VISHNU
UNIVERSAL LEARNING

B V RAJU INSTITUTE OF TECHNOLOGY

(UGC Autonomous)
Vishnupur, Narsapur, Medak District

Department of Computer Science and Engineering

Mini Project – Final Review

Title : REAL TIME THINGS IDENTIFICATION USING YOLOv8 AND FLASK

By:

23215A0531: V.Jathin

BVRIT.
B V Raju Institute of Technology

Guided By:
D. Nagasudha, MTech(Phd)
Assistant Professor

Outline

- Abstract
- Introduction
- Literature Survey
- Existing System
- Problem Statement & Objectives
- Proposed Modules
- Proposed Architecture diagram
- Algorithm Explanations
- Data Flow diagram
- UML Diagrams
- Results & Discussion
- Conclusion & future Enhancement
- References

Abstract

This project focuses on developing a user-friendly web platform powered by the YOLOv8 model for advanced image classification, object detection, and human pose estimation. The platform allows users to upload images and videos or specify video sources such as YouTube links, RTSP streams, or webcam feeds for real-time processing. YOLOv8, a state-of-the-art algorithm known for its speed, accuracy, and efficiency, will be integrated to detect and localize objects and estimate human poses within media content.Key features include customizable detection parameters, such as confidence thresholds and object classes, with results displayed as annotated overlays on the media. Users will have options to download or share the processed outputs. The platform's intuitive interface and robust functionality make it suitable for applications in surveillance, security, robotics, augmented reality, and human-computer interaction. This project aims to provide a powerful yet accessible tool that leverages YOLOv8's capabilities to meet diverse real-world needs in computer vision.

Introduction

1. The project is focused on simplifying real-time object detection by developing an easy-to-use web application.
2. It is built using Flask, a lightweight Python web framework, and YOLOv8, a powerful deep learning model known for its speed and accuracy.
3. The application is designed to be user-friendly and suitable for detecting objects in static images, live webcam feeds, and video streams.
4. By combining Flask and YOLOv8, the system provides an efficient and accessible solution for real-time object detection tasks.

Literature Survey

Year	Author Name	Paper Title	Research Design	Conceptual / Theoretical framework	Major theme in paper	Future Idea
2024	Preethi G., Shambavi Naik,Siri M. Kenchol, Shanta P. Jakalannanavar, Rachana M. S.	Object Detection Using FasterRCNN, YOLOv7 & YOLOv8	Comparative analysis of object detection algorithms, focusing on performance metrics like accuracy, speed, and usability.	Comparative analysis of object detection algorithms, focusing on performance metrics like accuracy, speed, and usability.	Enhancing object detection efficiency and usability through advanced algorithm optimization and dataset preprocessing	Integrating real-time detection on diverse platforms, refining dataset techniques, and expanding to specific domains like medical imaging and environmental monitoring
2023	Moahaimen Talib, Ahmed H. Y. Al-Noori, Jameelah Suad	YOLOv8-CAB: Improved YOLOv8 for Real-time Object Detection	Enhances YOLOv8 using Context Attention Block (CAB) and improved spatial attention techniques	Focuses on better feature extraction for small-object detection and multi-scale utilization	Real-time and precise object detection in challenging scenarios	Explore further optimization and real-world applications of YOLOv8 enhancements

Literature Survey

Year	Author Name	Paper Title	Research Design	Conceptual / Theoretical framework	Major theme in paper	Future Idea
2023	Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye	Object Detection in 20 Years: A Survey	Comprehensive literature survey	Categorization of object detection approaches into traditional and deep learning-based methods	Trends in object detection, including real-time systems, precision-focused applications, and large-scale datasets	Exploration of zero-shot learning, domain adaptation, and integration with broader AI systems like robotics and AR
2022	Tausif Diwan, G. Anirudh, and Jitendra V. Tembhurne	Object detection using YOLO: challenges, architectural successors, datasets and applications	Comparative analysis of single-stage versus two-stage object detectors	Highlights YOLO's architecture evolution and performance metrics	Speed and efficiency of YOLO models compared to other architectures, with a focus on real-world adoption.	Suggestions for improving detection accuracy, expanding applications, and integrating with newer deep learning techniques

Literature Survey

Year	Author Name	Paper Title	Research Design	Conceptual / Theoretical framework	Major theme in paper	Future Idea
2022	Abhinandan Tripathi, Manish Kumar Gupta	Object Detection Using YOLO: A Survey	Survey of YOLO's evolution and applications	YOLO's object detection capabilities, focusing on speed, accuracy, and use in real-time applications.	The paper emphasizes YOLO's rapid advancements, its real-time detection speed, and various specialized applications	Potential improvements in detection accuracy, speed optimization, and broader applicability in diverse domains

Existing system (Max 3 Points)

Commercial solutions like Amazon Recognition, Google Cloud Video Intelligence, and Microsoft Azure Cognitive Services offer scalable, pre-trained models but can be costly and complex. Open-source frameworks such as TensorFlow, OpenCV, and Detectron2 provide flexibility but require significant expertise to implement. In contrast, the proposed Flask web app stands out for its user-friendly interface, YOLOv8 integration for speed and accuracy, and support for diverse input options like images and video streams. By offering on-premise deployment and potential model customization, it addresses user needs for accessibility, data control, and flexibility, setting it apart

from existing solutions.

Problem Statement & Objectives

Problem Statement:

1.Existing object detection and pose estimation tools are often complex and not easily accessible.

2.This project aims to build a user-friendly web app using YOLOv8 for real-time detection and pose estimation.

3.The platform will offer customizable settings and intuitive outputs for various users and use cases.

Objectives:

1.Build a user-friendly web app with an intuitive interface for uploading images, selecting video sources, and adjusting detection settings.

2.Use YOLOv8 for fast, accurate object detection across images and video frames.

3.Support real-time detection from various media sources like images, webcams, RTSP streams, and YouTube videos.

Proposed Modules

1. **User Interface Module:** This module handles the interaction between the user and the system. It provides a user-friendly interface for uploading images, videos, or specifying video sources (e.g., YouTube, RTSP, webcam)

2. **Media Processing Module:** This module is responsible for preprocessing the uploaded media (images, videos, or live streams) before passing it to the YOLOv8 model.

3. **YOLOv8 Model Integration Module:** This module integrates the YOLOv8 model into the system for object detection, image classification, and human pose estimation.

4. **Detection and Pose Estimation Module:** This module processes the output from the YOLOv8 model and overlays the detection results (bounding boxes, skeletons) on the input media.

5. **Database Module:** This module stores processed data, results, and user information for future retrieval and analysis.

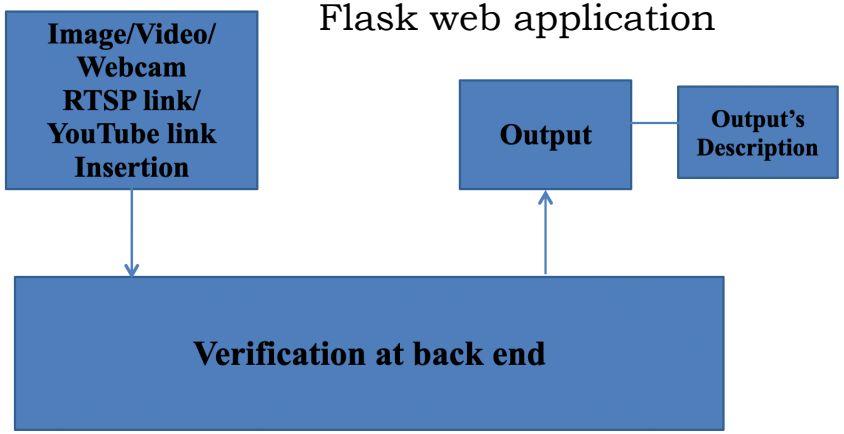
6. **Web Application Framework Module:** This module manages the overall web application using flask, including routing, media processing, and interfacing with the YOLOv8 model.

7. **Security and Privacy Module:**This module ensures the security and privacy of user-uploaded media and detection results.

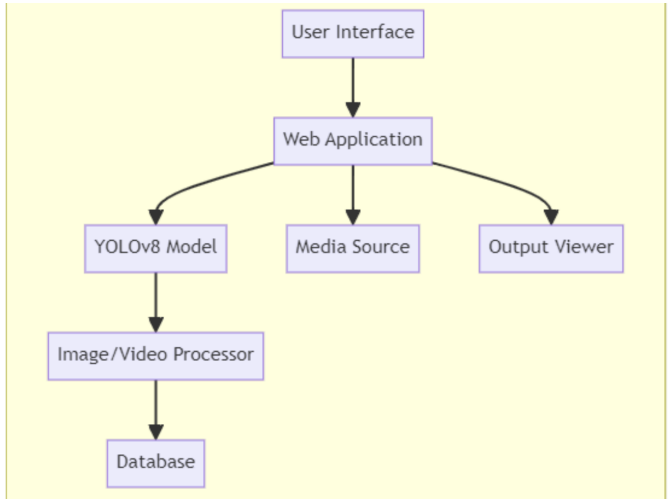
8. **Performance Optimization Module:** This module ensures the system performs efficiently, providing real-time or near-real-time processing capabilities.

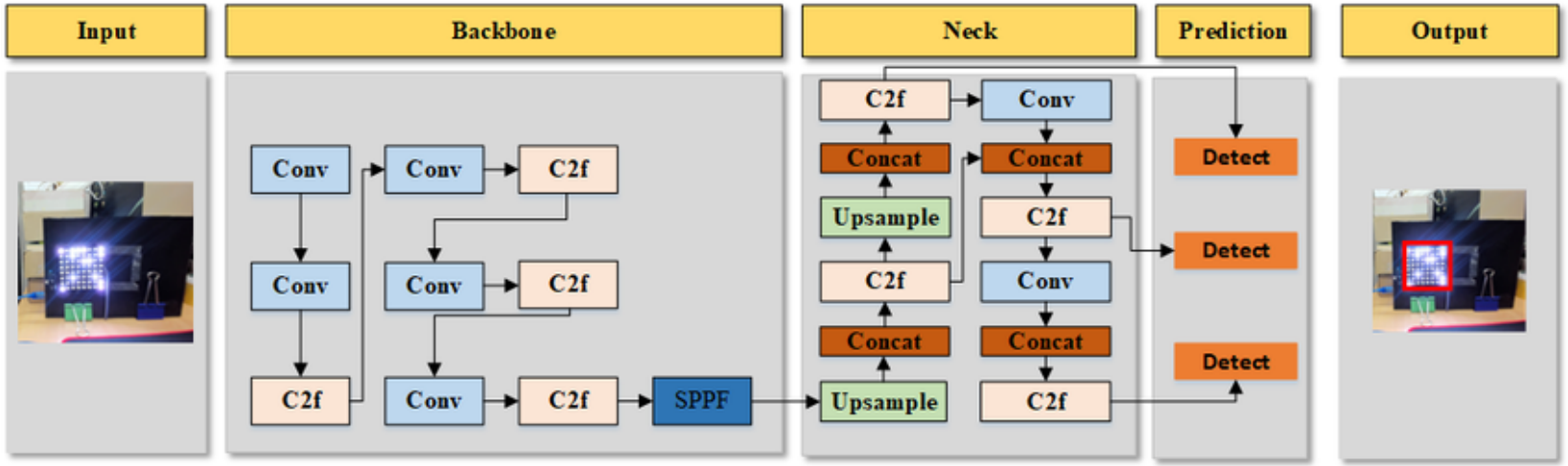
Proposed Architecture Diagram

Flask web application



Proposed system architecture





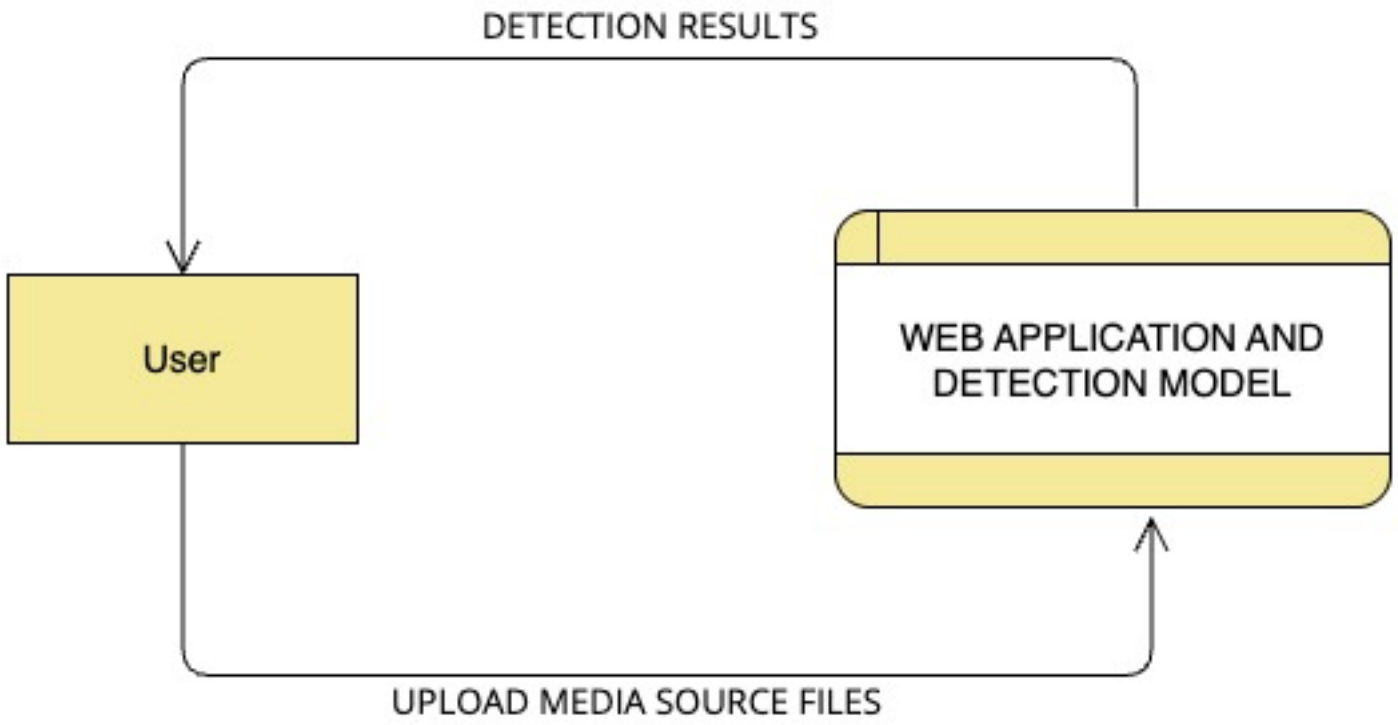
YOLO v8 architecture

Algorithm Explanation

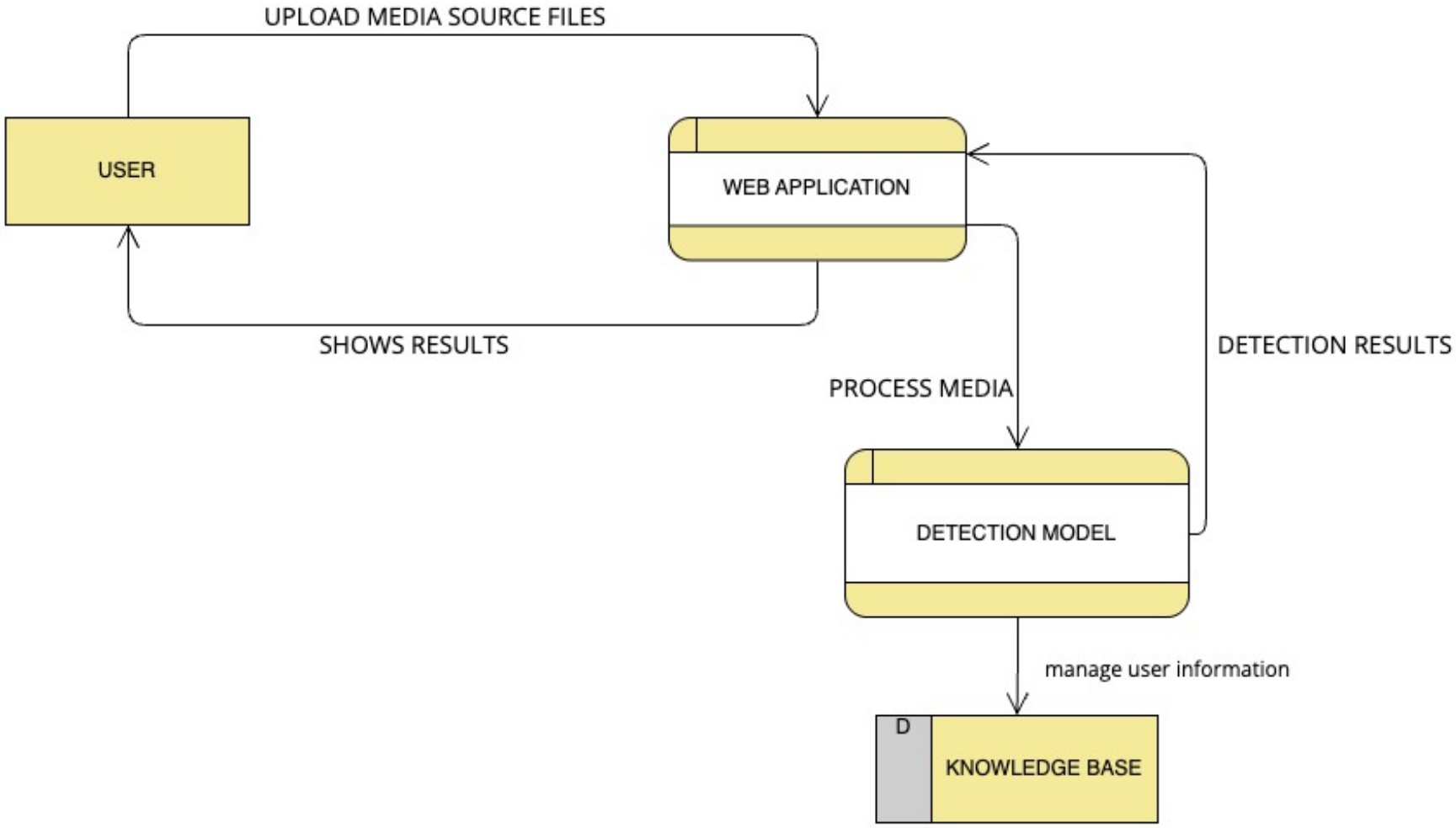
Flow of the Algorithm

1. **User Interaction**:The user uploads media(images or videos) or specifies a video source (e.g., YouTube link, RTSP stream, or webcam feed) through the web interface.
2. **Preprocessing**:Prepare the input data for the YOLOv8 model by performing necessary preprocessing steps (resizing, normalization, etc.).
3. **Model Inference**:The preprocessed media is passed through the YOLOv8 model for detection, classification, or pose estimation.
4. **Post-Processing**:Refine the model's output to improve accuracy and usability.The model's output is refined using NMS and threshold filtering**.**
5. **Visualization**:The results are overlaid on the input media and displayed to the user web interface.
6. **Output**:The user can download or share the processed results.Enables users to retain and share detection results.Supports multiple output formats for flexibility**.**

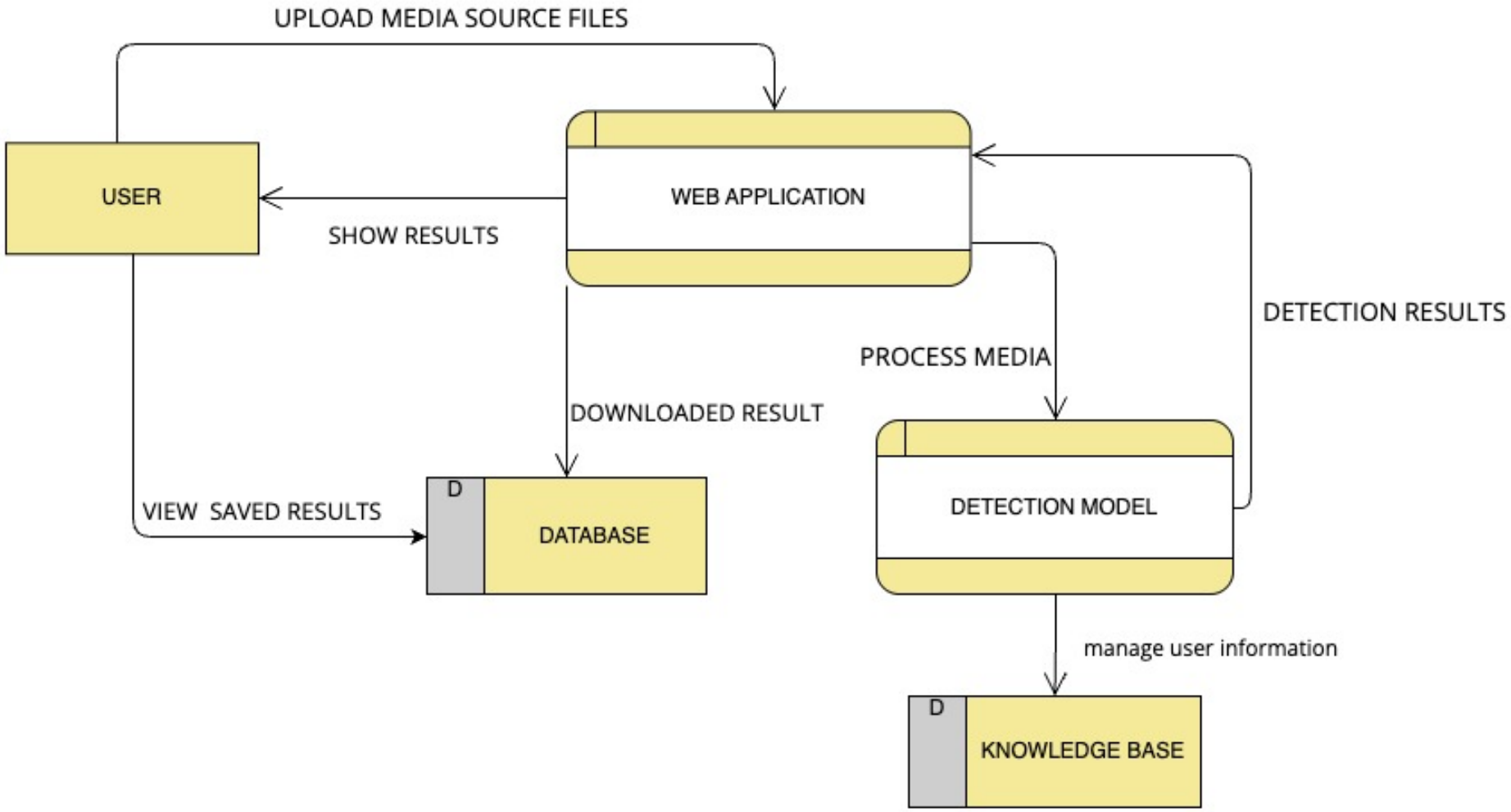
Data Flow diagram (Level-0)



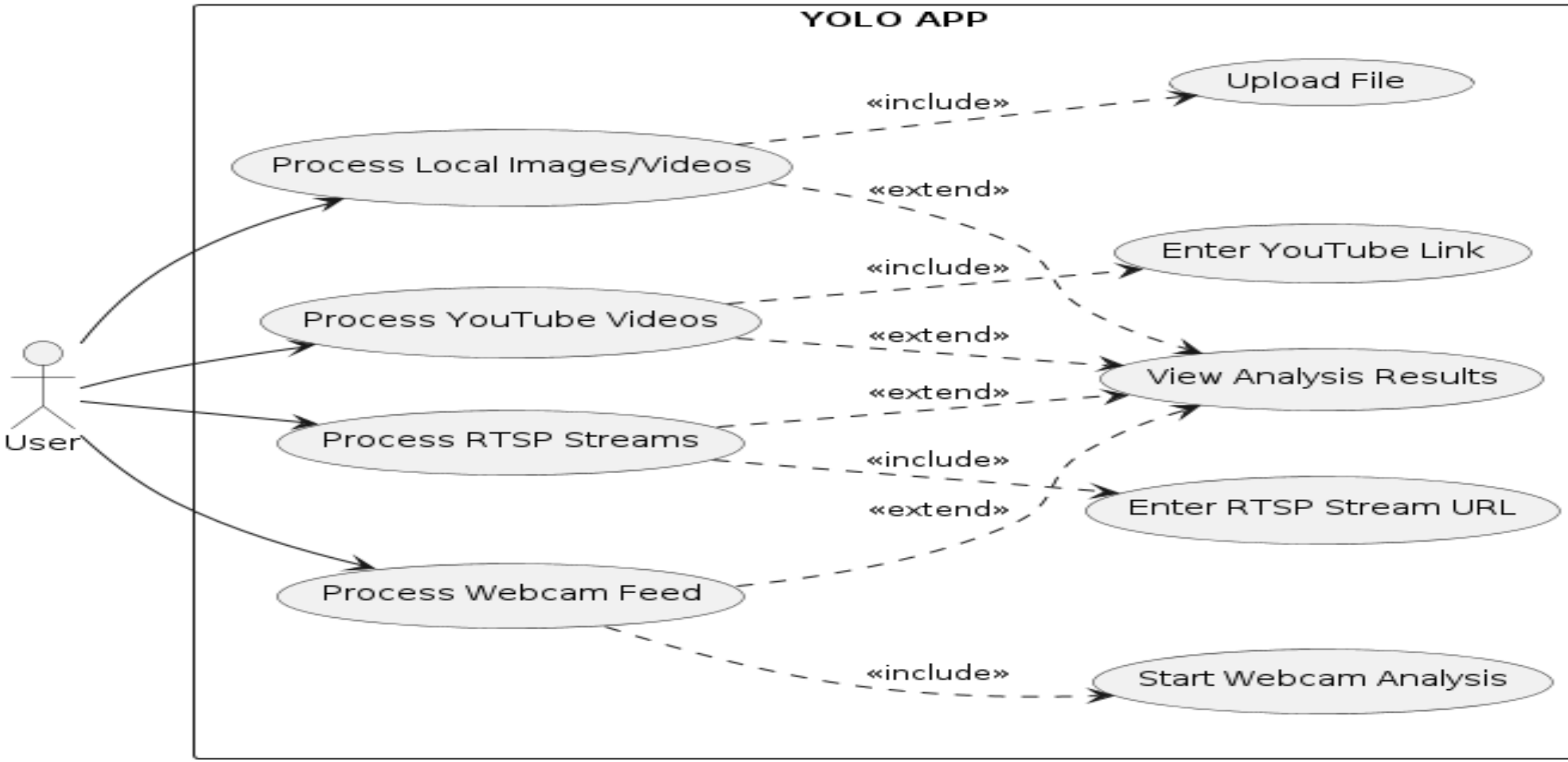
Data Flow diagram (Level-1)



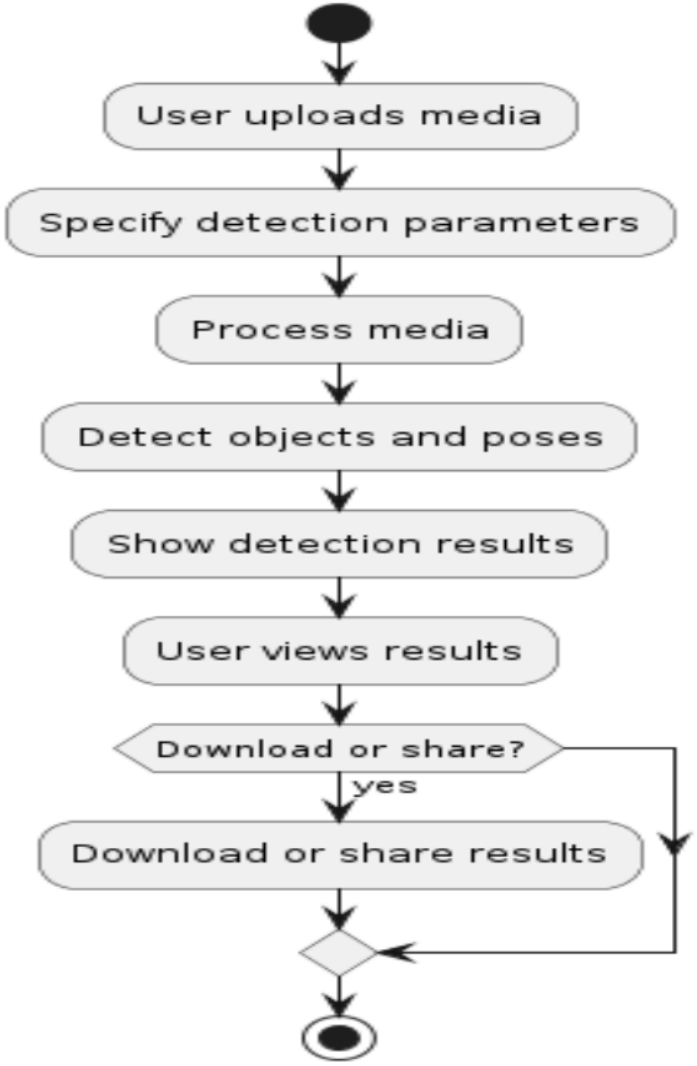
Data Flow diagram (Level-2)



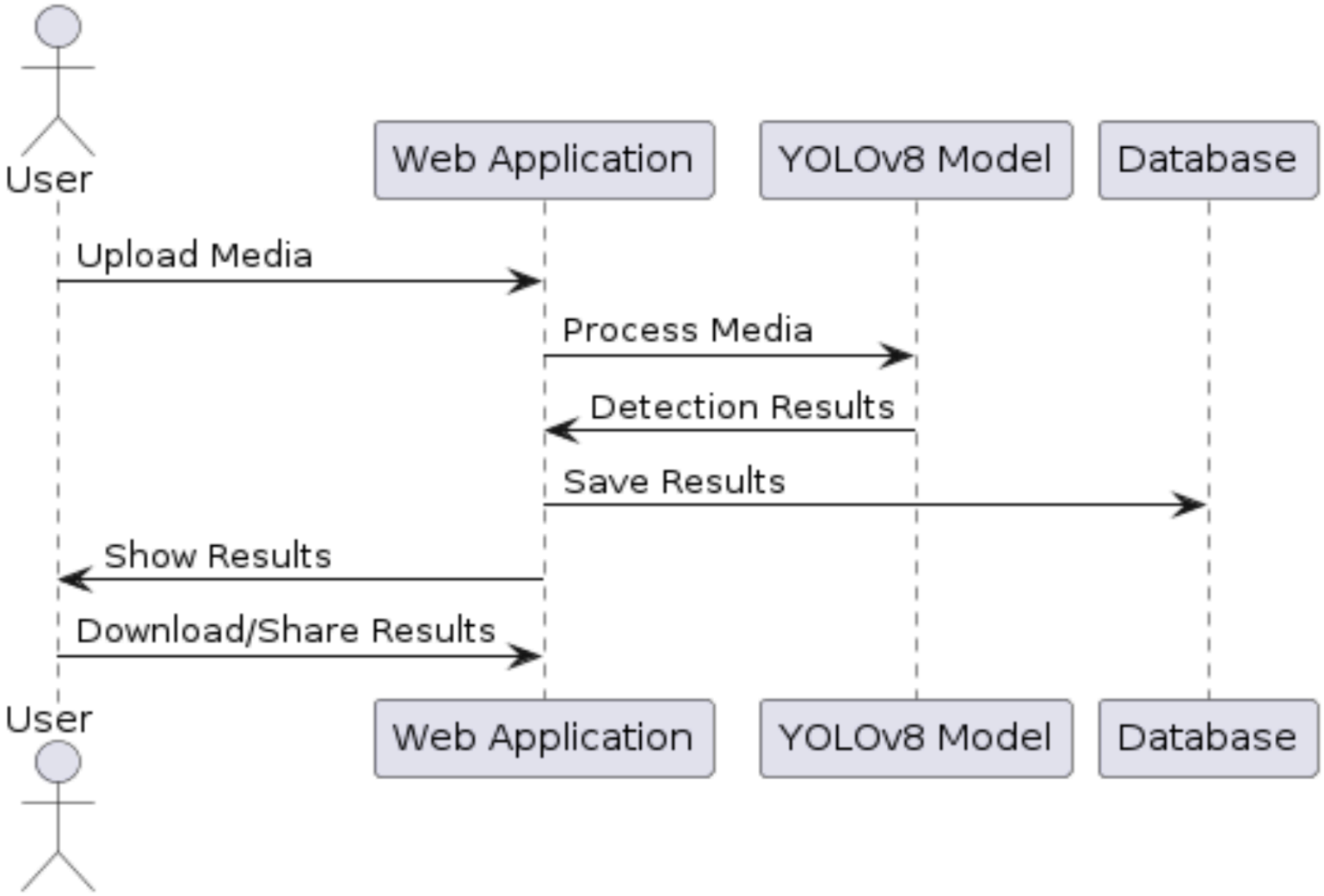
UML Diagrams – Use case



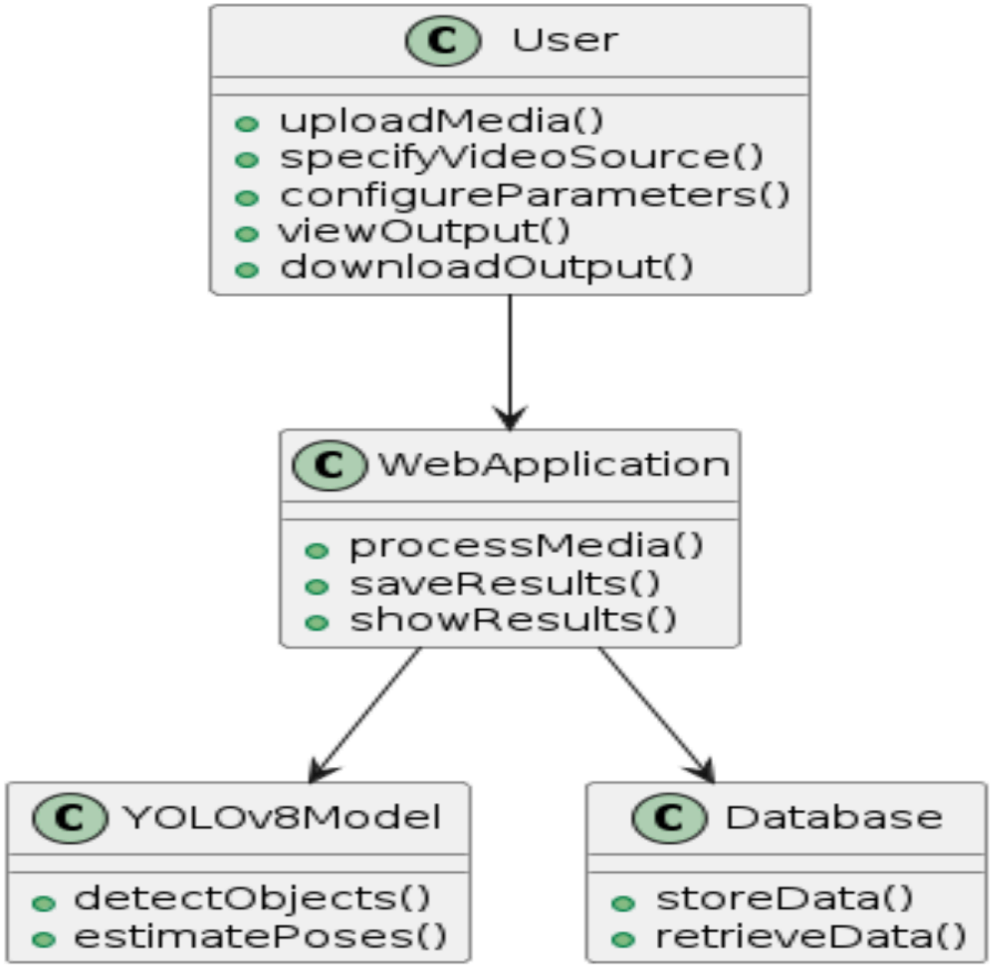
UML Diagrams – Activity Diagram



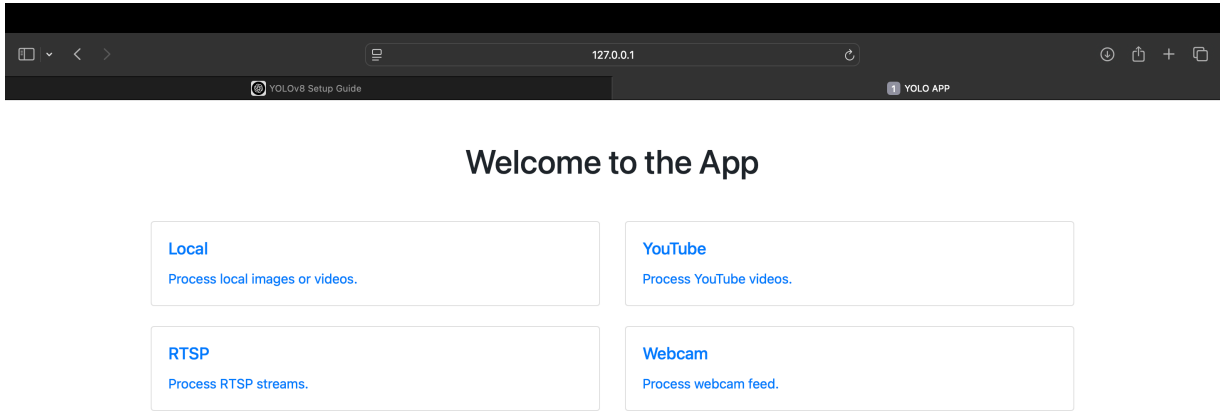
UML Diagrams – Sequence Diagram



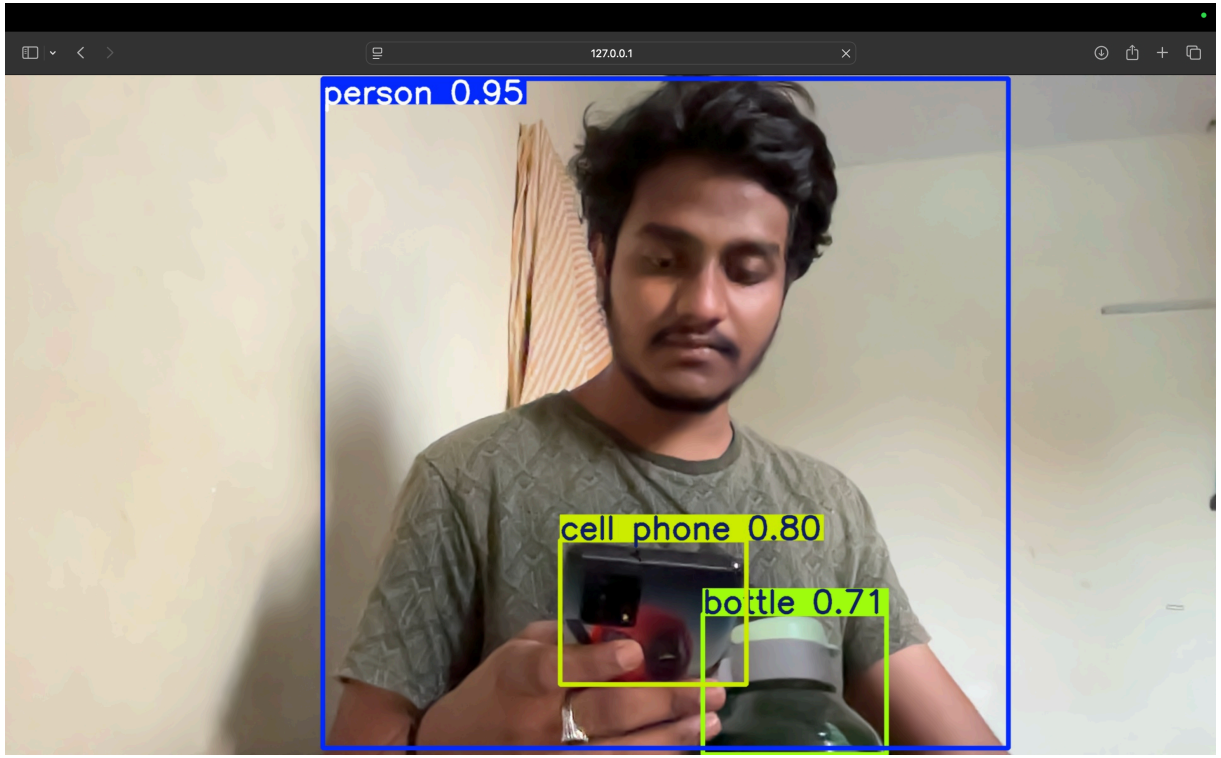
UML Diagrams – Class Diagram



Results and Discussions(Implementation screenshots)

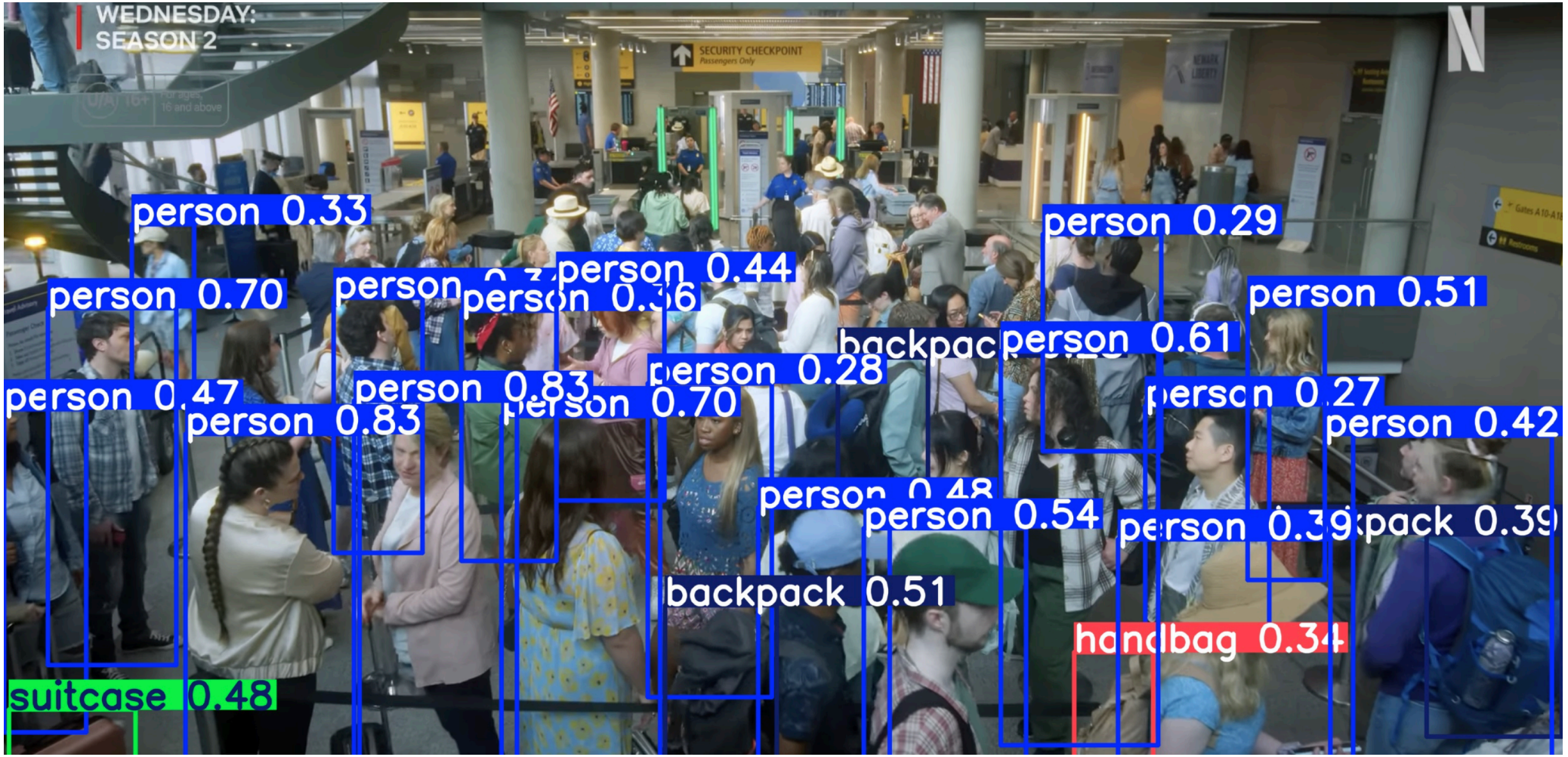


Flask web framework Interface



Live Web Cam Analysis

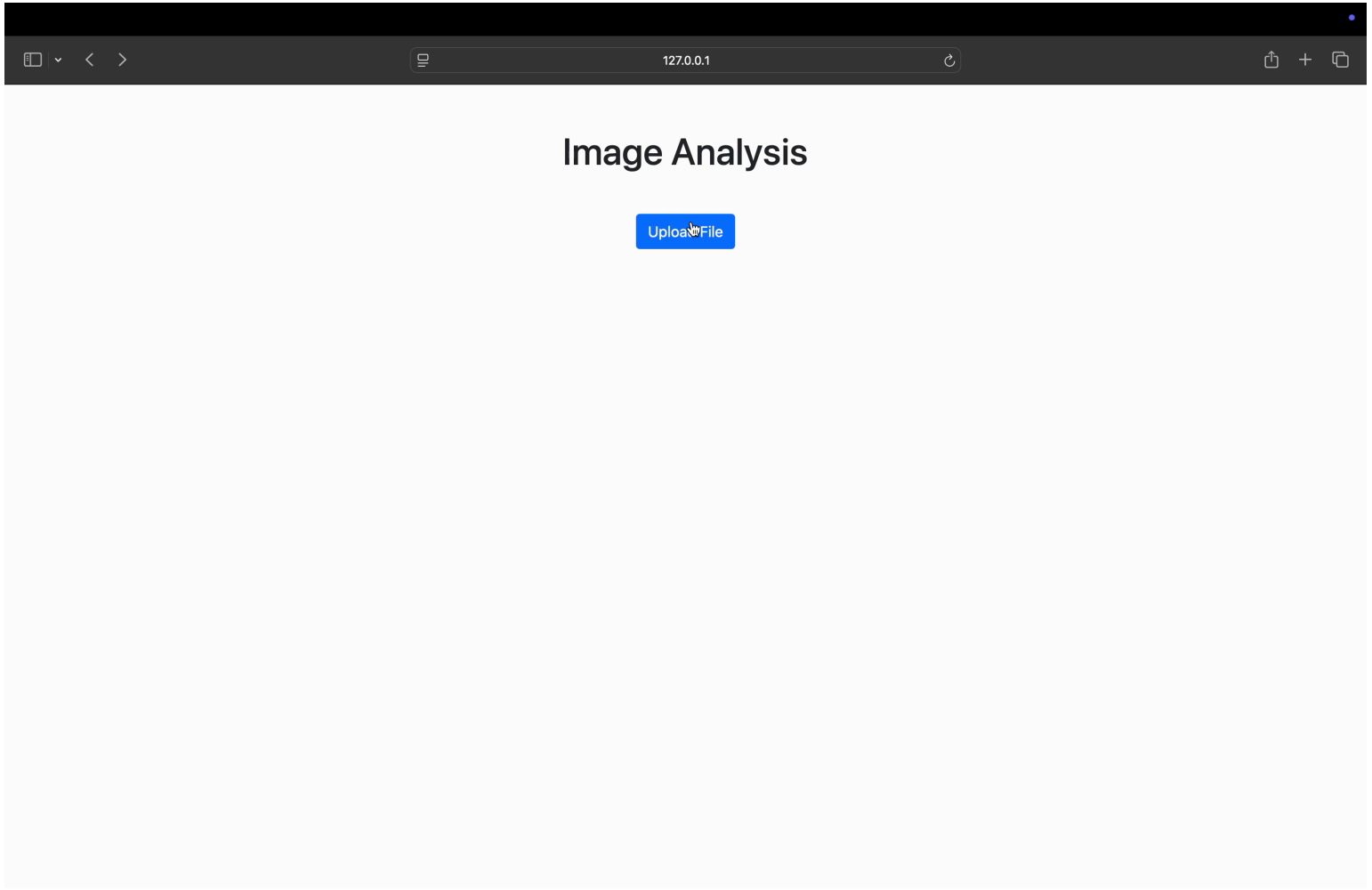
Implementation screenshots



Object detection for static images

Implementation screenshots

You tube links and Video Analysis



Results and Discussions(Expected results comparison)

Metric	Proposed System (YOLOv8)	YOLOv7	Faster R-CNN	OpenPose
Accuracy (mAP)	56.8% (COCO dataset)	55.2%	53.5%	N/A
Speed (FPS)	83 FPS	67 FPS	7 FPS	22 FPS
Resource Usage	4.5 GB GPU memory	5.2 GB	8.1 GB	6.8 GB
Scalability	50+ concurrent users	30+ users	10+ users	15+ users
Pose Estimation (AP)	72.5% AP	N/A	N/A	74.3% AP
Training Time	12 hours	14 hours	20 hours	18 hours
Input Source Support	5 input sources	3 sources	2 sources	2 sources
Output Options	4 output formats	2 formats	2 formats	2 formats

Conclusion and Future Enhancement

Conclusion:

1. The project effectively utilizes YOLOv8 for object detection, image classification, and human pose estimation with high accuracy and efficiency.
2. It proves suitable for real-world applications like security, autonomous driving, and human-computer interaction due to its robustness and versatility.

Future Enhancement:

1. Future improvements include adding real-time processing and expanding the training dataset for better accuracy and adaptability.
2. Additional features like multi-camera support, advanced tracking, and detection of complex object interactions will enhance the system's capabilities.

References

[1] P. G., S. Naik, S. M. Kenchol, S. P. Jakalannanavar, and R. M. S., "Object Detection Using FasterRCNN, YOLOv7 & YOLOv8," Indiana Journal of Multidisciplinary Research, vol. 4, no. 3, pp. 136-141, 2024. doi: 10.5281/zenodo.12674762

[2] M. Talib, A. H. Y. Al-Noori, and J. Suad, "YOLOv8-CAB: Improved YOLOv8 for Real-time Object Detection," Karbala International Journal of Modern Science, vol. 10, no. 3, pp. 34-45, 2024. doi: 10.33640/2405-609X.3339.

[3] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object Detection in 20 Years: A Survey," Proceedings of the IEEE, vol. 111, no. 3, pp. 257-276, Mar. 2023, doi: 10.1109/JPROC.2023.3238524.

[4] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," Multimedia Tools and Applications, vol. 82, no. 13, pp. 14823–14855, 2022. doi: 10.1007/s11042-022-13644-y

[5] T. Ahmad, Y. Ma, M. Yahya, B. Ahmad, S. Nazir, and A. ul Haq, "Object Detection through Modified YOLO Neural Network," Scientific Programming, vol. 2020, Article ID 8403262, 9 pages, 2020. doi: 10.1155/2020/8403262.

[6] Abhinandan Tripathi,Manish Kumar Gupta, "Object Detection using YOLO: A Survey," in Proceedings of the 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), 2022, pp. xx-xx. doi: 10.1109/IC3I56241.2022.10073281.

[7] Abhishek Sarda; Shubhra Dixit,"Object Detection for Autonomous Driving using YOLO algorithm," in Proc. IEEE Int. Conf. on Intelligent Engineering and Management (ICIEM), 2021.doi: 10.1109/ICIEM51511.2021.9445365

[8] S. Li, Y. Li, Y. Li, M. Li, and X. Xu, "YOLO-FIRI: Improved YOLOv5 for Infrared Image Object Detection," IEEE Access, vol. 9, pp. 141861-141875, 2021. doi: 10.1109/ACCESS.2021.3120870.

[9] Fang, W., Li, Y., Wang, Z., & Xu, X. (2019). Tinier-YOLO: A Real-Time Object Detection Method for Constrained Environments. IEEE Access, 7, 158719–158728. https://doi.org/10.1109/ACCESS.2019.2961959

[10] L. Chao, W. Jiaan, and D. Tianyuan, "Object Detection Based on YOLO Network," Proc. IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC), 2018, pp. 157-162. doi: 10.1109/ITOEC.2018.8740604.

Thank You .....