Object Tracking and Saliency Mapping Using YOLO Segmentation

Apurva Madhavan Pillai
Computer Science
Clemson University
Clemson, SC, USA
pillai4@clemson.edu

Abstract—**This project focuses on combining object tracking and motion saliency visualization to enhance video analysis. Using a YOLO model with segmentation capabilities, the system detects and tracks objects in real time, assigning unique IDs and highlighting them with colored masks. Motion saliency is captured through frame differencing, which identifies areas of significant movement. These saliency maps are blended with the original video and YOLO annotations to create a visually enriched output. The result is a dynamic system that highlights both object activity and movement within the scene, making it useful for applications like wildlife monitoring, video surveillance, and action analysis.**

Index Terms—**YOLOv8, Object Detection, Saliency Mapping, Computer Vision, Dynamic Tracking, Image Segmentation**

I. Introduction

This project explores the integration of YOLO (You Only Look Once) object detection with optical flow saliency detection to enhance video processing capabilities. By leveraging YOLO's real-time object detection and the motion analysis provided by optical flow, the aim is to improve the accuracy and efficiency of tracking dynamic objects in video streams. This combination addresses challenges faced in various applications such as surveillance, traffic monitoring, autonomous driving, robotics, and sports analytics, where precise object detection and motion tracking are critical.

YOLO, a cutting-edge deep learning framework for real-time object detection, excels in identifying and localizing multiple objects within a scene with remarkable speed and accuracy. Optical flow, on the other hand, captures the pixel-level movement within video frames, offering valuable insights into motion patterns. By integrating these two technologies, the project seeks to develop a robust system capable of detecting and tracking objects even in complex and fast-changing environments, such as crowded public spaces or high-speed sporting events.

The synergy between YOLO's powerful detection capabilities and optical flow's nuanced motion tracking creates a system with the potential to outperform traditional methods in both precision and efficiency. Such a system can be instrumental in improving decision-making processes in real-time scenarios, like monitoring traffic conditions to reduce congestion, analyzing player movements in sports for strategic insights, or enhancing security measures in surveillance systems.

Furthermore, this initiative demonstrates the practical application of advanced computer vision techniques in addressing real-world problems. By implementing this hybrid approach, the project contributes to the growing field of automated video analysis, paving the way for innovations in intelligent systems that can better understand and interpret dynamic environments.

II. Methodology

A. You Only Look Once (YOLO)

The YOLO (You Only Look Once) series of models has evolved significantly over time. **YOLOv1** introduced the concept of real-time object detection by dividing the image into a grid and predicting bounding boxes and class probabilities for each grid cell. **YOLOv2** improved on this by using a more powerful network architecture, anchor boxes, and higher input resolution, which led to better accuracy. **YOLOv3** further enhanced the model by adding multiple detection layers to detect objects at different scales, using the Darknet-53 backbone. **YOLOv4** made significant improvements in both speed and accuracy, employing techniques like multi-scale training and data augmentation. **YOLOv5**, developed by Ultralytics, became widely used due to its ease of use, modularity, and real-time performance. **YOLOv6** focused on optimizing the model for edge devices and industrial applications, while **YOLOv7** introduced features like object tracking for dynamic scenes. The latest version, **YOLOv8**, improves performance further with better segmentation capabilities and enhanced feature extraction, making it suitable for both detection and segmentation tasks.

The model used in this project is the **YOLOv11n-seg** , designed specifically for efficient segmentation tasks. Its lightweight architecture allows it to perform real-time segmentation, even on devices with limited computational power. This makes it particularly suitable for applications like video analysis, where speed and accuracy are crucial.

We selected this model for its ability to efficiently track and segment objects in dynamic video scenarios, ensuring reliable performance while handling complex scenes.

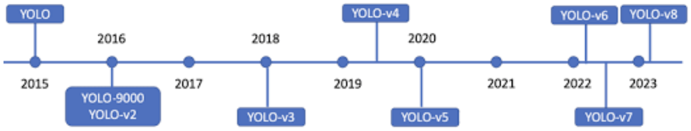

Fig. 1. Evolution of the YOLO model

It is a deep learning model designed for real-time object detection. It works by dividing an image into a grid and predicting bounding boxes and class probabilities for each grid cell. The architecture of YOLO is based on a convolutional neural network (CNN), and it processes the entire image in one pass, making it extremely efficient for real-time detection. In this project, we utilized a YOLO model with segmentation capabilities, which allows for pixel-level object segmentation along with bounding box detection.

Input Layer: The input to the YOLO model is a preprocessed image or frame from a video, which is passed through the network.

Convolutional Layers: These layers extract features from the image, learning patterns such as edges, textures, and object parts.

Detection Layer: The network divides the image into a grid and for each grid cell, predicts bounding boxes, class labels, and associated probabilities.

Segmentation Layer: In addition to detecting objects, YOLO can also segment objects by predicting pixel-wise masks that define the exact shape of the object.

Output Layer: The model outputs the class labels, bounding box coordinates, and segmentation masks for detected objects.

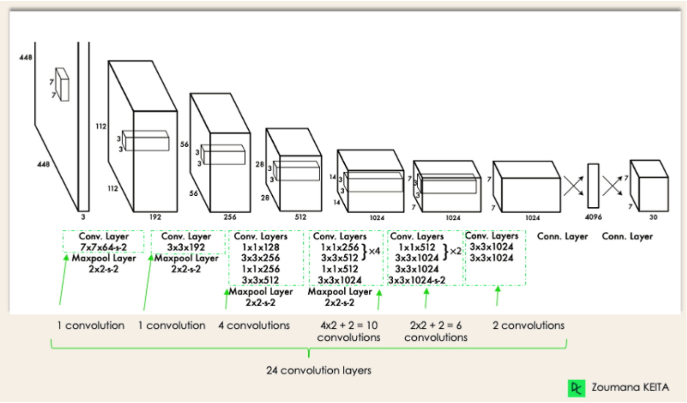

Fig. 2. YOLOv8 architecture

B. Saliency Detection

Saliency detection is the method of identifying the most noticeable or important areas in an image or video. These areas stand out due to significant differences in color, intensity, or motion. In video analysis, saliency helps the system focus on the key parts of a scene, such as moving objects or areas with noticeable changes, while ignoring less important parts.

It's important because it helps in identifying what to focus on in a scene, making it easier to track objects, analyze movement, and detect significant events. By highlighting the most relevant parts, it makes processing faster and more efficient, especially when dealing with large amounts of visual data. This method is commonly used in surveillance, video compression, and action recognition, where identifying key elements in the scene is crucial.

In this project, saliency detection is achieved through frame differencing between consecutive video frames. The following steps outline how saliency detection is achieved:

Frame Conversion to Grayscale: The first step is to convert both the current and previous video frames to grayscale. This simplifies the processing by removing color information, allowing the focus to be placed on intensity changes, which indicate motion in the scene. The grayscale conversion is done using the OpenCV cv2.cvtColor() function.

Frame Differencing: The difference between the current frame and the previous frame is calculated using the cv2.absdiff() function. This computes the absolute difference in pixel values between the two grayscale frames. The result highlights areas that have changed between the frames, which typically correspond to moving objects or regions of interest. This is achieved using the formula:

$$D(i,j) = |I_1(i,j) - I_2(i,j)|$$

Where:

- $D(i,j)$ is the difference at pixel location (i,j),
- $I_1(i,j)$ and $I_2(i,j)$ are the intensity values of the two consecutive frames at pixel (i,j).

Thresholding: After obtaining the difference between frames, the next step is thresholding. This process converts the difference image into a binary mask, where areas with significant change (motion) are highlighted. The cv2.threshold() function is used to set a threshold value, and areas where the difference exceeds this value are set to white (255), while all other areas are set to black (0). In this implementation, a threshold value of 25 is chosen to detect noticeable changes. To create a binary mask highlighting significant changes, a threshold is applied to the difference image. This can be represented as:

$$S(i,j) = \begin{cases} 255 & \text{if } D(i,j) > T \\ 0 & \text{if } D(i,j) \leq T \end{cases}$$

Where:

- $S(i,j)$ is the resulting binary mask at pixel location (i,j),
- $D(i,j)$ is the difference at pixel (i,j),
- T is the threshold value (in this case, $T = 25$).

Visualization of Saliency: The binary saliency map is then colorized using the cv2.applyColorMap() function, which applies a color map to the binary mask. This step enhances the visual representation of the detected saliency, making it easier to observe regions with significant changes. In this case, the cv2.colorMap-jet color map is applied to provide a clear visual distinction between the salient areas and the background.

Combining with Original Frame: To integrate the saliency map with the original video frame, both images are combined using cv2.addWeighted(). This function blends the original frame with the colorized saliency map, giving more weight to the saliency map to ensure that the regions of motion are visually emphasized while retaining the context of the original scene. This is done using a weighted sum:

$$I_{\text{final}} = \alpha \times I_{\text{original}} + \beta \times I_{\text{saliency}}$$

Where:

- I_{final} is the final combined image,
- I_{original} is the original frame,
- I_{saliency} is the colorized saliency map,
- α and β are the weights applied to the original frame and saliency map, respectively. In this case, $\alpha = 0.2$ and $\beta = 0.9$.

III. Result and Discussion

The implemented code successfully integrates object detection, tracking, and saliency detection to analyze video content effectively. Using the **yolo11n-seg** model, the system accurately detects and tracks multiple individuals within the video, maintaining reliable performance even in challenging conditions. The saliency map highlights regions of significant activity or change in the video, drawing attention to dynamic and visually important areas. The output offers a comprehensive visual representation by integrating object detection, tracking, and saliency information, enhancing the understanding of scene dynamics.

The combination of object detection and saliency detection provides a deeper understanding of video content, with saliency highlighting dynamic activity and tracking offering insights into movement patterns and interactions. The **yolo11n-seg** model's segmentation capabilities enable accurate localization and tracking of objects, even with partial occlusion, ensuring precision in complex video scenarios. Integrating saliency into the output simplifies identifying key areas within cluttered scenes. Future improvements could include incorporating features like motion detection or object classification for more refined analysis, using advanced tracking algorithms like Kalman filtering to enhance robustness in challenging scenarios, and adding a user-friendly interface for adjustable parameters, increasing system flexibility. This integration demonstrates a promising approach to dynamic video analysis, with potential applications in fields such as surveillance and behavioral studies.

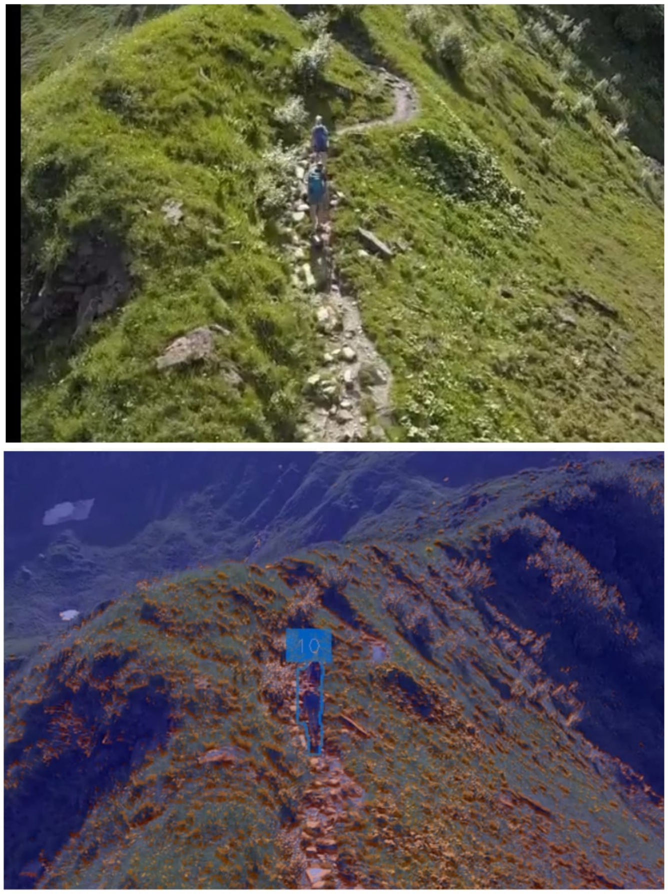

Fig. 3. Berghouse Leopard Jog Video

The top image is a screenshot from the ”Berghouse Leopard Jog” video which showcases a group of people hiking on a narrow mountain trail. The hikers are in the center of the frame, with lush green vegetation and rocky terrain surrounding them. This video is given as the input.

The bottom screenshot is from the output video generated, the segmentation highlights the hikers, trail, and surrounding vegetation. The hikers are segmented in a blue color, the trail in a yellow color, and the surrounding vegetation in various shades of orange and purple.

The top image is a screenshot from a soccer video which captures a scene from a soccer game, with players in white and black uniforms. The soccer field is green, with a white line marking the center circle. The stands are packed with spectators in the background, and the game is in progress.

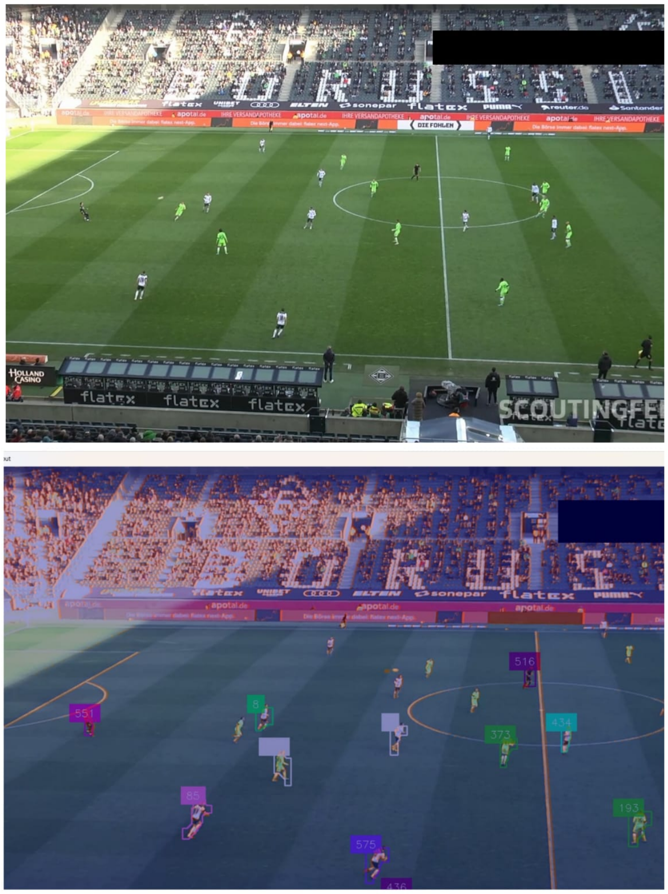

Fig. 4. soccer Video

The bottom screenshot is from the output video generated, the segmentation focuses on the players, highlighting them in various colors with identifying numbers. The soccer field is segmented in a deep green color, while the stands are segmented in a lighter shade of green. The scoreboards and other signs on the field are also segmented in different colors.

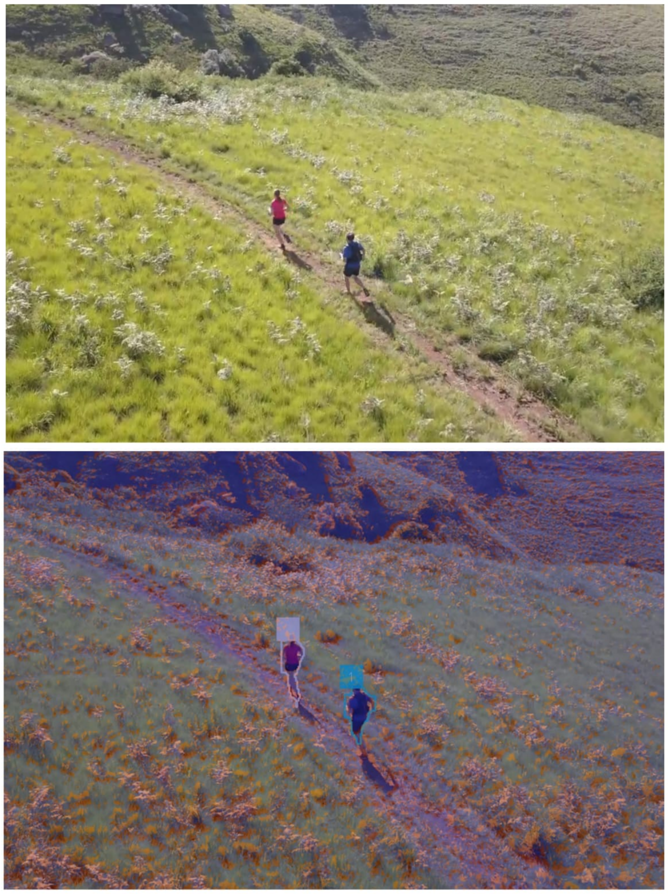

Fig. 5. Vertical Flyover video

The top image is a screenshot from the ”Vertical Flyover” video, it shows two people running on a dirt trail in a grassy field. The runners are in the middle of the frame, with a view of the surrounding green hills in the background. The sky is partially visible at the top of the image.

The bottom screenshot is from the output video generated, it highlights the runners and the trail in a bright blue color, while the grass and surrounding terrain are segmented in various shades of orange and purple.

IV. Conclusion

This project successfully demonstrated the integration of the YOLOv11n-seg model with optical flow-based saliency detection for real-time video analysis. The system effectively tracks multiple objects in the video, highlighting their movement and areas of significant change. The YOLOv11n-seg model, with its advanced segmentation features, enabled precise localization and tracking, even in complex and dynamic video environments. By combining segmentation and saliency detection, the system provides a more detailed and comprehensive understanding of the video content. This approach can be applied to various real-world scenarios, including wildlife monitoring, video surveillance, and action analysis, where real-time tracking and object detection are crucial.

Looking ahead, there are several potential directions to further improve this project. One area of focus could be the implementation of more advanced tracking algorithms, such as Kalman filtering or deep learning-based tracking, to enhance the accuracy and robustness of object tracking. Additionally, incorporating motion detection and object classification could make the system even more capable of identifying and categorizing different types of movement or behavior within a scene. Another important improvement would be the development of a user-friendly interface that allows easy adjustments to parameters like thresholds for saliency detection and object detection, making the system more accessible and adaptable to different use cases.

In conclusion, this research contributes to the growing field of intelligent video analysis by providing a more efficient and accurate method for interpreting dynamic environments. The ability to analyze and track objects in real time, combined with segmentation and saliency detection, offers great potential for a wide range of applications. With further refinement, the system could pave the way for more sophisticated and reliable video analysis solutions in fields such as security, wildlife conservation, sports analytics, and beyond.

References

[1] Anh Nguyen, Zhisheng Yan, Klara Nahrstedt *Your Attention is Unique: Detecting 360-Degree Video Saliency in Head-Mounted Display for Head Movement Prediction*MM’18, October 22-26, 2018, Seoul, Republic of Korea

[2] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi *You Only Look Once: Unified, Real-Time Object Detection* https://arxiv.org/abs/1506.02640

[3] Momina Liaqat Ali,Zhou Zhang *The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection*, 23 October 2024 doi: 10.20944/preprints202410.1785.v1

[4] Tanvir Ahmad, Yinglong Ma, Muhammad Yahya, Belal Ahmad, Shah Nazir, Amin ul Haq *Object Detection through Modified YOLO Neural Network*2020 https://doi.org/10.1155/2020/8403262