Automated Identification of Student Royal Court Member using YOLOv8 and ChatGPT

Group Members: Allan Muir, Uzma Amir, Oshin Wilson, Tanjee Afreen, Patrick Adegbaye, Justin An, Onyinye Obioha Val

Professor: Dr. Yu

Date: December 13, 2024

Title: **Automated Identification of Student Royal Court Member using YOLOv8 and ChatGPT**

Abstract

This report outlines a system for the automated identification of UDC (University of the District of Columbia) student members of a royal court using YOLOv8 for object detection and ChatGPT for contextual reasoning. The system leverages YOLOv8's real-time detection capabilities to recognize royal court-specific attire, accessories, and context (e.g., ballons, backpacks, crowns, robes, scepters). At the same time, ChatGPT interprets this data to confirm membership status and determine if the individuals are students. This project finds applications in educational institutions, ceremonial events, and automated attendee analysis systems.

Introduction

The YOLO (You Only Look Once) series of models has significantly shaped the field of object detection by offering exceptional speed and accuracy in real-time applications. As the most recent iteration in this lineage, YOLOv8 builds upon its predecessors with enhanced capabilities, such as improved architecture, flexible training processes, and robust performance. This essay delves into a Jupyter Notebook implementation of YOLOv8, examining its workflow, results, and the broader implications of using this advanced model for object detection tasks [1] .

Background/Related Work/Key Problem/Motivation

Object detection is a fundamental task in computer vision, requiring the identification and localization of objects within an image. Traditional methods, such as the Viola-Jones framework, relied on handcrafted features and sliding window techniques, but these approaches were constrained by their reliance on fixed feature extraction methods and high computational demands. The advent of deep learning transformed object detection, introducing Convolutional Neural Networks (CNNs) capable of end-to-end feature extraction and classification. Early models, such as R-CNN (Region-based Convolutional Neural Network), pioneered the use of region proposals and classification but suffered from computational inefficiencies due to their multi-stage pipelines [1] .

YOLO (You Only Look Once) revolutionized object detection by reformulating it as a single regression problem, predicting bounding boxes and class probabilities in one pass. Unlike region-based methods, YOLO divided the image into a grid and performed detections directly, achieving real-time performance. Subsequent versions, such as YOLOv3 and YOLOv4, introduced enhancements like multi-scale predictions and improved backbone architectures, achieving a better balance between speed and accuracy. YOLOv8 represents the latest

advancement in this lineage, offering an optimized architecture that incorporates dynamic anchors, enhanced training procedures, and refined post-processing techniques [2]. These innovations make YOLOv8 a robust choice for real-time object detection across a variety of applications.

YOLOv8 features a cutting-edge single-shot object detection framework that leverages various mathematical principles and functions. This includes a sophisticated loss function that integrates bounding box, classification, and distribution focal losses, all optimized through stochastic gradient descent with momentum and weight decay. Additionally, evaluation metrics, particularly the mean average precision (mAP), are employed to address challenges such as small object detection, occlusion handling, and class distinction. These mathematical functions are integral to the development and application of YOLOv8.

$$L(\theta) = \frac{\lambda_{box}}{N_{pos}} L_{box}(\theta) + \frac{\lambda_{cls}}{N_{pos}} L_{cls}(\theta) + \frac{\lambda_{dfl}}{N_{pos}} L_{dfl}(\theta) + \varphi \parallel \theta \parallel_2^2$$

Eq1. Generalized Loss Function

$$V^t = \beta V^{t-1} + \nabla_\theta L(\theta^{t-1})$$

$$\theta^t = \theta^{t-1} - \eta V^t$$

Eq2. Velocity and Weight Updates

$$L = \frac{\lambda_{box}}{N_{pos}} \sum_{x,y} 1_{e_x^{x,y}} \left[1 - q_{x,y} + \frac{\parallel b_{x,y} - \hat{b}_{x,y} \parallel_2^2}{\rho^2} + \alpha_{x,y} v_{x,y} \right] + \frac{\lambda_{cls}}{N_{pos}} \sum_{x,y} \sum_{c \in classes} [y_c \log \hat{y}_c + (1 - y_c) \log(1 - \hat{y}_c)] + \frac{\lambda_{dfl}}{N_{pos}} \sum_{x,y} 1_{e_z^{x,y}} [-(q_+^{(x,y)} - q_{x,y}) \log_q(x,y) + (q_{x,y} - q_-^{(x,y)}) \log q_-^{(x,y)}]$$

Eq3. YOLOv8 Loss Function

Where:

- $q_{x,y}$: **intersection over Union (IoU)**

- $v_{x,y}$: **Aspect ratio loss**
- ρ : **length of the smallest enclosing box**
- $1_{e_2^x, y}$ **Indicator for cells containing objects**

Despite advancements in object detection models, several challenges persist. A critical issue is the trade-off between speed and accuracy. High-accuracy models, such as R-CNN derivatives, demand extensive computational resources, making them impractical for real-time or edge-device applications. Conversely, faster models often compromise on precision, particularly when detecting small or overlapping objects. Another challenge lies in generalizing across domains. Models trained on specific datasets, like COCO, frequently underperform in tasks involving different environments, lighting conditions, or object types. Moreover, detecting objects in complex scenes with clutter, occlusion, or unusual orientations remains a significant hurdle. Resource constraints further complicate the deployment of object detection models on edge devices, which typically lack the computational power to run sophisticated architectures efficiently.

The motivation for employing YOLOv8 stems from its ability to address these enduring challenges effectively [6]. As a real-time detection model, YOLOv8 is designed for applications requiring low latency, such as autonomous vehicles, surveillance systems, and robotics. Its improved architecture and dynamic computation methods achieve high accuracy even in demanding scenarios, such as detecting small or occluded objects. Additionally, YOLOv8's versatility allows it to be fine-tuned for specific tasks, making it suitable for both general-purpose and specialized use cases. Its lightweight design ensures compatibility with resource-constrained environments, enabling deployment on devices like drones, smartphones, and IoT systems.

In an era where industries increasingly depend on computer vision for automation, safety, and efficiency, the demand for robust, real-time object detection solutions has grown. YOLOv8 addresses this demand by offering a model that balances speed, accuracy, and resource efficiency. By leveraging YOLOv8's strengths, this work seeks to demonstrate its practical applicability in addressing real-world challenges, paving the way for advancements in diverse fields such as healthcare, logistics, public safety, and entertainment.

Work Done/Approach

The notebook provides a systematic implementation of the YOLOv8 pipeline, showcasing the full journey from model configuration to real-world deployment. It begins by setting up the YOLOv8 model through a Python package, likely ultralytics, which simplifies interaction with

the model. The configuration process allows customization of parameters like input resolution, learning rates, and batch sizes to tailor the model to specific applications.

Following the setup, the dataset is prepared for training. This involves ensuring that images and annotations align with YOLO's input requirements, such as defining bounding boxes and class labels. Data augmentation techniques, including flipping, scaling, and color jittering, are employed to increase the diversity of the training set and enhance the model's robustness.

Training the YOLOv8 model marks a critical stage in the workflow. During this phase, the model is exposed to thousands of annotated images, learning to identify patterns that define various object categories. Performance metrics, including precision, recall, and mean Average Precision (mAP), are logged to evaluate the model's progress. These metrics offer a quantitative understanding of how well the model distinguishes between objects and minimizes false detections.

After training, the model undergoes evaluation on a dedicated test dataset. The results reveal the strengths and areas for improvement in the trained model. Visualizations of the model's predictions on test images are included in the Notebook, demonstrating its ability to generate accurate bounding boxes and class labels.

Finally, the model is deployed for real-time object detection. The Notebook showcases YOLOv8's capacity to process unseen data, detecting and classifying objects in diverse environments with impressive speed. These demonstrations highlight YOLOv8's suitability for applications that demand real-time insights, such as surveillance systems and autonomous navigation.

To achieve this and gain a deeper understanding of the process, we referenced a GitHub repository [3] and a YouTube video [4] to create our own implementation for showcasing the results.

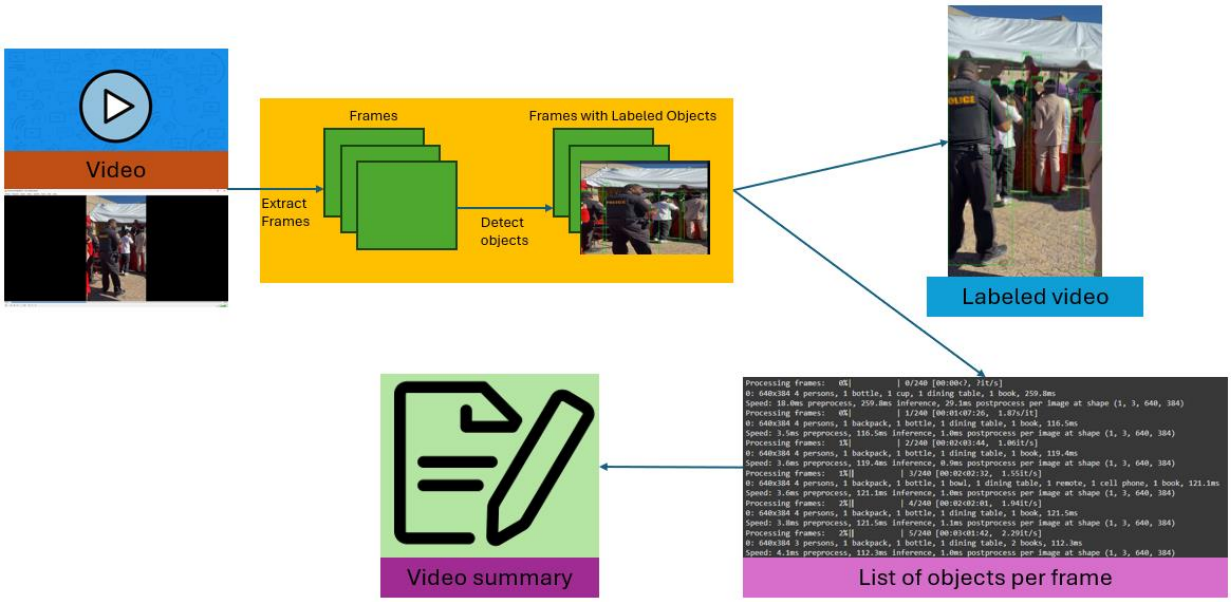

Figure 1. Model Workflow for Video Labelling.

Results

Homecoming Video A

THE SUMMARY OF THE VIDEO IS AS FOLLOWS:

The video captures a festive outdoor event under a large tent, likely a pageant or ceremonial gathering. It features a group of people dressed in formal attire, with some individuals wearing crowns and sashes, suggesting a celebratory occasion. The setting includes decorated tables and balloons, and a police officer is present, ensuring the event runs smoothly.

Screen shot after YOLOv8 it represents detection of the object:

```
Speed: 8.0ms preprocess, 98.1ms inference, 0.9ms postprocess per image at shape (1, 3, 640, 384)
Processing frames:  91%|████████▉ | 143/158 [00:40<00:04,  3.68it/s]
0: 640x384 6 persons, 103.5ms
Speed: 3.3ms preprocess, 103.5ms inference, 0.9ms postprocess per image at shape (1, 3, 640, 384)
Processing frames:  91%|████████▉ | 144/158 [00:40<00:03,  3.69it/s]
0: 640x384 6 persons, 85.9ms
Speed: 3.3ms preprocess, 85.9ms inference, 0.9ms postprocess per image at shape (1, 3, 640, 384)
Processing frames:  92%|█████████ | 145/158 [00:40<00:03,  3.74it/s]
0: 640x384 4 persons, 1 banana, 95.3ms
Speed: 3.9ms preprocess, 95.3ms inference, 0.9ms postprocess per image at shape (1, 3, 640, 384)
Processing frames:  92%|█████████ | 146/158 [00:40<00:03,  3.75it/s]
0: 640x384 5 persons, 1 chair, 91.7ms
Speed: 3.0ms preprocess, 91.7ms inference, 1.0ms postprocess per image at shape (1, 3, 640, 384)
Processing frames:  93%|█████████▏ | 147/158 [00:41<00:02,  3.82it/s]
0: 640x384 5 persons, 1 chair, 91.6ms
Speed: 6.2ms preprocess, 91.6ms inference, 0.9ms postprocess per image at shape (1, 3, 640, 384)
Processing frames:  94%|█████████▏ | 148/158 [00:41<00:02,  3.79it/s]
0: 640x384 5 persons, 1 chair, 82.3ms
Speed: 3.2ms preprocess, 82.3ms inference, 0.9ms postprocess per image at shape (1, 3, 640, 384)
Processing frames:  94%|█████████▏ | 149/158 [00:41<00:02,  3.84it/s]
0: 640x384 4 persons, 1 chair, 91.1ms
Speed: 4.8ms preprocess, 91.1ms inference, 0.9ms postprocess per image at shape (1, 3, 640, 384)
Processing frames:  95%|█████████▎ | 150/158 [00:42<00:02,  3.78it/s]
0: 640x384 5 persons, 1 chair, 84.2ms
Speed: 3.6ms preprocess, 84.2ms inference, 1.0ms postprocess per image at shape (1, 3, 640, 384)
Processing frames:  96%|█████████▎ | 151/158 [00:42<00:01,  3.74it/s]
0: 640x384 4 persons, 1 chair, 86.4ms
Speed: 7.7ms preprocess, 86.4ms inference, 0.9ms postprocess per image at shape (1, 3, 640, 384)
Processing frames:  96%|█████████▎ | 152/158 [00:42<00:01,  3.72it/s]
```

Figure 2. Screenshot of Yolov8 Object Detection HomecomingA.

Homecoming Video B

THE SUMMARY OF THE VIDEO IS AS FOLLOWS:

The video appears to depict a ceremonial or celebratory event taking place outdoors under a tent. It features a formal gathering with individuals wearing sashes and crowns, suggesting a pageant or similar occasion. The presence of police officers implies a security measure. The tent is decorated with balloons, enhancing the festive atmosphere.

This screen shot after YOLOv8

```
Speed: 3.3ms preprocess, 94.1ms inference, 1.0ms postprocess per image at shape (1, 3, 640,
Processing frames:  16%|▉         | 25/158 [00:08<00:36,  3.64it/s]
0: 640x384 5 persons, 1 backpack, 1 umbrella, 143.5ms
Speed: 3.8ms preprocess, 143.5ms inference, 1.5ms postprocess per image at shape (1, 3, 640, 384)
Processing frames:  16%|▉         | 26/158 [00:08<00:39,  3.34it/s]
0: 640x384 5 persons, 1 backpack, 173.9ms
Speed: 3.2ms preprocess, 173.9ms inference, 1.2ms postprocess per image at shape (1, 3, 640, 384)
Processing frames:  17%|▉         | 27/158 [00:09<00:44,  2.93it/s]
0: 640x384 4 persons, 1 backpack, 177.4ms
Speed: 3.3ms preprocess, 177.4ms inference, 1.4ms postprocess per image at shape (1, 3, 640, 384)
Processing frames:  18%|▉         | 28/158 [00:09<00:48,  2.67it/s]
0: 640x384 5 persons, 1 backpack, 157.5ms
Speed: 3.1ms preprocess, 157.5ms inference, 1.5ms postprocess per image at shape (1, 3, 640, 384)
Processing frames:  18%|▉         | 29/158 [00:10<00:49,  2.61it/s]
0: 640x384 5 persons, 1 backpack, 129.9ms
Speed: 3.2ms preprocess, 129.9ms inference, 1.3ms postprocess per image at shape (1, 3, 640, 384)
Processing frames:  19%|▉         | 30/158 [00:10<00:47,  2.67it/s]
0: 640x384 5 persons, 1 backpack, 164.0ms
Speed: 3.7ms preprocess, 164.0ms inference, 1.2ms postprocess per image at shape (1, 3, 640, 384)
Processing frames:  20%|▉         | 31/158 [00:10<00:48,  2.60it/s]
0: 640x384 5 persons, 1 backpack, 150.9ms
Speed: 7.5ms preprocess, 150.9ms inference, 1.8ms postprocess per image at shape (1, 3, 640, 384)
Processing frames:  20%|▉         | 32/158 [00:11<00:49,  2.57it/s]
0: 640x384 4 persons, 1 backpack, 138.7ms
Speed: 3.3ms preprocess, 138.7ms inference, 1.3ms postprocess per image at shape (1, 3, 640, 384)
Processing frames:  21%|▉         | 33/158 [00:11<00:49,  2.51it/s]
```

Figure 3. Screenshot of Yolov8 Object Detection HomecomingB .

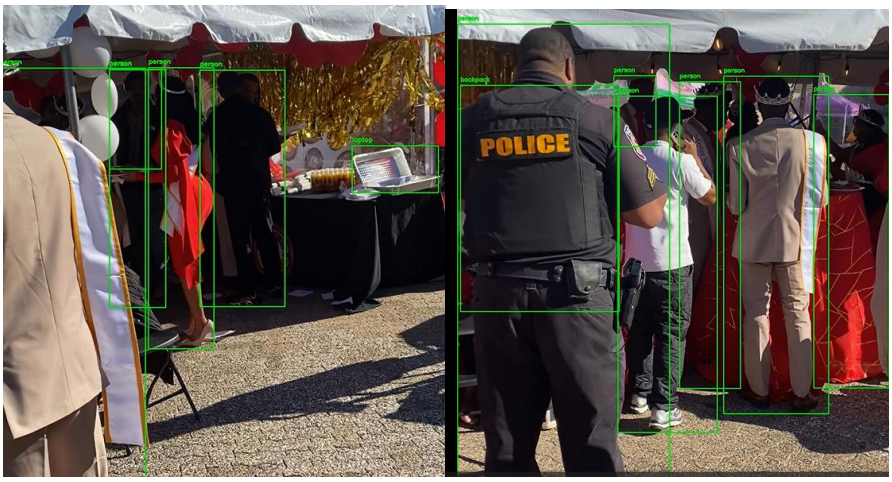

Figure 4. Output of Homecoming VideoB of Yolov8 Object Detection.

Classroom Video:

THE SUMMARY OF THE VIDEO IS AS FOLLOWS:

The video captures a series of scenes in an office or classroom setting, depicting a casual and friendly atmosphere. Throughout the frames, a group of people are seen engaging informally, either seated at tables with items like notebooks, smartphones, wipes, markers, and keys or interacting with each other. The setting includes desks with computers and backpacks, suggesting a working or academic environment. The presence of food and electronic equipment adds to the relaxed and informal vibe of the scenes.

```
Processing frames:  92%|████████  | 221/240 [01:08<00:05,  3.48it/s]
0: 640x384 4 persons, 1 bottle, 1 dining table, 90.8ms
Speed: 3.6ms preprocess, 90.8ms inference, 1.0ms postprocess per image at shape (1, 3, 640, 384)
Processing frames:  92%|████████  | 222/240 [01:08<00:05,  3.46it/s]
0: 640x384 5 persons, 1 bottle, 1 dining table, 1 book, 90.8ms
Speed: 7.6ms preprocess, 90.8ms inference, 0.9ms postprocess per image at shape (1, 3, 640, 384)
Processing frames:  93%|████████  | 223/240 [01:08<00:04,  3.48it/s]
0: 640x384 4 persons, 1 bottle, 1 dining table, 1 book, 103.8ms
Speed: 3.6ms preprocess, 103.8ms inference, 0.9ms postprocess per image at shape (1, 3, 640, 384)
Processing frames:  93%|████████  | 224/240 [01:09<00:04,  3.52it/s]
0: 640x384 5 persons, 1 bottle, 1 dining table, 1 book, 135.0ms
Speed: 4.3ms preprocess, 135.0ms inference, 2.1ms postprocess per image at shape (1, 3, 640, 384)
Processing frames:  94%|████████  | 225/240 [01:09<00:04,  3.37it/s]
0: 640x384 5 persons, 1 bottle, 1 dining table, 1 book, 89.6ms
Speed: 3.2ms preprocess, 89.6ms inference, 0.9ms postprocess per image at shape (1, 3, 640, 384)
Processing frames:  94%|████████  | 226/240 [01:09<00:04,  3.43it/s]
0: 640x384 4 persons, 1 bottle, 1 dining table, 1 book, 107.1ms
Speed: 3.3ms preprocess, 107.1ms inference, 1.0ms postprocess per image at shape (1, 3, 640, 384)
Processing frames:  95%|████████  | 227/240 [01:10<00:03,  3.44it/s]
0: 640x384 4 persons, 1 bottle, 1 dining table, 1 refrigerator, 1 book, 102.9ms
Speed: 3.2ms preprocess, 102.9ms inference, 1.0ms postprocess per image at shape (1, 3, 640, 384)
```

Figure 5. Screenshot of Yolov8 Object Detection Classroom

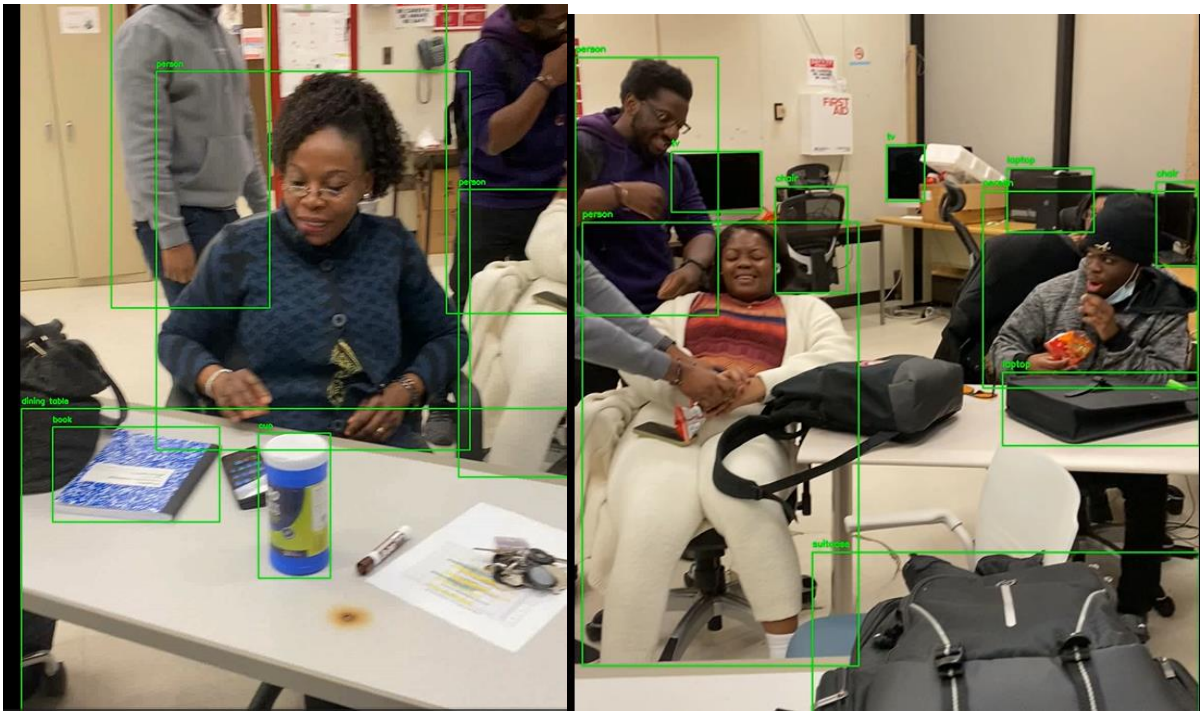

Figure 6: Output of Classroom VideoB of Yolov8 Object Detection.

Conclusion

In conclusion, the development of an automated system using YOLOv8 and ChatGPT for identifying student royal court members demonstrates the potential of integrating advanced object detection and contextual reasoning models. The system effectively leverages YOLOv8's real-time detection capabilities and ChatGPT's interpretive power to address a specific ceremonial context. However, there remains significant room for improvement and expansion.

Future enhancements could involve diversifying datasets, integrating additional AI models for multi-layered verification, optimizing for resource-constrained devices, and ensuring ethical considerations such as privacy and cultural sensitivity. By refining the system and testing it in larger, more diverse environments, this work can contribute to advancements in automated recognition systems, with applications extending beyond ceremonial events to broader fields such as surveillance, education, and public safety.

Future Work

Building on the current system's foundation, future work could explore enhancements to its robustness, scalability, and versatility across diverse contexts. To improve accuracy, the dataset could be expanded to include varied lighting conditions, poses, and attire, along with synthetic data generation to address challenges like occlusion and clutter. Integrating additional AI

models, such as ensemble methods combining YOLOv8 with alternative detection frameworks and leveraging facial recognition or gait analysis could provide further identification layers.

Optimizing the system for edge devices would enable real-time performance on resource-constrained hardware, making it suitable for use on smartphones or drones at events. Contextual awareness could be enhanced by integrating metadata such as event schedules, attendance records, and university databases for automated cross-verification. Furthermore, a user-friendly interface and speech recognition could improve usability for event organizers, while privacy-preserving measures and cultural sensitivity considerations would address ethical concerns. Testing the system at large-scale events would help evaluate its scalability and reliability, and publishing findings or open-sourcing components could foster broader collaboration and innovation in the field.

Again, we shall incorporate Personality Engineering into the YOLOv8 Chat GPT model to enhance personality traits. Enhancing these traits makes the output more intuitive and user friendly which stimulates human-like conversations, while at the same time, enhancing the accuracy, precision and promptness for effective decision making [5].

References

[1] Terven, J., Córdova-Esparza, D. -M., & Romero-González, J. -A. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*, *5*(4), 1680-1716. https://doi.org/10.3390/make5040083

[2] Reis, D., Kupec, J., Hong, J., & Daoudi, A. (2024, May 22). *Real-time flying object detection with yolov8*. arXiv.org. https://arxiv.org/abs/2305.09972

[3] Doleron. (n.d.). *Doleron/Yolov5-opencv-CPP-python: Example of using ultralytics Yolo V5 with OpenCV 4.5.4, C++ and python*. GitHub. https://github.com/doleron/yolov5-opencv-cpp-python

[4] YouTube. (n.d.). *YOLO and ChatGPT for Video Summarization and Understanding: Python Program*. YouTube. https://www.youtube.com/watch?v=syWa2WAVTM0&themeRefresh=1

[5] Yu, B., & Kim, J. (2023). Personality of AI. ArXiv preprint arXiv:2312.02998. https://doi.org/10.48550/arXiv.2312.02998

[6] Raza, M. (2024, January 8). *Yolo V8: A deep dive into its advanced functions and new features*. Medium. https://medium.com/@mujtabaraza194/yolo-v8-a-deep-dive-into-its-advanced-functions-and-new-features-f008599fe604