iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** III **Month of publication:** March 2025

DOI: https://doi.org/10.22214/ijraset.2025.67254

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue III Mar 2025- Available at www.ijraset.com

Video Summarization Using Deep Learning

K Mohan Kumar[1], K Sai Teja Reddy[2], Malathesh C[3], P Rajiv Dikshith[4], Nagaraj M[5]
Rao Bahadur Y Mahabaleswarappa Engineering College, India

***Abstract:** This project presents a Video Summarization Tool that processes and summarizes videos by identifying and emphasizing objects of interest. Implemented with Python, OpenCV, and Tkinter, the tool has a simple graphical user interface for video processing. The main functionality is based on the YOLOv3-tiny object detection model, which is optimized for quicker processing. The model, which is trained using OpenCV's DNN module, finds objects in video frames and outlines them with rectangles. The application analyzes videos by bypassing frames and resampling them, storing processed frames in an output video file. It reads and writes in several video formats and outputs the MP4 codec. The most prominent feature is the GUI based on Tkinter, enabling users to load input videos and see the output summary. The interface provides video selection and output buttons for ease of use by users with little technical knowledge.*

I. INTRODUCTION

Video summarization is the act of shortening a lengthy video into a concise, manageable version without losing the important information and main events. With the exponentially increasing amount of video content on websites such as YouTube, social media, and surveillance systems, manual watching and analysis are no longer feasible. Video summarization provides a solution to obtain the most important parts of a video. Machine learning (ML) has transformed video summarization through the use of automated techniques for determining significant events, objects, and scenes. Traditional approaches tend to be based on manual feature extraction, whereas ML-based methods enable systems to learn from data and apply it to new video inputs. Such models can examine a video's visual, audio, and contextual cues to decide what parts to keep. Extractive Summarization: This process picks out individual frames, scenes, or segments from the source video itself to form a summary. Machine learning algorithms, including supervised learning and clustering, are employed to recognize key frames or significant moments as per the relevance of the content. Most video summarization systems based on machine learning employ models such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformers in order to grasp the temporal and spatial relationships among video data. These models may be trained over large datasets of labeled video clips to forecast where in the video there are interesting actions, persons, or objects. With the increasing use of machine learning in video summarization, many applications have been developed, such as content recommendation systems, video indexing, video surveillance, and media editing. Efficient processing and summarization of video data are important in many areas, ranging from entertainment to security.

II. LITERATURE REVIEW

Video summarization is an active area of research in the field of computer vision and multimedia processing. The goal is to create concise and informative summaries of long videos, which can be useful in various applications such as media, entertainment, education, and security. This literature survey provides an overview of key techniques and advancements in video summarization.

A. Traditional Video Summarization Techniques

1) Key frame Extraction

One of the earliest methods for video summarization involves extracting keyframes that represent significant content in the video. Techniques such as shot boundary detection and clustering are used to identify keyframes. Examples: [Zhuang et al., 1998](https://ieeexplore.ieee.org/document/710889), [Hanjalic and Zhang, 1999](https://ieeexplore.ieee.org/document/796471).

2) Shot Boundary Detection :

This technique segments a video into shots, which are sequences of frames captured continuously by a single camera. Methods include histogram comparison, edge detection, and machine learning based approaches. Examples: [Lienhart, 2001](https://ieeexplore.ieee.org/document/905940), [Smeaton et al., 2010](https://ieeexplore.ieee.org/document/5504803). Machine Learning and Deep Learning Approaches Supervised Learning : Supervised learning techniques use labeled data to train models that can identify important segments in videos. Examples: [Gygli et al., 2014] (https://ieeexplore.ieee.org/document/6909637), [Song et al., 2015] (https://ieeexplore.ieee.org/document/7298710).

3) Unsupervised Learning

Unsupervised learning techniques do not require labeled data and instead rely on clustering and anomaly detection to identify key segments. Examples: [Zhao and Xing, 2014] (https://ieeexplore.ieee.org/document/6909638), [Mahasseni et al., 2017] (https://ieeexplore.ieee.org/document/8099791).

III. PROBLEM DEFINITION

As more video content is being produced and viewed, there is a requirement for effective ways to summarize videos by finding and emphasizing primary objects in the frames. The purpose of this project is to meet this requirement by creating a video summarization tool that utilizes the YOLOv3-tiny model for real-time object detection.

IV. METHODOLOGY

The project methodology of the video summarization includes a series of steps such as loading the YOLO model, video frame processing, object detection, and saving the summarized video.

1) Loading the YOLO Model: The pre-trained object detection model YOLOv3-tiny is loaded. The function loads the model weights, configuration files, and fetches the output layer names and class labels.

2) Processing the Video: The function opens the input video and extracts its properties, including frame width, height, and frames per second (FPS). An output video is initialized through a video writer to write the output video.

3) Object Detection: Each frame is converted into a blob and fed into the YOLO model for object detection. Detection is filtered based on confidence scores, and bounding boxes are drawn around detected objects.

4) Saving and Displaying the Video: Processed frames along with bounding boxes are saved to the output video file. Also, the frames are shown in a window, and the loop can be terminated by pressing the 'q' key.

5) Releasing Resources: Once all the frames are processed, the video capture and writer objects are released, and all OpenCV windows are closed.

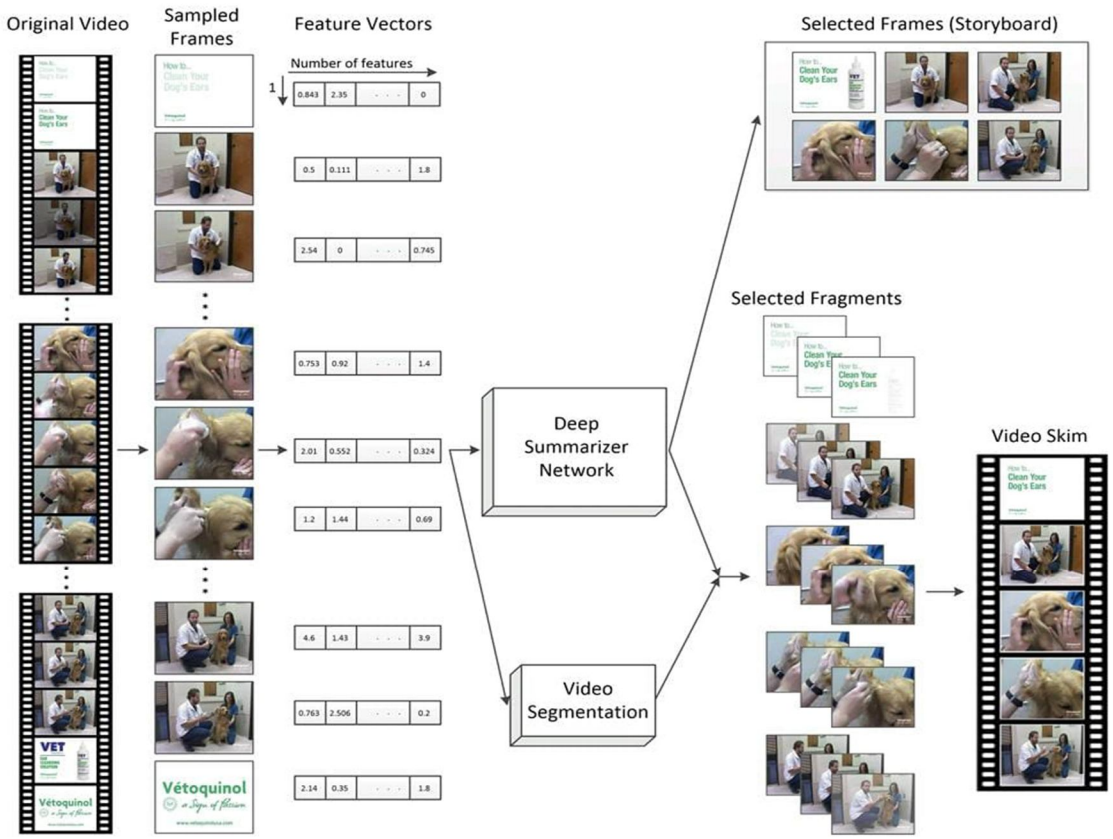


Figure 1 Video summarization workflow

V. RESULTS AND EVALUATION

The project effectively achieves a video summarization tool by utilizing the YOLOv3-tiny object detection algorithm. The tool has a friendly graphical user interface (GUI) to facilitate users in uploading a video, processing it to detect and mark objects, and display the summarized output. The functionalities of video upload, real-time video processing, object detection, generation of the output video, and video display are the main ones achieved. The tool effectively handles video frames by using the YOLOv3-tiny model, which is made to function faster with acceptable detection accuracy. The GUI is user-friendly and easy to use, giving unambiguous instructions and feedback to the user. The evaluation indicates that the tool satisfies the functional requirements, operates efficiently, and delivers accurate object detection and serves as a convenient solution for summarizing video content based on objects detected. In general, the project shows the effective incorporation of object detection into video processing pipelines, providing a useful tool for users to rapidly summarize and analyze video content.

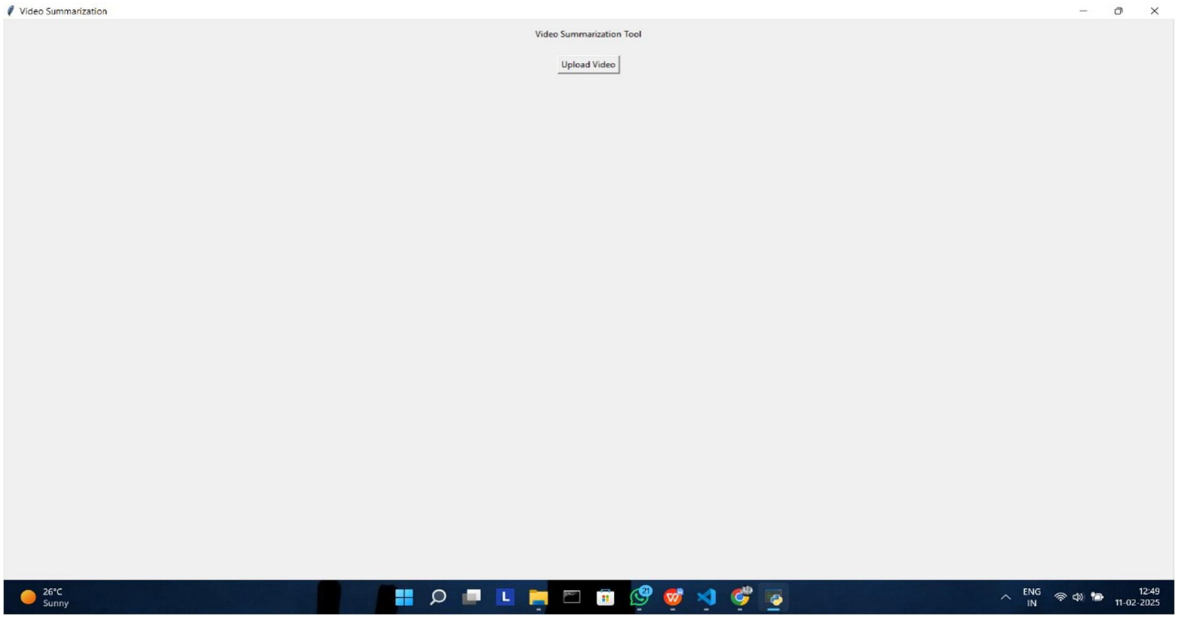


Figure 2 user screen

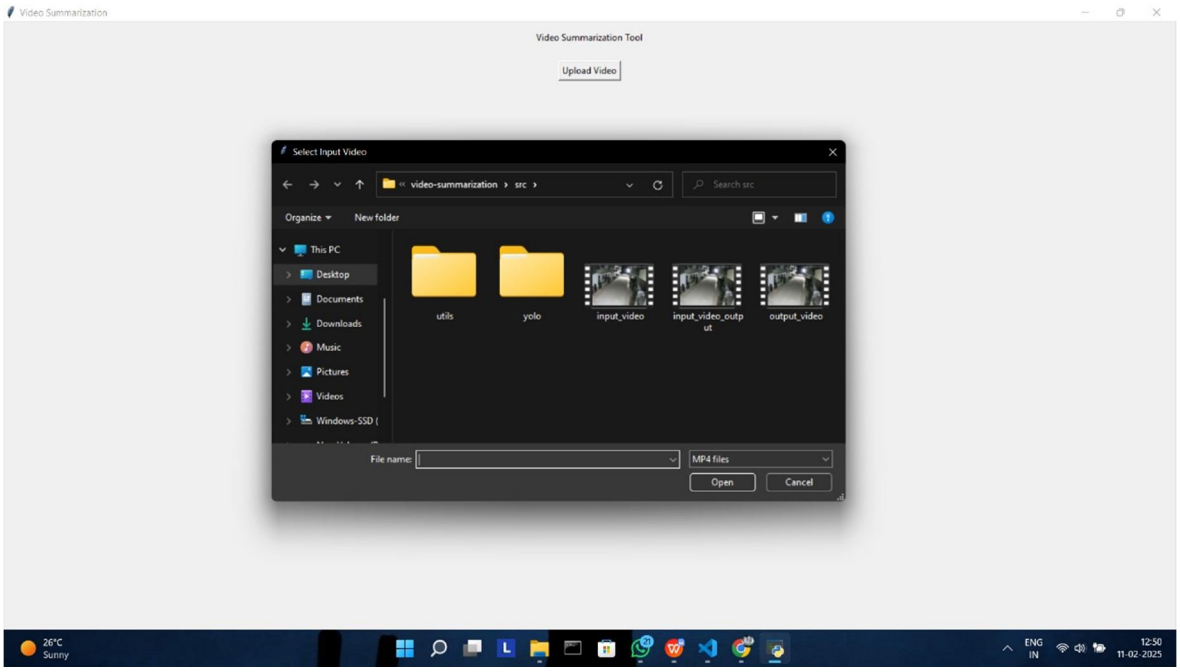


Figure 3 Upload a input video

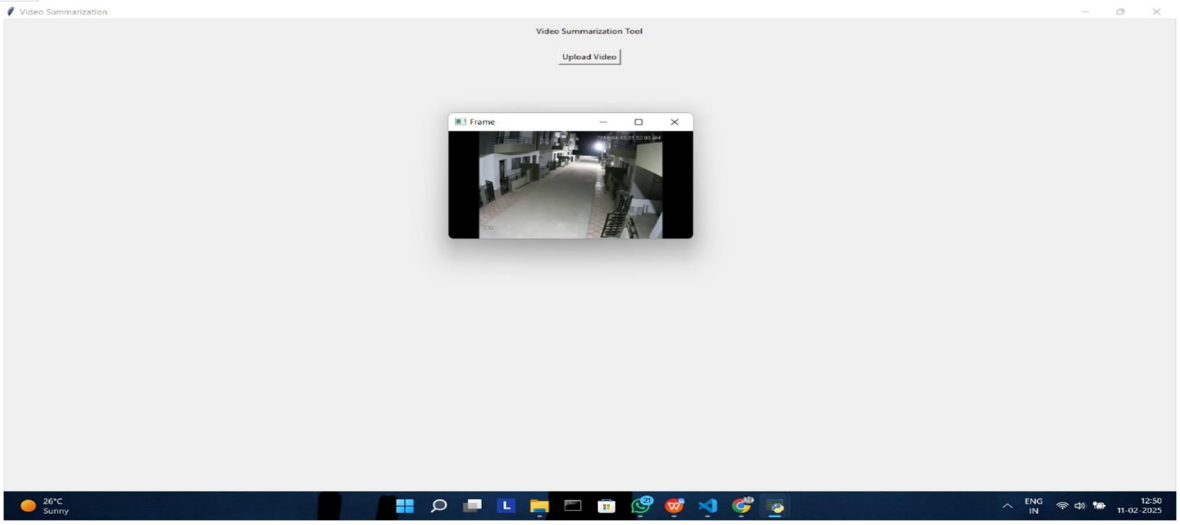


Figure 4 Video Processing

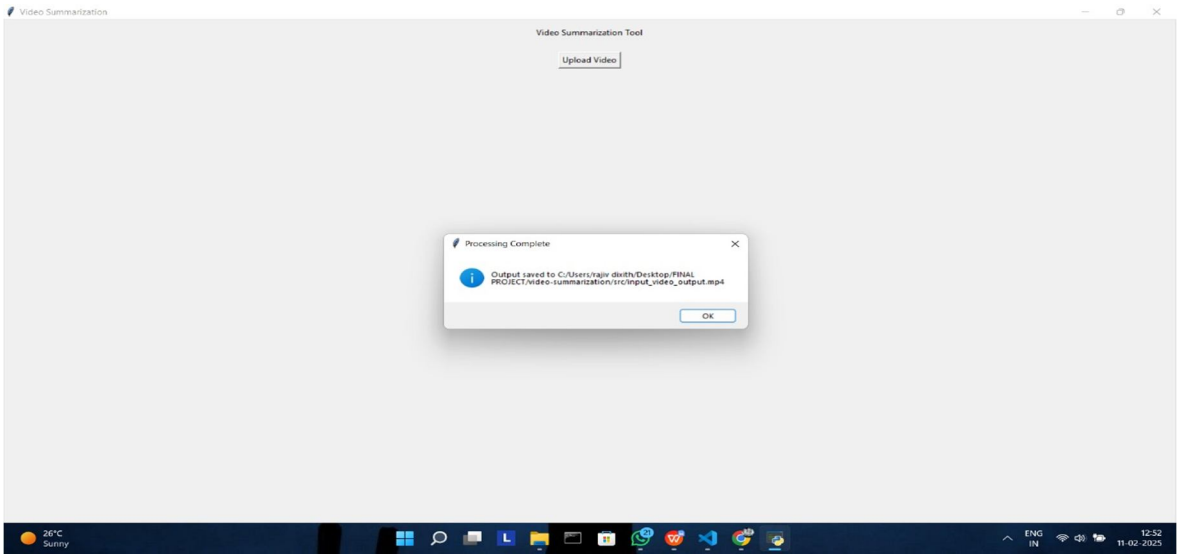


Figure 5 Video Summarizer

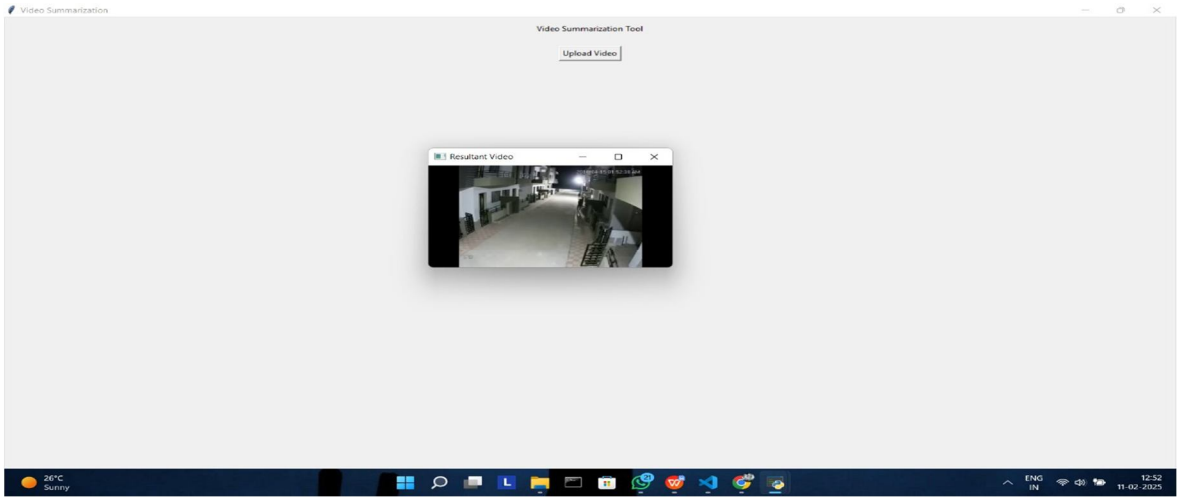


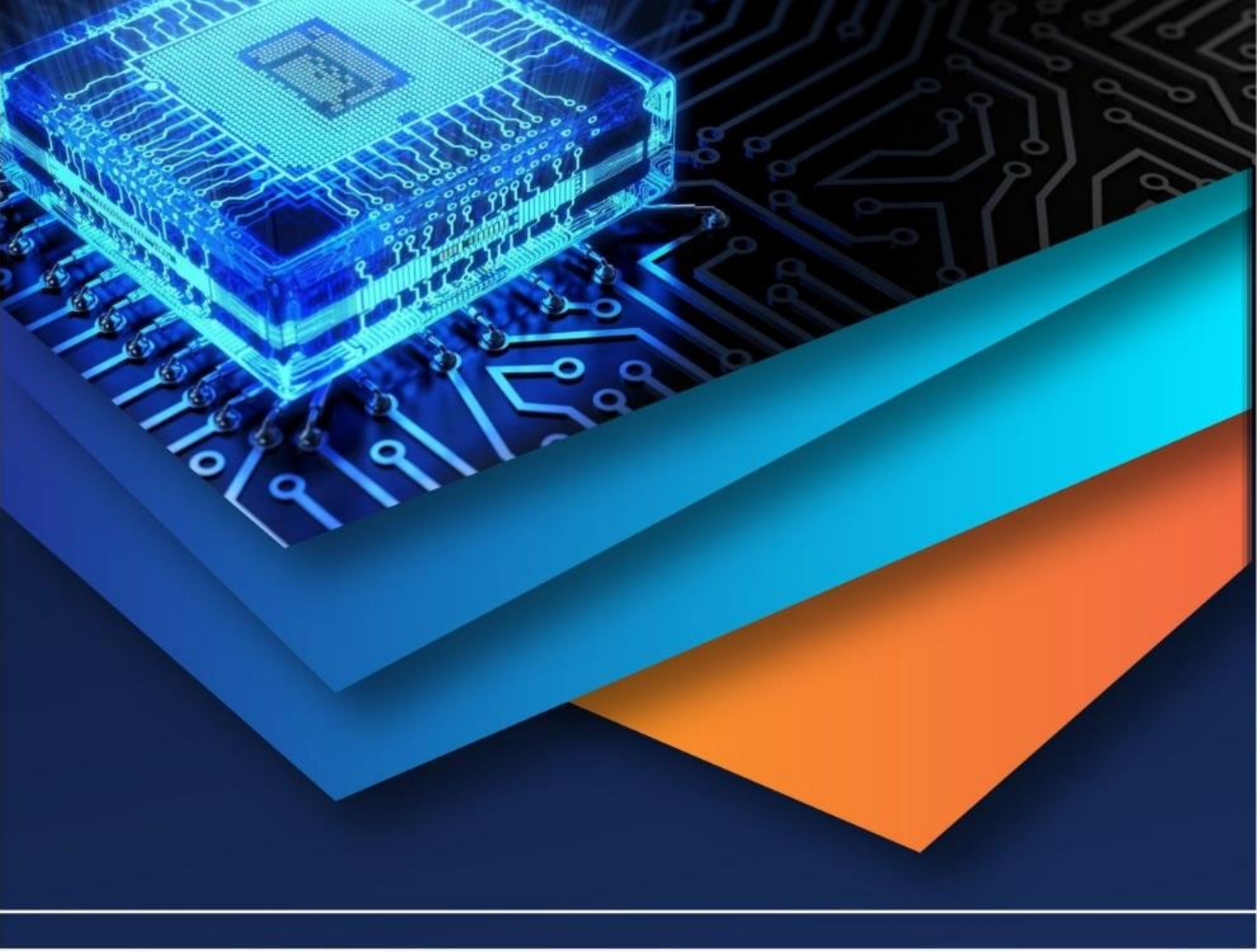
Figure 6 Resultant video

VI. CONCLUSION

The Video Summarization Tool project can efficiently overcome the obstacle of summarizing and processing videos well by identifying and highlighting target objects. Dependent on Python, OpenCV, and Tkinter, utilizing the YOLOv3-tiny model to detect objects precisely and in real-time, the tool helps highlight objects of interest. The project features an organized video processing technique with the addition of a graphical user interface that is easy to use, facilitating easy uploading, processing, and displaying of summarized output. The ability to handle different video formats, optimizing processing by frame skipping and resizing, and providing instant feedback through real-time display enhances its efficiency and usability. Comprehensive testing, including unit, integration, and performance tests, ensures the stability and reliability of the tool. Overall, the Video Summarization Tool offers a useful solution for applications such as surveillance, video analysis, and content abstraction, and thus is a useful and effective tool for users who want to summarize video content effectively and accurately.

REFERENCES

[1] U. Cisco. (2018). Cisco Annual Internet Report (2018–2023) White 923 Paper. [Online]. Available: https://www.cisco.com/c/en/us/solutions/ 924 collateral/executive-perspectives/annual-internet-report/white-paper-c11- 925 741490.html 926
[2] J. Lei, Q. Luan, X. Song, X. Liu, D. Tao, and M. Song, ‘‘Action 927 parsing-driven video summarization based on reinforcement learning,’’ 928 IEEE Trans. Circuits Syst. Video Technol., vol. 29, no. 7, pp. 2126–2137, 929 Jul. 2019. 930
[3] S. S. Thomas, S. Gupta, and V. K. Subramanian, ‘‘Context driven optimized 931 perceptual video summarization and retrieval,’’ IEEE Trans. Circuits Syst. 932 Video Technol., vol. 29, no. 10, pp. 3132–3145, Oct. 2019. 933
[4] C. Huang and H. Wang, ‘‘A novel key-frames selection framework for 934 comprehensive video summarization,’’ IEEE Trans. Circuits Syst. Video 935 Technol., vol. 30, no. 2, pp. 577–589, Feb. 2020. 936
[5] M. Ma, S. Mei, S. Wan, Z. Wang, D. D. Feng, and M. Bennamoun, 937 ‘‘Similarity based block sparse subset selection for video summarization,’’ 938 IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 10, pp. 3967–3980, 939 Oct. 2021.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY