to achieve optimal clustering accuracy with minimal over-merging.

The clusters will be more accurate when merging and eliminating redundant clusters corresponding to the same object. This, in turn, positively impacts the overall precision of the output video summary.

3.7 Video Summary Creation

To produce the video summary, the frames that best reflect the object of each cluster are chosen and saved in a summarized video. Additional surrounding frames (four before and four after each selected frame) may be included for a more compact summary. This helps preserve temporal coherence and ensures smooth visual transitions between frames, minimizing any jarring effects caused by frame skipping. Currently, one representative frame is randomly chosen from each cluster to build a summarized video. In future work, the strategies based on object movement, uniqueness, or centrality within the cluster timeline can be adopted.

The implemented steps of the proposed framework are given in Algorithm 1.

Algorithm 1: The Proposed Video Summarization Framework

Input:
V: Input video file
Output:
VSm: Video Summarization
Initialization
Initialize Frame extraction rate (R =1 frame every 15 frames)
Initialize Resized object dimensions (S=224, 224)
Initialize Similarity threshold ST ($0 \leq 3.0 \leq 1$)
Initialize Frames per second for output video (FPS =30)
Initialize an empty set of frames to store extracted video frames (Frames).
Initialize an empty set of objects to store cropped and resized objects (Objects).
Initialize an empty set to store extracted features of the ResNet50 technique (FV_1).
Initialize an empty set to store extracted features of the HOG technique (FV_2).
Initialize an empty set of clusters to store the video clusters (VClusters).
Start
1. Load YOLOv8 model (pre-trained on COCO).
2. For every R frame f in video V:
3. Add f to Frames
4. Detect objects using YOLOv8
5. For each detected object:
6. Crop using the bounding box & Resize to S
7. Add to Objects
8. For each object in Objects:
9. Extract features with ResNet50 → FV_1
10. Extract features with HOG → FV_2
11. Cluster objects using HDBSCAN on FV_1 and FV_2 → VClusters
12. For each cluster C in VClusters:
13. Extract Gabor features from a sample in C
14. If the similarity between any two clusters≥ ST, merge the clusters
15. Randomly select a frame from each merged cluster and add it to VSm
16. Generate a summarized video from VSm at FPS
17. Return VSm

4. Experimental Results

An experimental analysis of object detection and video summarization is presented to demonstrate the effectiveness of the proposed framework.

The programming language used for the experiments is Python, and the development environment is PyCharm. All tests and experiments were conducted on a computer with an Intel Core i7 processor (12th generation) and 16 GB of RAM, providing the necessary computational resources for the tasks.

4.1 Dataset

About eight video files have been chosen from the SumMe dataset and are available at [21]. The SumMe dataset was selected for evaluation because it is widely used for benchmarking video summarization methods. It contains a variety of real-world, user-generated videos covering different scenes and activities, making it suitable for assessing the generalization and robustness of summarization approaches. The list of video files is presented in Table 2 for review.

Table 2: List of selected video files from the SUMME dataset

No.	Video Name	Duration in Seconds	No. of Frames	Scenarios
V1	Air Force One	179	4494	Focuses on the arrival and presence of an airplane, highlighting travel-related scenes and aviation.
V2	Car rail crossing	169	5075	A group of people were on the road when an accident occurred. It happened because a car got stuck while crossing a railway track as a train approached. The group quickly came together to try to resolve the situation and fix the problem
V3	Jumps	38	950	Show someone wearing a helmet, sliding, and jumping into a pool.
V4	Kids playing in leaves	106	3187	Children playing with autumn leaves, capturing moments of pure joy and carefree fun.
V5	Playing ball	103	3119	The video shows a person throwing a small ball to a crow and a dog, and the animals play with it.
V6	St Maarten Landing	70	1751	The video highlights the dramatic approach and landing of an airplane at St. Maarten Airport, emphasizing how closely the plane passes by spectators.
V7	Statue of Liberty	154	3863	The video captures scenes of the Statue of Liberty, highlighting its surroundings and featuring people who came to see it.
V8	Uncut Evening Flight	322	9672	A continuous shot of an aircraft in motion, highlighting its smooth and uninterrupted journey.

4.2 Metrics

The performance of the proposed framework is assessed using the F-score and Accuracy. The mathematical formulas for these evaluation metrics are as follows [22, 23]:

$$\text{Precision (P)} = \frac{TP}{(TP + FP)} \quad (1)$$

$$\text{Recall (R)} = \frac{TP}{(TP + FN)} \quad (2)$$

$$\text{F-Score} = \frac{2 \times P \times R}{P + R} \quad (3)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \quad (4)$$

Where TP is True Positive, FP is False Positive, TN is True Negative, and FN is False Negative [22, 23].

4.3 Results and Analysis

Object detection accuracy is an essential feature of a practical video summarization experiment. It involves evaluating how accurately objects are detected and the speed of the object detection process.

The YOLOv8 model showcases remarkable object detection capabilities, achieving nearly flawless accuracy levels. Its performance is highly reliable and consistently produces precise and dependable results in detecting objects.

In terms of execution time, the YOLOv8 model outperforms other models used in object detection, such as Faster Region-Convolutional Neural Network (Faster-RCNN) and Single Shot Detector (SDD) models, as shown in Figure 5.

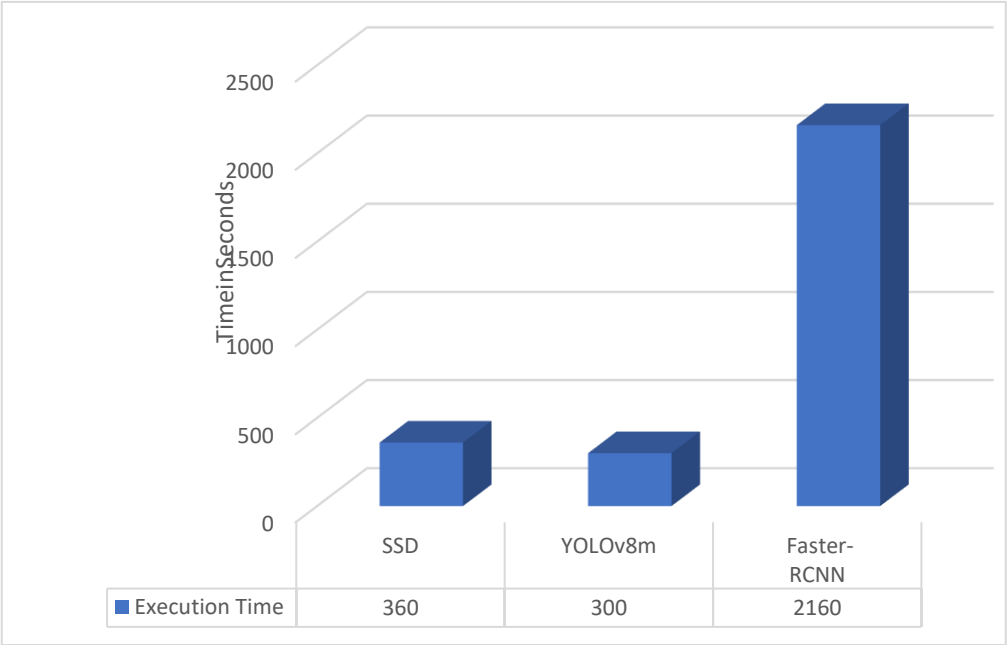


Figure 5: The execution time comparison of yolov8m with other

Finding the best feature representation is essential to achieving optimal clustering results. As mentioned earlier, the deep features of the ResNet-50 model and HOG features are combined to increase the clustering accuracy and robustness.

The clustering process's performance was assessed using HOG features alone, ResNet-50 features alone, and a combination of both techniques, to provide a clear overall indication of the feature combination efficiency.

The Playing ball video (V5) was selected to evaluate the effectiveness, and the result is shown in Figure 6.

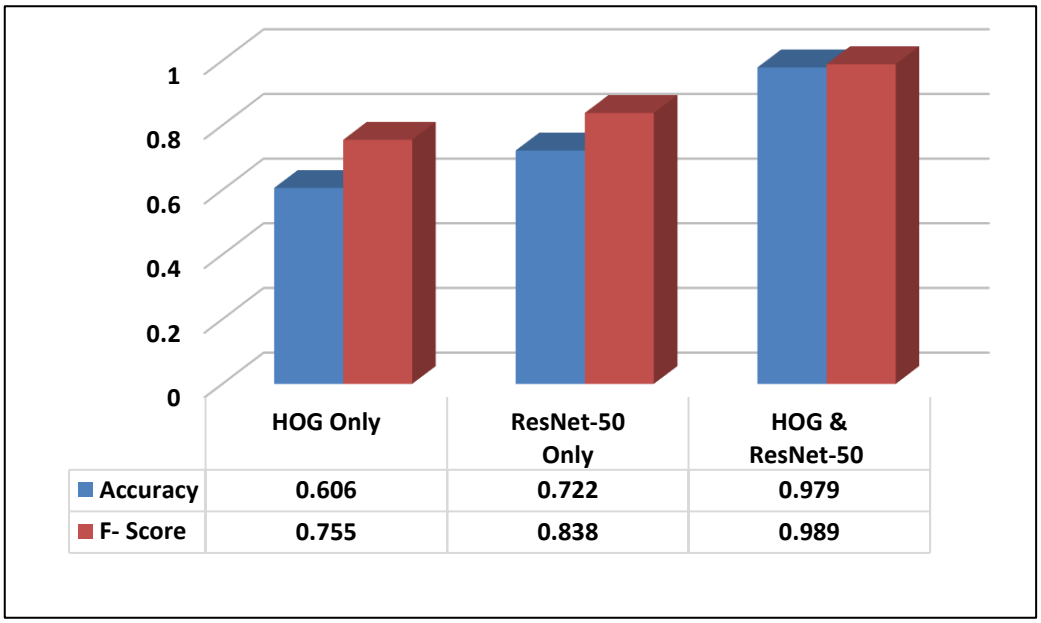


Figure 6: Impact of features used in the proposed

Referring to Figure 6, the accuracy rates indicate that using HOG or ResNet-50 features separately resulted in lower accuracy than integrating both techniques, achieving the highest accuracy. This is consistent with the expected results.

The clustering performance has been improved by eliminating redundant clusters corresponding to the same object, leading to more accurate detection. The two clusters are merged if the similarity ratio between the features of the selected image is greater than a specific threshold. An experiment was performed by changing threshold values until the best overall performance was achieved. The threshold was set to 3. The performance evaluations are reported in Table 3 according to Recall and Precision, while Table 4 presents the results based on Accuracy and F-Score.

Table 3: The performance of the clustering process according to Recall and Precision.

Videos	Before the Accuracy Improvement Step		After the Accuracy Improvement Step	
	Recall	Precision	Recall	Precision
V1	1	0.576	1	0.803
V2	1	0.760	1	0.939
V3	1	0.917	1	0.838
V4	1	0.826	1	0.831
V5	1	0.978	1	1
V6	1	0.930	1	0.974
V7	1	0.966	1	0.990
V8	1	0.624	1	0.816
Average	1	0.8221	1	0.8988

Table 4: The performance of the clustering process before and after accuracy improvement

Videos	Before the Accuracy Improvement Step		After the Accuracy Improvement Step	
	Accuracy	F- Score	Accuracy	F- Score
V1	0.575	0.731	0.801	0.891
V2	0.761	0.864	0.941	0.969
V3	0.918	0.957	0.837	0.912
V4	0.8267	0.905	0.831	0.908
V5	0.979	0.989	1	1
V6	0.931	0.964	0.974	0.987
V7	0.967	0.983	0.991	0.995
V8	0.625	0.769	0.816	0.899
Average	0.8228	0.8952	0.8988	0.9451

Figures 7 and 8 summarize the overall accuracy of the clustering process before and after accuracy improvement in a graphical representation.

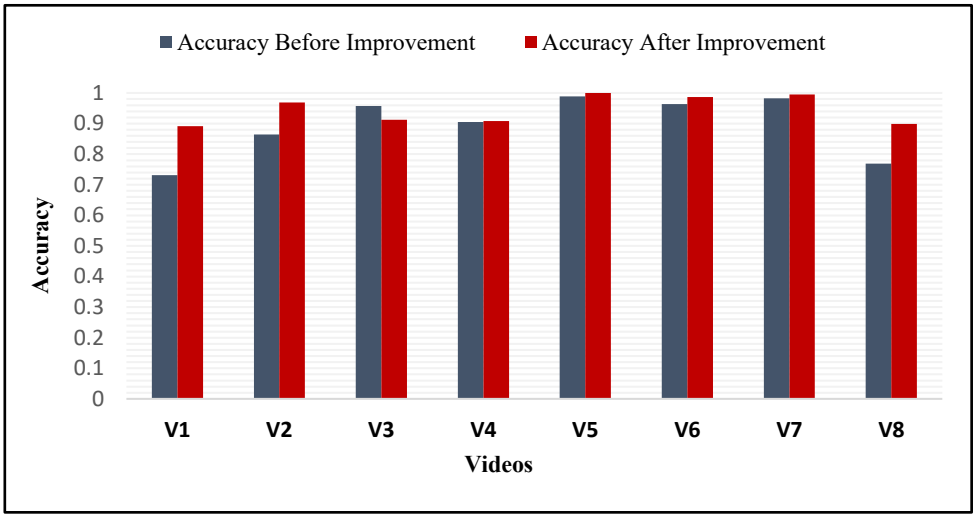


Figure 7: The Accuracy after the

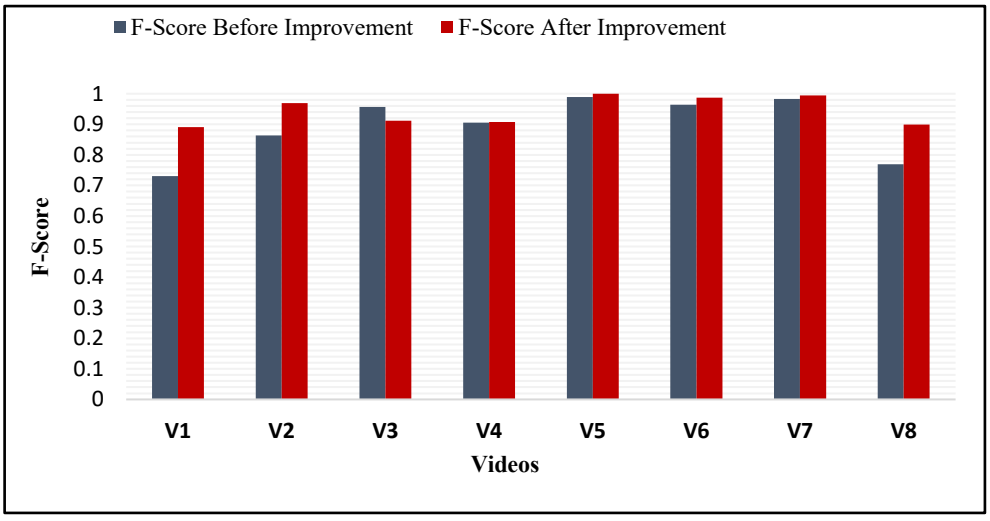


Figure 8: The F-Score after improvement step

The nearest value to one is the best accuracy of object clustering. In Figure 8, the clustering accuracy after applying the improvement step increased from 0.82 to 0.89. The improvement step helps to recognize objects in different video frames in the presence of sudden brightness variations and illumination changes that represent the main sources of misleading detection. Following the improvement step, V3's accuracy was relatively low. The errors were primarily caused by the rapid movement of the camera and the resulting blurred frames. For instance, the frames are significantly blurred in clusters one and four, as illustrated in Figure 9. Therefore, these frames were combined and treated as a single cluster.

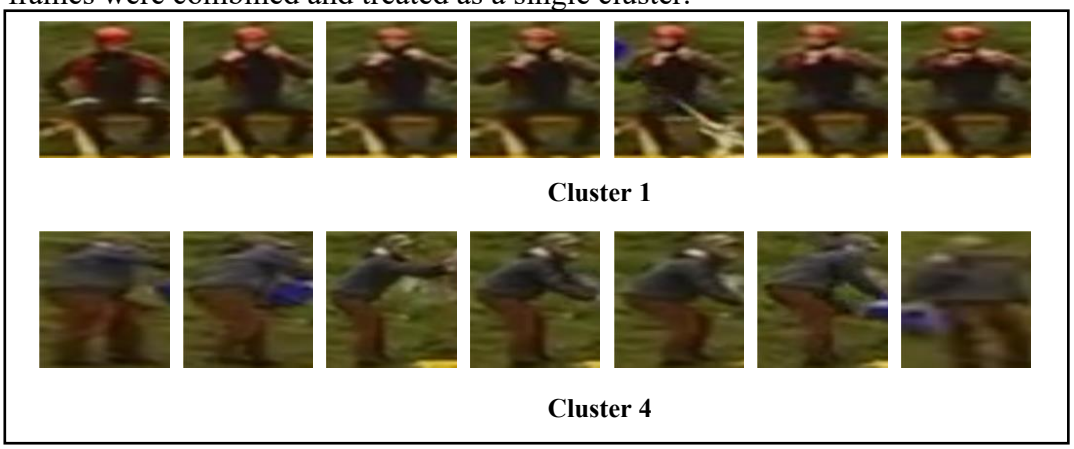


Figure 9: Example of incorrect merging of two clusters.

In addition to motion blur, other challenges affecting performance include object occlusion and small object size. Occluded or distant objects were sometimes mis detected or missed entirely, and abrupt visual changes led to fragmented clustering. These cases highlight practical limitations in detection and clustering under complex video conditions. Figure 10 represents an example of occluded objects and small objects. However, these limitations often have minimal impact on the final summary, as occluded objects frequently appear in other unobstructed frames, and very small objects are typically not central to the scene's main content.

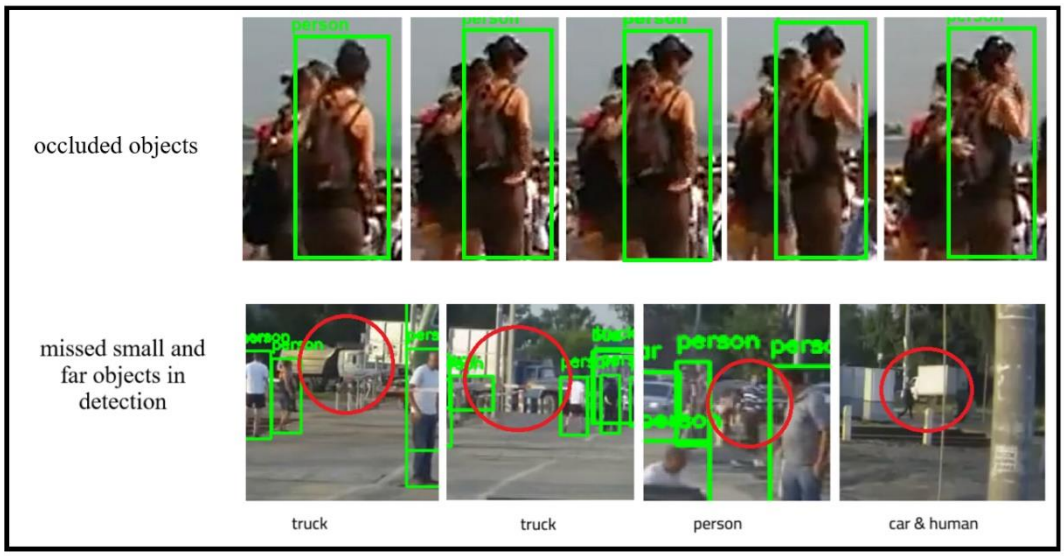


Figure 10: This Figure shows some cases of occluded objects and small objects that can cause failure in detecting objects.

The proposed framework captures frames containing target objects to produce a video summary. The result shows that the proposed framework distinguishes between objects and includes the most important frames for the resulting video summary. Table 5 presents the original videos and their summarized versions, including their time durations. The effectiveness and high performance of the proposed algorithm were clear. It significantly reduced the duration of the original video.

Table 5: Time duration of original videos and their summarized version.

Videos	Video Duration in Seconds	Summarized Duration in Seconds	Saved Time in Seconds
V1	179	1.93	98.92 %
V2	169	17.2	89.82 %
V3	38	0.8	97.89 %
V4	106	5.93	94.40 %
V5	103	1.53	98.51 %
V6	74	1.66	97.75 %
V7	70	4.93	92.95 %
V8	154	8.33	94.59 %
Average	**111.62**	**5.288**	**95.60%**

The proposed method is noteworthy because it differentiates between various types of objects. For instance, it not only identifies a person but also distinguishes between different individuals in the video. This is in contrast to the majority of the other strategies that focus on the general detection of objects rather than one specific object. Figure 11 illustrates examples of detecting different types of objects.

Figure 11: Example of detection of different types of person objects

The proposed method generates a video summary that highlights all detected objects. Additionally, it can create a summary for a specific predefined object. This demonstrates that the proposed method generates a video summary by emphasizing objects of interest instead of eliminating unnecessary frames and scenes. For instance, Figure 12 presents an example of a video summary featuring an airplane object from video V6.

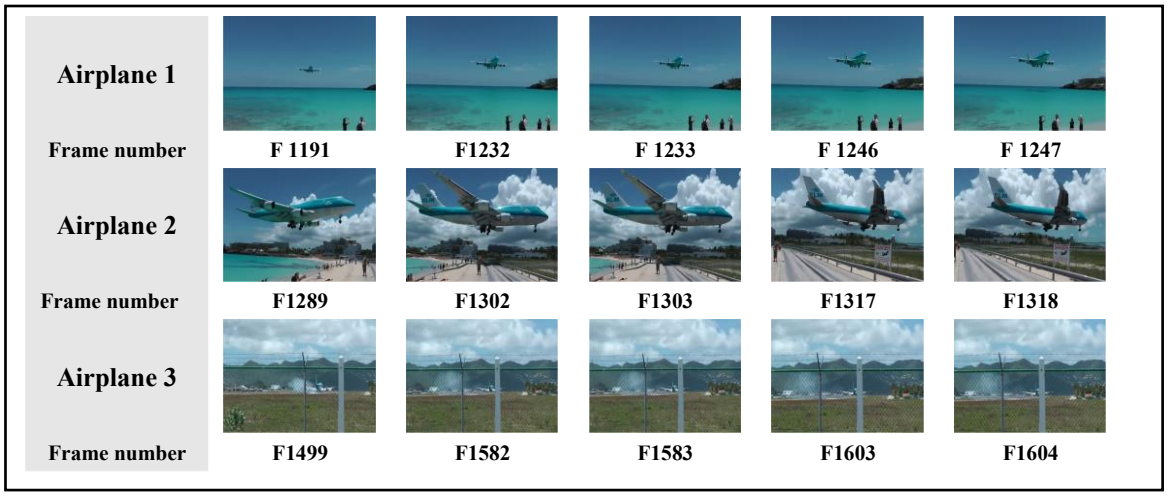


Figure 12: Example of video summary featuring an airplane object from video V6. highlights key appearances of the airplane at different timestamps.

Computational complexity is an important aspect of effective video summarization methods, so this point should be emphasized. The execution time was measured by the time required for object detection using Yolo, feature extraction using both HOG and ResNet50, and clustering using HDSCAN. Table 6 shows the execution time (for V3 as an example) for different stages of the proposed framework.

Table 6: Execution time of the proposed framework.

Video	YOLOv8	Execution Time in Seconds HOG and ResNet50	HDBSCAN
	For one frame 0.436 For all selected frames 27.9	For one frame 0.239 For all selected frames 15.3	- For all selected frames 0.025
Total		**43.225**	

It can be inferred from the time required that the proposed framework had a low execution time, making it practically applicable to various video summarization applications, such as surveillance and educational videos.

The computational complexity of the proposed framework grows mainly with the number of processed frames (n). Object detection using YOLOv8 runs once per frame, so it has a time complexity of O(n). Feature extraction and clustering depend on the number of detected objects (m), with clustering typically taking around O(m log m) time. Since the system samples one frame every 15 frames, the total processing time stays manageable even for longer videos. Memory use also increases based on how many objects are detected and saved.

5. Conclusions

This paper introduces a reliable framework for video summarization that integrates YOLOv8 for object detection, ResNet-50 for feature extraction, and HDBSCAN for clustering. Key insights reveal that combining HOG with ResNet-50 improves feature robustness while adding a step to merge similar clusters reduces redundancy and enhances the accuracy of the video summary. A major contribution of this study is the enhancement step, where clusters are re-evaluated by analysing the similarity of their features. Highly similar clusters are merged to address over-segmentation and enhance the accuracy of the final summary. Another significant contribution of this work is demonstrating that the method does not require users to select specific objects for processing, making it more flexible and user-independent. Experimental results on the SumMe dataset show that combining HOG with ResNet-50 improves the robustness of feature extraction, leading to better clustering accuracy. These improvements work together to enhance the framework's ability to create concise and meaningful video summaries. Although the method works well, it has challenges, like being sensitive to video distortions and pointing to areas for future improvement. While the current implementation processes frames at intervals (1 per 15), making it suitable for offline summarization, optimization using lighter backbones or edge-processing techniques could bring this method closer to real-time applicability.

References

[1] T. Psallidas and E. Spyrou, "Video Summarization Based on Feature Fusion and Data Augmentation," *Computers*, vol. 12, no. 9, Art. no. 9, Sep. 2023.

[2] R. Savran Kızıltepe, J. Q. Gan, and J. J. Escobar, "A novel keyframe extraction method for video classification using deep neural networks," *Neural Comput & Applic*, vol. 35, no. 34, pp. 24513–24524, Dec. 2023.

[3] E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris, and I. Patras, "Video Summarization Using Deep Neural Networks: A Survey," *arXiv*: arXiv:2101.06072, Sep. 27, 2021.

[4] V. Tiwari, C. Bhatnagar "A survey of recent work on video summarization: approaches and techniques "*Multimedia Tools and Applications*, Jan, 28, 2025. [Online]. Available: https://link.springer.com/article/10.1007/s11042-021-10977-y.

[5] H. Y. Adel, R. M. Elmasry, and M. A.-M. Salem, "Object-Based Video Archive Summarization," in *2023 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, Sep. 2023.

[6] S. H. Mousa, N. M. Shati, and N. Sakthivadivel, "DeepRing: Convolution Neural Network based on Blockchain Technology," *Al-Mustansiriyah Journal of Science*, vol. 35, no. 2, pp. 61–69, Jun. 2024.

[7] M. A. Al-Bayati, "Deep Learning for Handwritten Digit Recognition System: A Convolution Neural Network Approach," *FPA*, vol. 17, no. 2, 2025.

[8] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," *Multimed Tools Appl*, vol. 82, no. 6, pp. 9243–9275, Mar. 2023.

[9] A. Sharba and H. Kanaan, "Improving Tiny Object Detection in Aerial Images with Yolov5," *Journal of Engineering and Sustainable Development*, vol. 29, no. 1, pp. 57–67, Jan. 2025.

[10] M. Yaseen, "What is YOLOv8: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector," *arXiv*: arXiv:2408.15857 Aug. 28, 2024.

[11] M. Sohan, T. Sai Ram, and Ch. V. Rami Reddy, "A Review on YOLOv8 and Its Advancements," in *Data Intelligence and Cognitive Informatics*, I. J. Jacob, S. Piramuthu, and P. Falkowski-Gilski, Eds., Singapore: Springer Nature, 2024, pp. 529–545.

[12] M. S. Nair and J. Mohan, "Static video summarization using multi-CNN with sparse autoencoder and random forest classifier," *SIViP*, vol. 15, no. 4, pp. 735–742, Jun. 2021.

[13] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios," in *2021 IEEE/CVF*

International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada: IEEE, Oct. 2021.

[14] H.-K. Jung and G.-S. Choi, "Improved YOLOv5: Efficient Object Detection Using Drone Images under Various Conditions," *Applied Sciences*, vol. 12, no. 14, Art. no. 14, Jan. 2022.

[15] 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, no. 1, p. 7453744, 2022.

[16] A. Negi, K. Kumar, P. Saini, and S. Kashid, "Object Detection based Approach for an Efficient Video Summarization with System Statistics over Cloud," in *2022 IEEE 9th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, Dec. 2022.

[17] M. Tahir, Y. Qiao, N. Kanwal, B. Lee, and M. N. Asghar, "Privacy Preserved Video Summarization of Road Traffic Events for IoT Smart Cities," *Cryptography*, vol. 7, no. 1, Art. no. 1, Mar. 2023.

[18] H. B. Haq, W. Suwansantisuk, and K. Chamnongthai, "An Optimized Deep Learning Method for Video Summarization Based on the User Object of Interest," *International Journal of Advanced Computer Science and Applications*, vol. 14, Jan. 2023.

[19] F. Alharbi, S. Habib, W. Albattah, Z. Jan, M. D. Alanazi, and M. Islam, "Effective Video Summarization Using Channel Attention-Assisted Encoder–Decoder Framework," *Symmetry*, vol. 16, no. 6, Art. no. 6, Jun. 2024.

[20] P. Kadam, D. Vora, S. Patil, S. Mishra, and V. Khairnar, "Behavioral profiling for adaptive video summarization: From generalization to personalization," *MethodsX*, vol. 13, p. 102780, Dec. 2024.

[21] "Papers with Code - SumMe Dataset." Accessed: Dec. 14, 2024. [Online]. Available: https://paperswithcode.com/dataset/summe.

[22] E. Hato, Z. S. Abduljabbar, and Z. J. Ahmed, "Comparative Analysis for Bag of Features (BoF) Performance," *Iraqi Journal of Science*, pp. 4606–4622, Aug. 2024.

[23] A. H. Sadhin, S. Z. M. Hashim, H. Samma, and N. Khamis, "YOLO: A Competitive Analysis of Modern Object Detection Algorithms for Road Defects Detection Using Drone Images," *Baghdad Sci.J*, vol. 21, no. 6, p. 2167, Jun. 2024.