VISHNU
UNIVERSAL LEARNING

B V RAJU INSTITUTE OF TECHNOLOGY

(UGC Autonomous)
Vishnupur, Narsapur, Medak District

Department of Computer Science and Engineering

Mini Project – First Review (2022 Batch)

Title : REAL TIME THINGS IDENTIFICATION USING YOLOv8 AND FLASK

By:

23215A0531: V.Jathin

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

BVRIT.
B V Raju Institute of Technology

Outline

- Overview of 0th Review
- Recommendations & Suggestions by panel member in 0th Review.
- Proposed System Architecture diagram
- Proposed Flow diagram (Flow Chart, which you kept in the proposal document)
- Proposed Data Flow diagram
- Proposed List of Modules
- Proposed Algorithm Working Steps.
- UML Diagrams (Use Case, Class Diagram, Sequence Diagram, Activity Diagram)
- Expected Results in the Proposed Model.

Overview of 0th Review

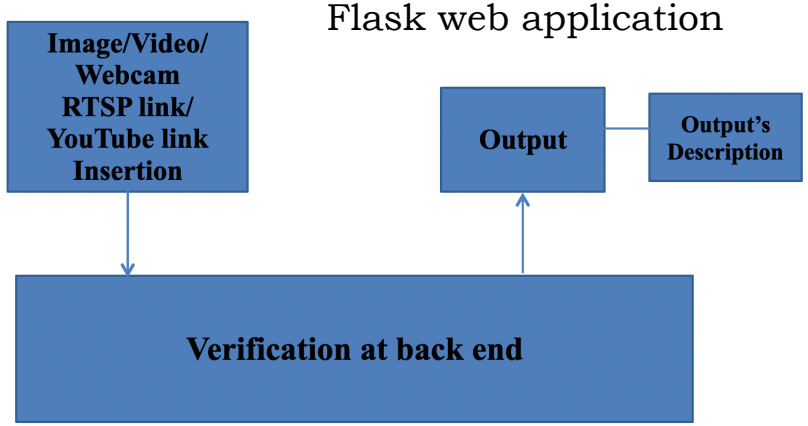
- **Short points about topics presented in 0th review.**

1. Project Overview – A web-based real-time object detection platform using YOLOv8 and Flask, supporting images, videos, and live streams with user-friendly customization options.
2. Problem Statement – Existing object detection systems are either too complex or expensive; this project aims to provide an accessible, real-time, and customizable solution.
3. Objectives – Develop an intuitive web app with real-time object detection and pose estimation, allowing users to adjust parameters and process multiple media formats.
4. Proposed Model – Features real-time processing, bounding box overlays, customization settings, and a simple UI for uploading media and tuning detection accuracy.
5. Applications – Can be used in security, robotics, surveillance, and augmented reality, providing an efficient and scalable solution for various real-world detection tasks.

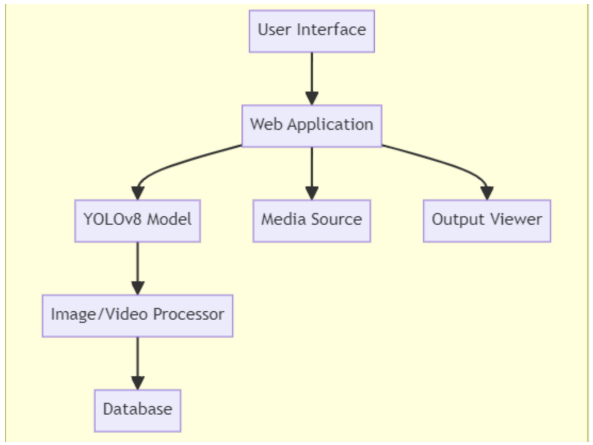
S.NO	Recommendations & Suggestions given by panel	Changes made by member as per suggestions
1	Study base paper thoroughly	selected and Studied the recent published base paper
2	Learn models related to your project	Researched about how yolov8 object detection is possible
3	Comparison about the present proposed model and existing models	YOLOv8 is better than YOLOv7 because it offers improved accuracy across various model sizes, faster detection speeds, a more flexible architecture

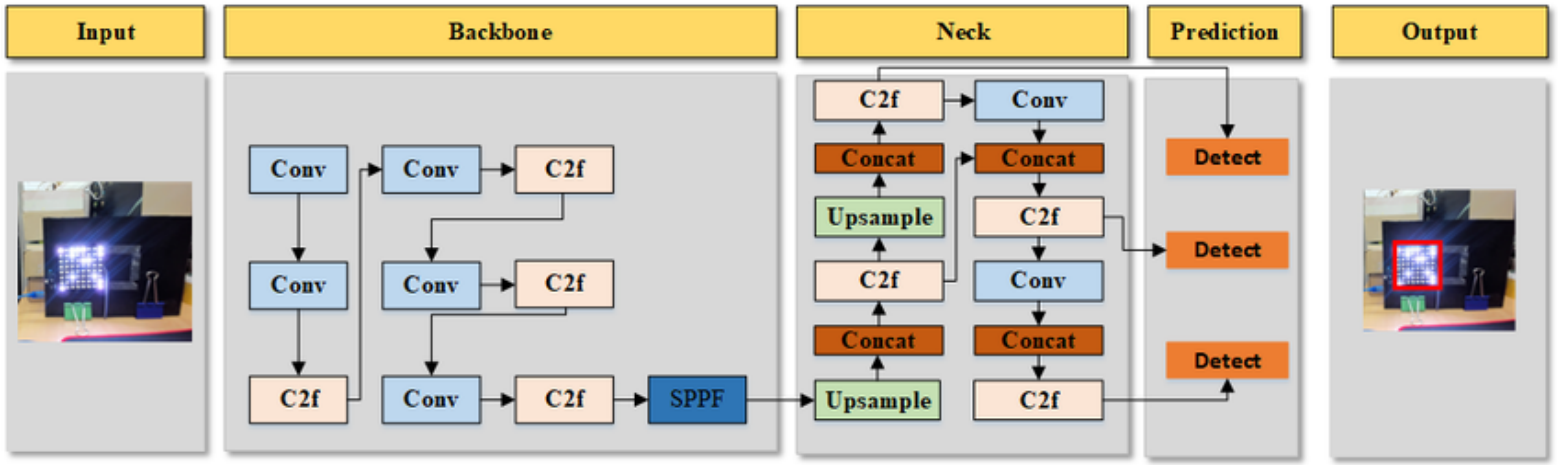
Proposed Architecture diagram

Flask web application



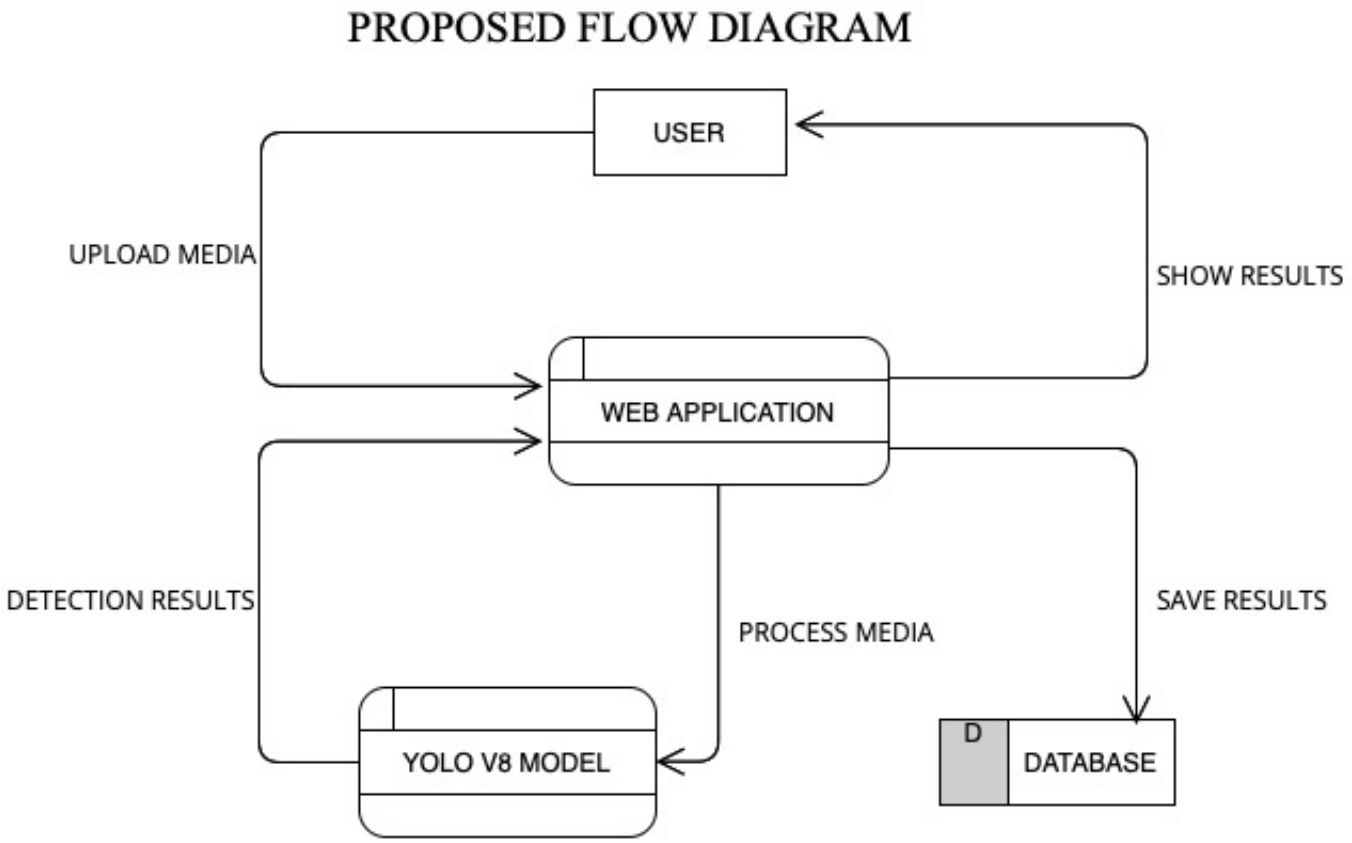
Proposed system architecture



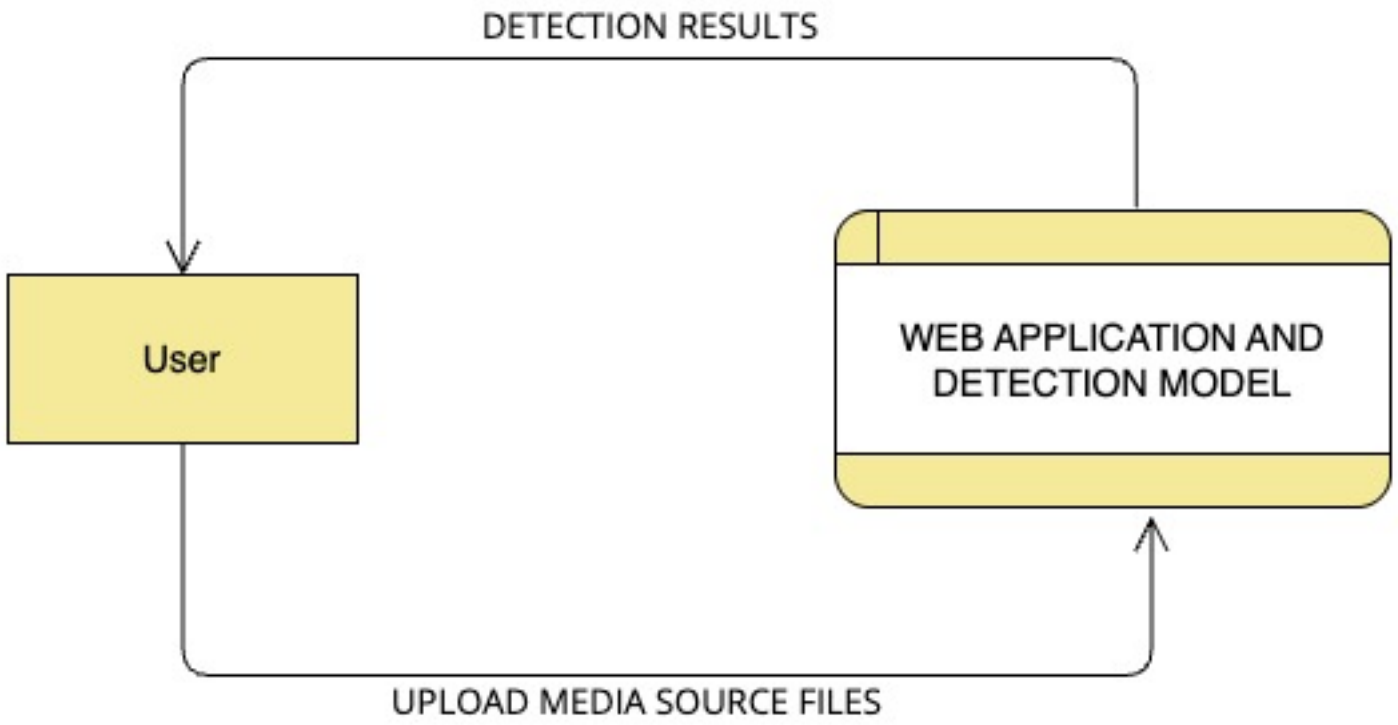


YOLO v8 architecture

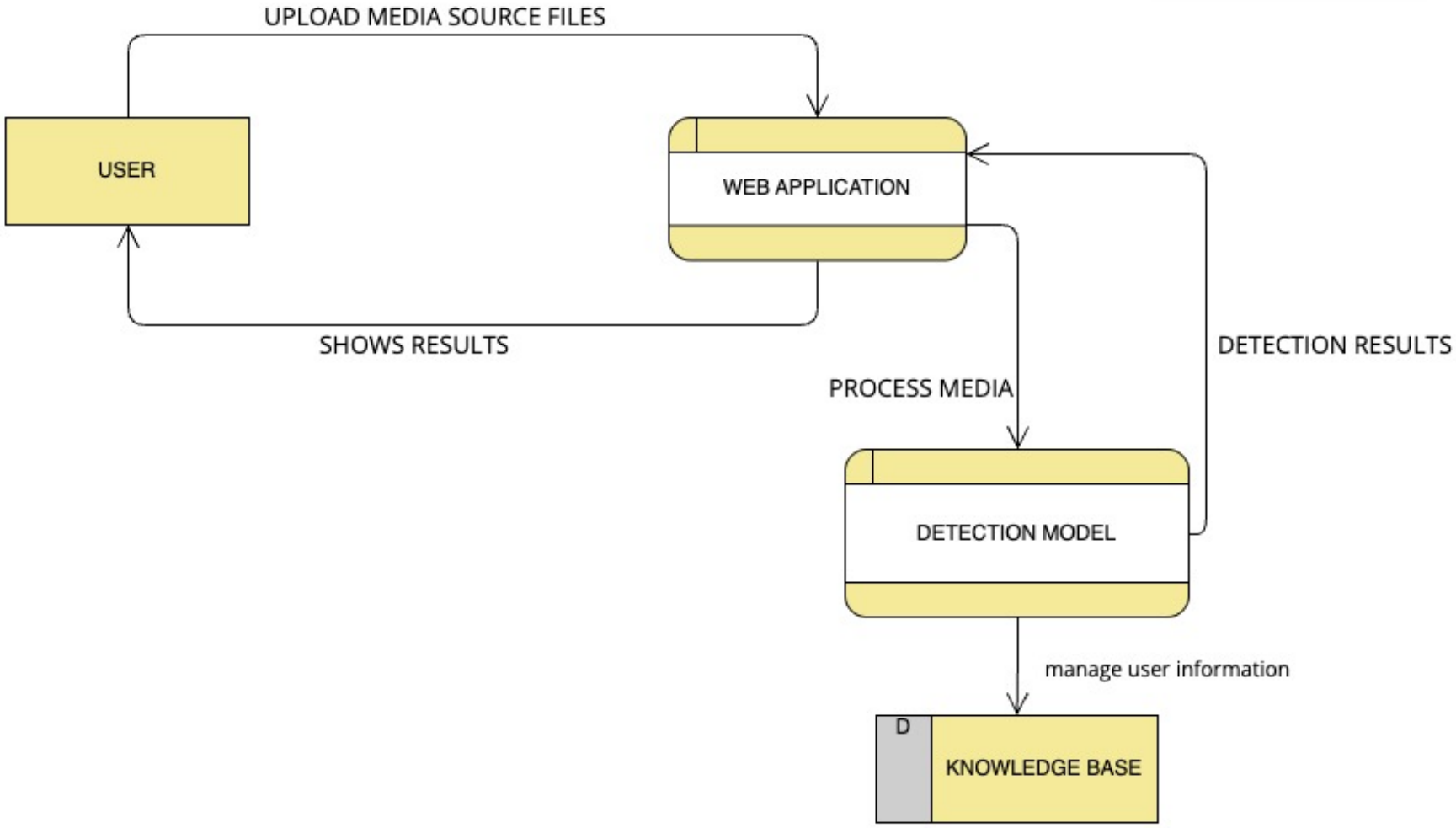
Proposed Flow diagram



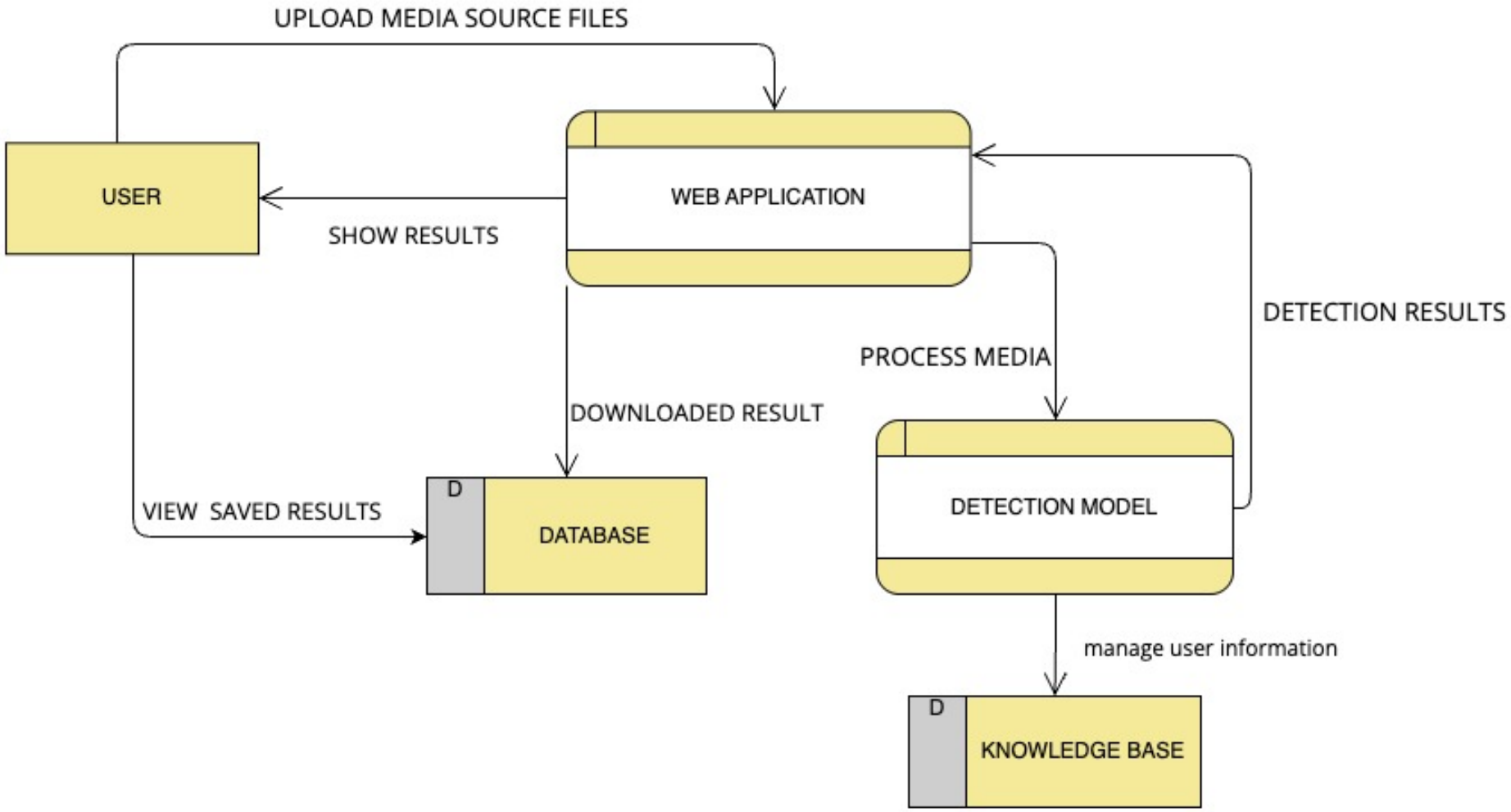
Data Flow Diagram(Level 0)



Data Flow Diagram(Level-1)



Data Flow Diagram(Level-2)



Proposed List of 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.

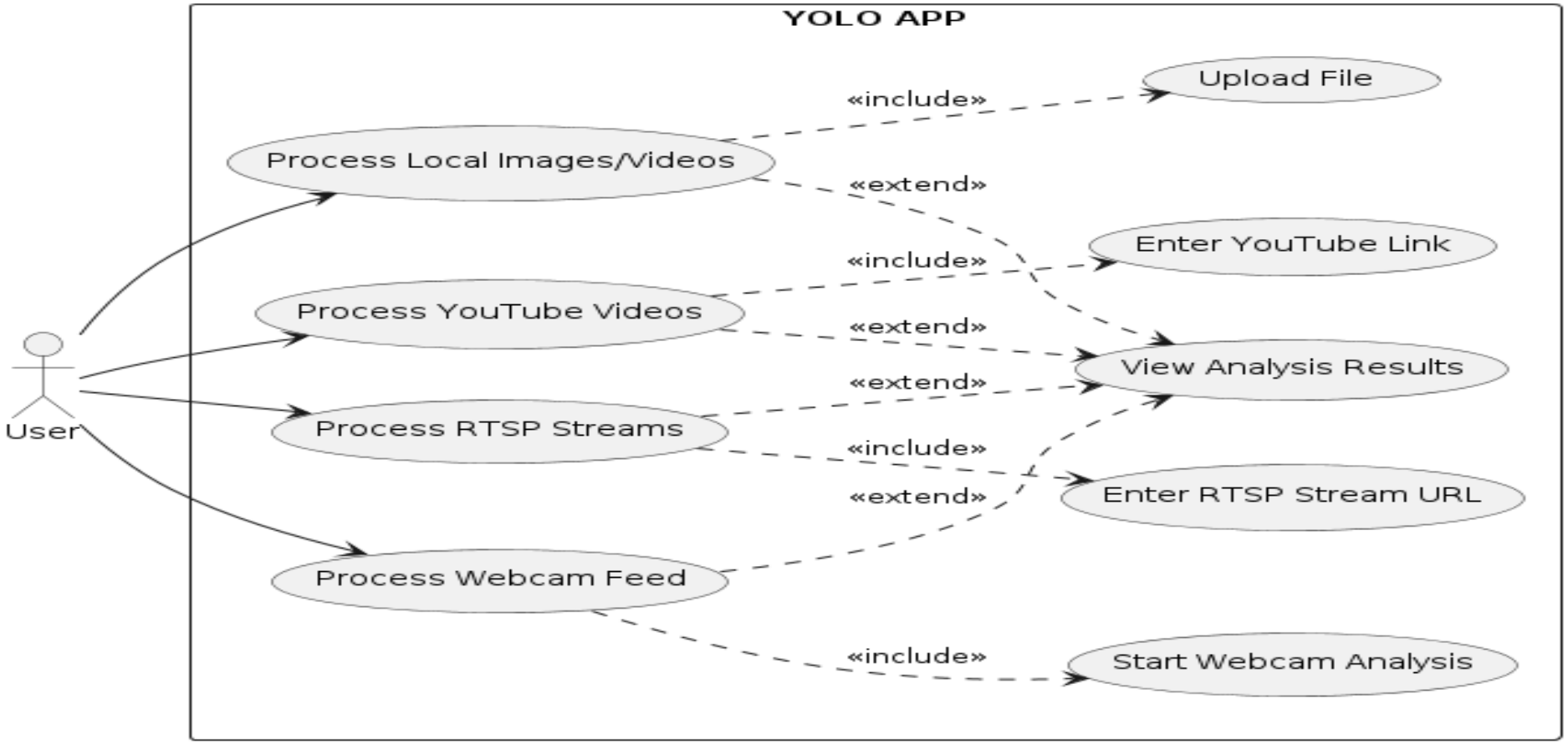
8. **Performance Optimization Module:** This module ensures the system performs efficiently, providing real-time or near-real-time processing capabilities.

Proposed Algorithm Explanation

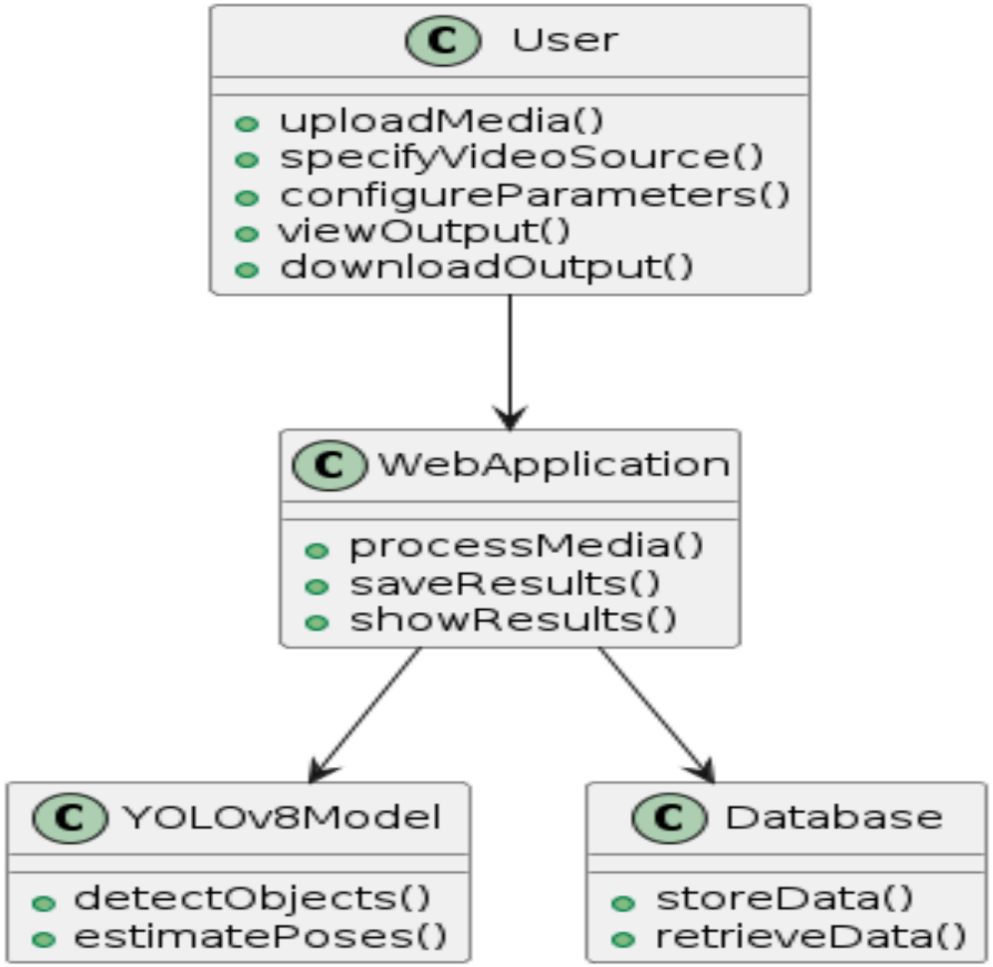
Flow of the Algorithm

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**.**

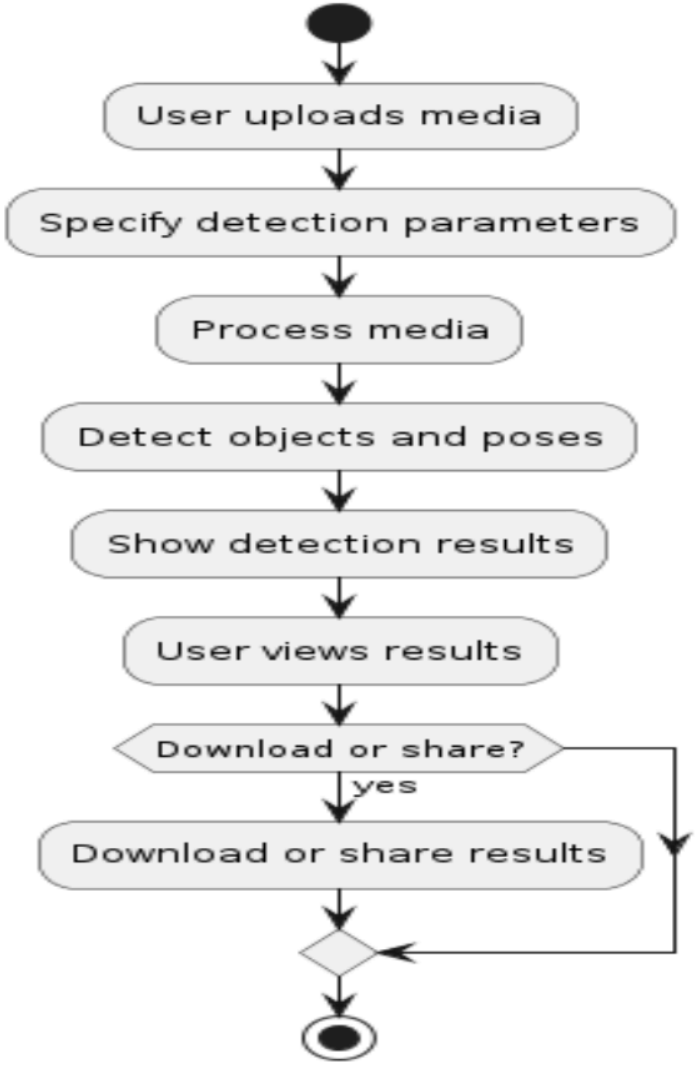
UML Diagram:-Use Case Diagram



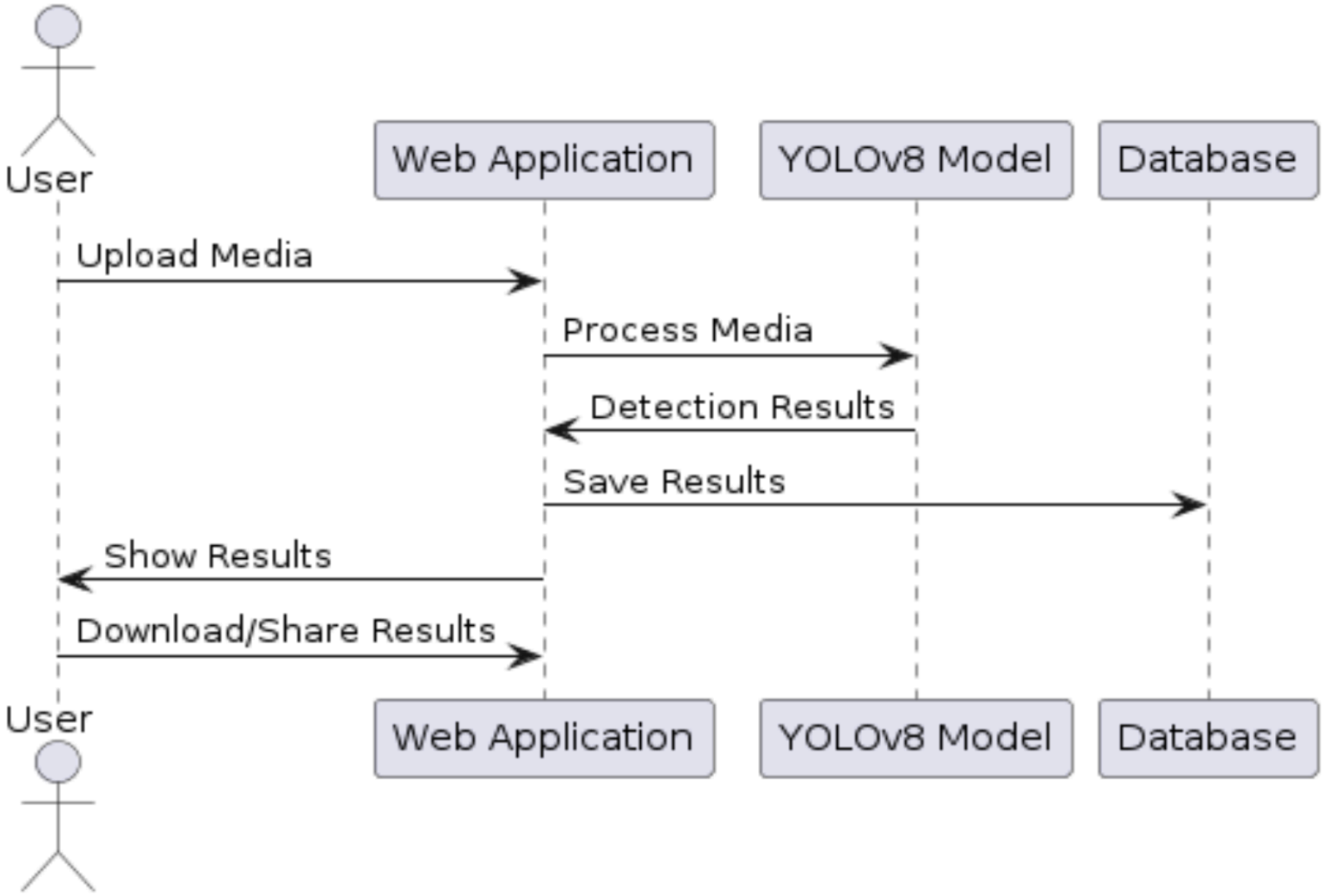
UML Diagram:-Class Diagram



UML Diagram:-Activity Diagram



UML Diagram:-Sequence Diagram



Expected Results in Proposed model

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

References (Min 10 Reference in chronological order)

[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 .....