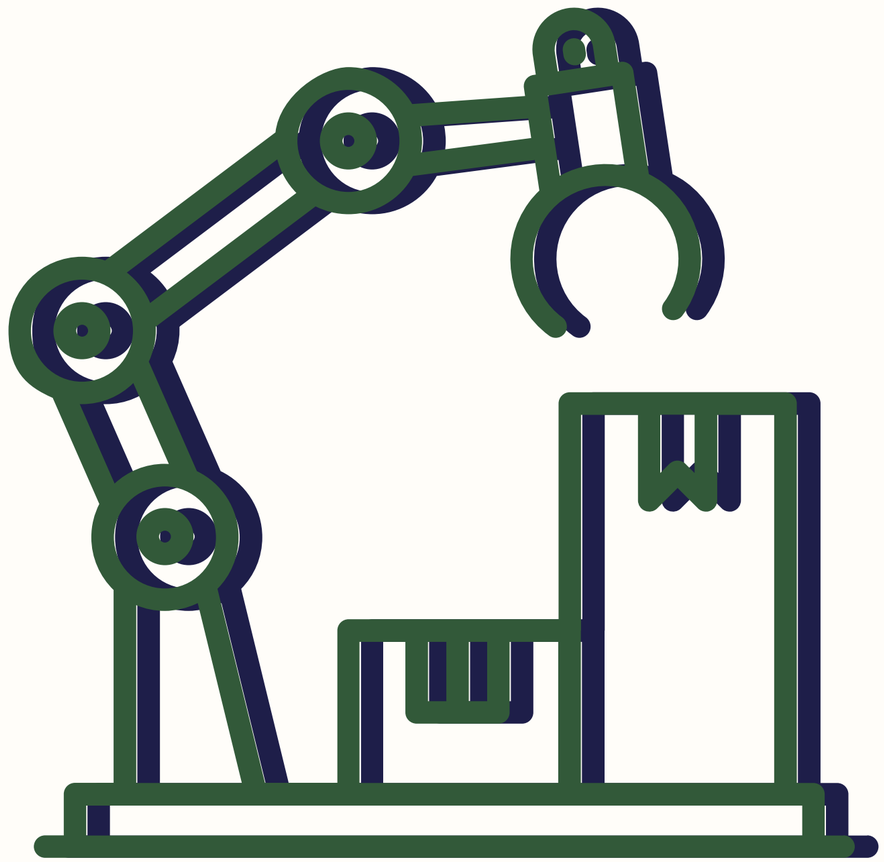
Voice Automated Helping Hand using Object Detection

Project Review

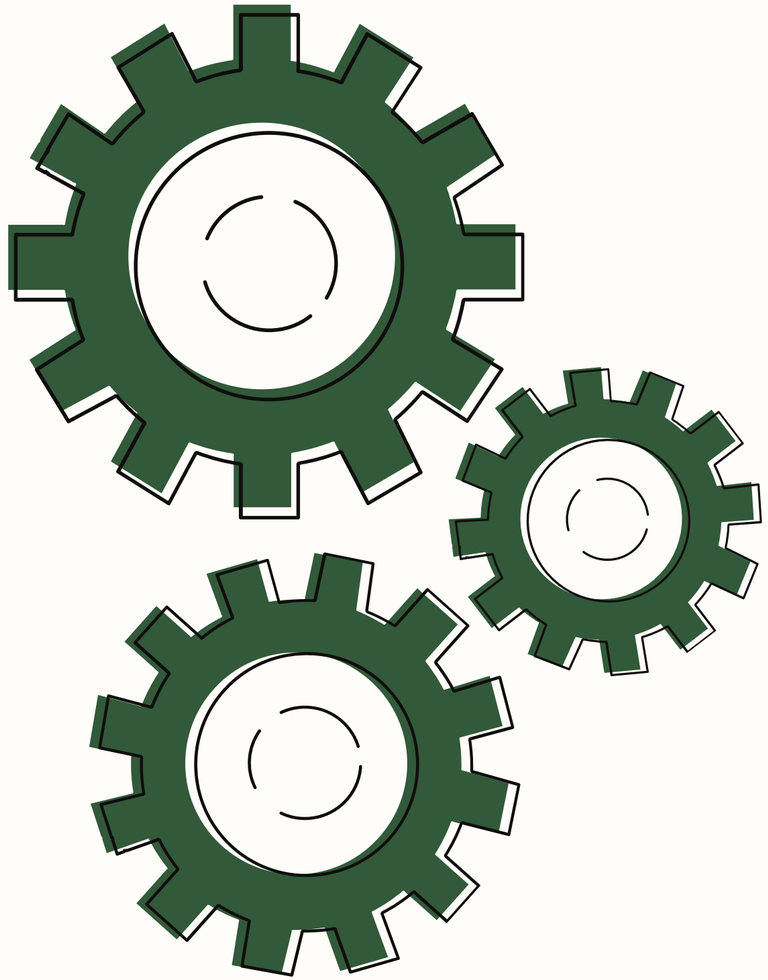
Nigel C. Saldanha (202211298)
Pritesh Falkar (202211783)
Ashwesh Mayekar (202211439)
Atharva Gayakwad (202211358)



Project Guide: Dr. Sangeeta Mahaddalkar

Agenda

1. Introduction
2. Abstract
3. Literature Review
4. Primary and Secondary Objectives
5. Operations flowchart
6. Block diagram, component specification and B.O.M
7. Estimated timeline
8. References
9. Simulation Review



1. Introduction

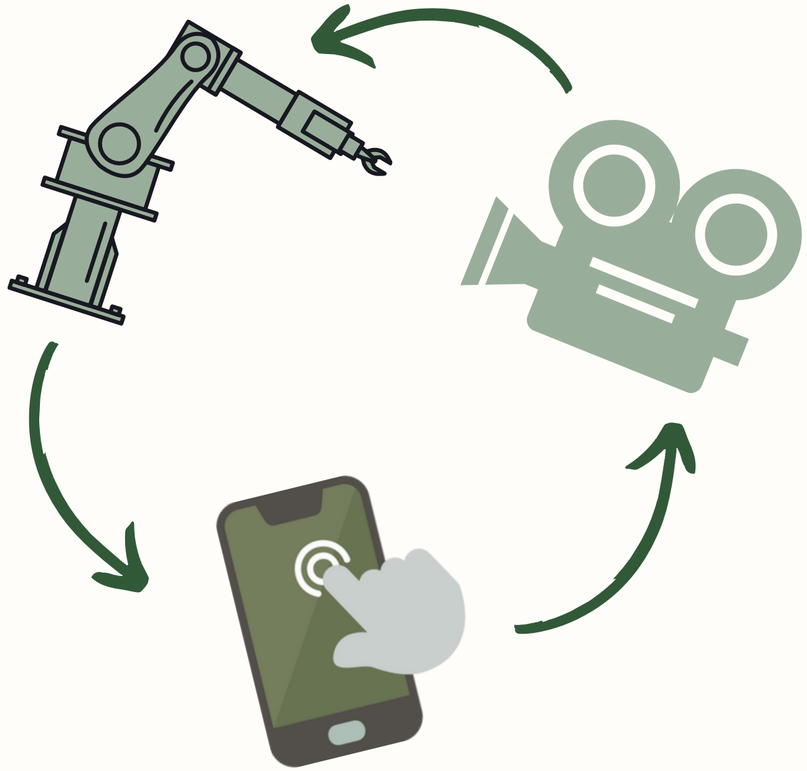
- **Elderly and physically challenged individuals** often face difficulties in performing simple daily tasks like retrieving household objects.
- Existing robotic assistive devices are either too **complex**, **expensive**, or **not user-friendly** for independent use.
- Many available solutions lack natural interfaces (like voice control) and struggle with accurate object detection and reliable grasping.
- There is a gap in designing a **low-cost, intuitive, and accurate** helping hand system that can be practically deployed at home.

Our Approach:

- Develop a **Helping Hand** using:
  - Voice-based commands via a simple mobile app.
  - Dual-camera object detection.
  - A robotic arm modeled with inverse kinematics for reliable grasping.
- Aim: To provide ease of use, high accuracy, and adaptability for assistive living.

2. Abstract

This project presents the design and development of a **Voice Automated Helping Hand**, an intelligent robotic arm that assists users in **retrieving household objects** through **simple voice commands**. A **mobile application** serves as the user interface, enabling real-time **speech-to-text conversion**. A **dual-camera system** is employed, with a stationary overhead camera for global object localization and an arm-mounted camera for local refinement. Object detection is powered by **YOLO-based models**, while arm movements are modeled using and inverse kinematics. The system prioritizes **ease of use**, **reliability**, and **accuracy**, aiming to serve as a **practical assistive device** for elderly and physically challenged individuals.



3. Literature Review Summary:

S.No.	Paper Title	Author(s)/Year	Technologies/Methodologies	Key Results/findings	Remarks / Notes
[1]	Voice Controlled personal assistant robot for elderly people	Jishnu U.K.; Indu V.; K.J. Ananthakrishnan; Korada Amith; P Sidharth Reddy; Pramod S. 2020 5th International Conference on Communication and Electronics Systems (ICCES)	• Rasp pi 3B • YOLO • HC05 bluetooth mode • Four wheeled	• FSR - shows inverse relationship between resistance and applied force. i.e. voltage rises as resistance decreases. Can be effectively used to measure gripping force (low force-low voltage and vice versa)	• FSR grasping force
[2]	Assistive device for physically challenged person using voice controlled intelligent robotic arm	Ripcy Anna John; Sneha Varghese; Sneha Thankam Shaji; K.Martin Sagayam 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)	• 1.Voice commands • 4 DOF arm • Joystick • Rail system NEMA17 • Google Assistant	• Phoneme based approach acoustic model (86.5% acc) • SSD mobile net regression model (87.3%) • GoogleNet (78.5%) • ImageNet (87.8%)	• Success based on user acceptability • SNR (signal to noise ratio) 74% accuracy
[3]	Trends in service robots for the disabled and the elderly (ISAC-HERO system)	K. Kawamura; M. Iskarous Proceedings of IEEE/RSJ International Conference Intelligent Robots and Systems (IROS'94) Year: 1994 | Conference Paper | Publisher:IEEE	• Fuzzy command voice interpreter • Macro action builder • Blackboard logic for communication • Task planning - complex tasks broken into sub tasks • PID tuning on each joint, transputer based controller • Macvicar Whelan fuzzy	• Success based on user acceptability (user interface, learning and adaptation)	-
[4]	ROS based control of robot using voice recognition	Rajesh Kannan Megalingam; Racharla Shriya Reddy; Yannam Jahnavi; Manaswini Motheram 2019 Third International Conference on Inventive Systems and Control (ICISC) Year: 2019 | Conference Paper | Publisher: IEEE	• HMM model for speech enhancement • Pocket sphinx (offline) • Rasp pi 3B • ROS • Virtebri algo for best path to decode speech	-	• Uses recognizer and mediator programs

S.No.	Paper Title	Author(s)/Year	Technologies/Methodologies	Key Results/findings	Remarks / Notes
[5]	Small scale robot arm design with pick and place mission based on inverse kinematics	Adnan Rafi Al Tahtawi , Muhammad Agni , Trisiani Dewi Hendrawati	• Simple Kinematics equations for 3 DOF arm	-	-
[6]	Design, construction and control of SCARA prototype with 5 DOF	Delond Angelo Jimenez-Nixon; María Celeste Paredes-Sánchez; Alicia María Reyes-Duke 2022 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT) Year: 2022 | Conference Paper | Publisher: IEEE	• Denavit-Hartenberg algo and inverse kinematics using • Geometric method • 4 DOF	• Height limitation	• Used V-model for design planning and implementation
[7]	Designing 8 Degrees of Freedom Humanoid robotic arm	Le Bang Duc; Mohd Syaifuddin; Truong Trong Toai; Ngo Huy Tan; Mohd Naufal Saad; Lee Chan Wai 2007 International Conference on Intelligent and Advanced Systems Year: 2007 | Conference Paper | Publisher: IEEE	• 8 DOF robotic hand	-	• Shows different grasping techniques in case we want to make the end effector more versatile in picking different objects.
[8]	Swab-bot - an oral swabbing robotic arm	John Varghese Panicker; Divy Jain; Vibodh H. N; T. Vishnu Yeshwanth; Swetha Ramaiah 2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME) Year: 2023 | Conference Paper | Publisher: IEEE	• Rasp pi 4B • Adafruit servo hat • Rasp pi cam	• Contains useful kinematics equations for 4 DOF arm	• Arm mounted camera
[9]	Voice conversion based augmentation and a Hybrid CNN-LSTM model for improving speaker independent keyword recognition on limiited datasets	Yeshanew Ale Wubet; Kuang-Yow Lian IEEE Access Year: 2022 | Volume: 10 | Journal Article | Publisher: IEEE	• Speaker independent and detects keywords • ACVAE • Data augmentation (creates new slightly altered sample from original to augment data set) • Pytorch, Keras, Tensorflow	• 96% accuracy	• Speakers from 3 different countries, 12 keywords spoken 10 times
[10]	An MFCC-based Secure Framework for Voice Assistant Systems	Syed Fahad Ahmed;Rabeea Jaffari; Moazzam Jawaid; Syed Saad Ahmed;Shahnawaz Talpur 2022 International Conference on Cyber Warfare and Security (ICCWS) Year: 2022 | Conference Paper | Publisher: IEEE	• 1. Mel-frequency cepstral coefficients based user authenticated security	• 1.Trained on 10 authentic users in 5 conditions: 84% accuracy in normal condition and 59% accuracy in illness condition	• VC arms have limited actions and can take up a lot of electricity

S.No.	Paper Title	Author(s)/Year	Technologies/Methodologies	Key Results/findings	Remarks / Notes
[11]	Comparative study on various architecture of YOLO models used in object recognition	Baranidharan Balakrishnan; Rashmi Chelliah; Madhumitha Venkatesan; Chetan Sah 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) Year: 2022 | Conference Paper | Publisher: IEEE	-	• Darknet architecture gives the maximum accuracy among all the other architectures used. Following those come the Keras architecture and the ImageAI library, where in the ImageAI library, Tiny architecture gave the maximum among the other architectures in that library	-
[12]	Design a Human-robot interaction framework to detect household objects	Sadi Rafsan; Safayet Arefin; A. H. M. Mirza Rashedul Hasan; Mohammed Moshiul Hoque 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV) Year: 2016 | Conference Paper | Publisher: IEEE	• 3 methods to detect objects: size, color, position • Haar, openCV_traincascade, GentleAdaBoost	• 88% accuracy • 11.06% false positive	• Text based interaction
[13]	Object detection using YOLO-V8	B Karthika; M Dharssinee; V Reshma; R Venkatesan; R Sujarani 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT) Year: 2024 | Conference Paper | Publisher: IEEE	• DETR • EfficientDet • Roboflow for annotating	• 300 epochs, 85% precision, 80% recall, 82% mAP	-
[14]	Design and implementation of a robotic arm assistant with voice interaction using machine vision	Nantzios, G.; Baras, N.; Dasygenis, M. Design and Implementation of a Robotic Arm Assistant with Voice Interaction Using Machine Vision. Automation 2021, 2, 238–251. https://doi.org/10.3390/automation2040015	Nantzios, G.; Baras, N.; Dasygenis, M. Design and Implementation of a Robotic Arm Assistant with Voice Interaction Using Machine Vision. Automation 2021, 2, 238–251. https://doi.org/10.3390/automation2040015	• 90% accuracy (96% with QR) • 90% voice recognition (96% with QR) • Drop item failure 3% • Avg execution time 52secs	• Arm range 200 degrees • Movement speed reduced to 30% of attainable speed
[15]	Robotic arm vehicle using voice recognition for challenged people	Jia Nannda; Lokareddy Venkanna Dora; Tanvi Gupta; Aryan Chouhan; Anmol Verma 2024 2nd International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)	• Vehicle • Predefined commands for voice recognition • Ultrasonic sensors	• Total time completion formulas	-

4.1 Primary objectives:

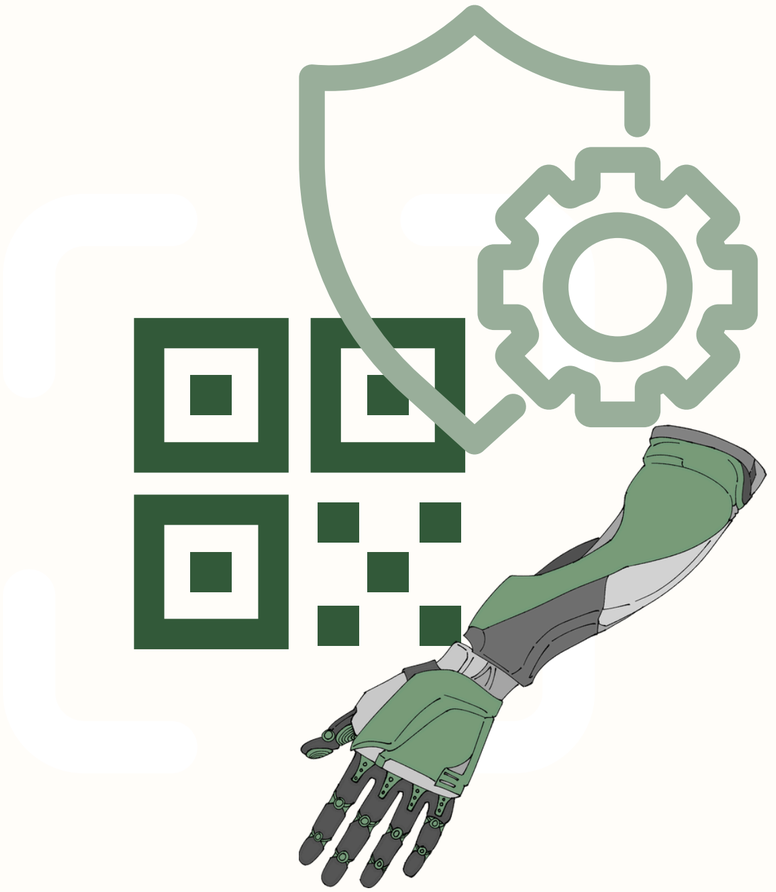
- **Build simple one button app**
  Aim: To design a user friendly app catered to an elderly audience and achieve voice accuracy >90%
  - App using MIT app inventor/ Flutter
  - OpenAI whisper/Vosk/SpeechRecognition for Speech-to-text

- **Using YOLOe/YOLOv12 for object detection**
  Aim: Reduce false positives <11.06% and achieve detection accuracy >90%
  - Dual-camera localization (stationary Darknet based model + dynamic arm mounted TinyYOLO based model)
  - Roboflow for annotating and labelling

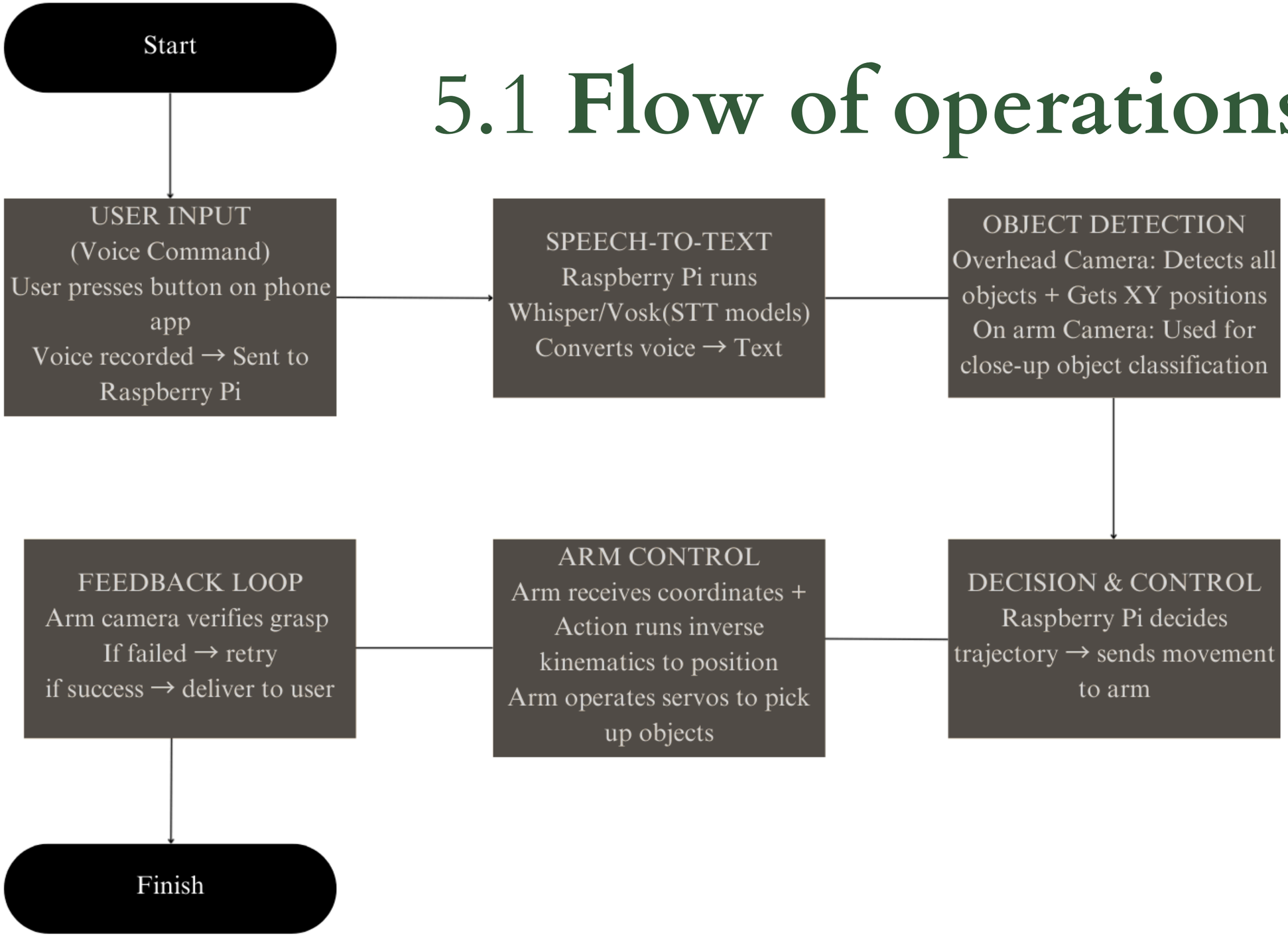
- **Arm mechanics**
  Aim: Reduce drop item failure <3%
  - Denavit-Hartenberg and geometric inverse kinematics to model arm movements mathematically
  - Implement appropriate gripping using FSR

4.2 Secondary objectives:

- Implement rail-based system across workspace to extend horizontal picking range
- Improve upon 74% SNR using HMM + Viterbi decoding
- Fine tune arm movements using PID and picking accuracy
- Lightweight data augmentation to artificially expand small speech datasets
- Use QR code object detection approach for commonly used household items
- Improve upon existing end-effector design to optimize picking objects of different shapes and sizes
- Security layer for voice recognition



5.1 Flow of operations



5.2 High level overview

- **User Input (App Interface)**
  - The user presses and holds a large button on the app.
  - While holding, the app records the audio command.
  - When released, recording stops and the audio is sent to the bot.
  - The app then converts speech to text and displays it on screen for user verification.

- **Command Verification**
  - If the text matches what the user intended, the app sends it forward.
  - If not, the user can re-record.
  - Once verified, the command is passed to the bot (Raspberry Pi).

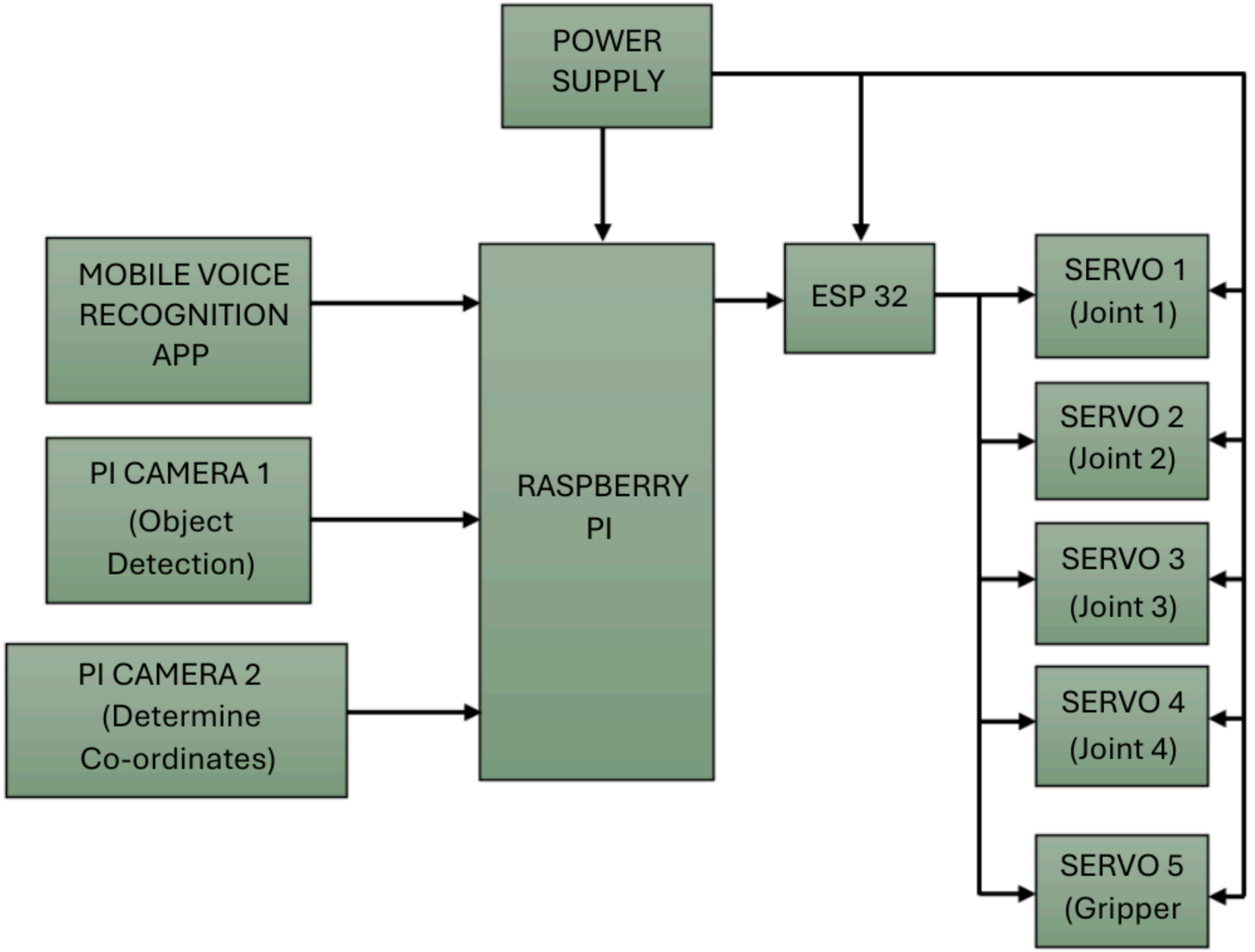
- **Stationary Camera and Dynamic Arm-mounted camera**
  - A fixed overhead camera scans the workspace.
  - A YOLO model identifies all visible objects and determines their approximate (x,y) coordinates and the number of objects using primary camera and identification of the objects is done by the dynamic cam.

- **Arm Movement Initialization**
  - The ESP32 controlling the robotic arm receives the target coordinates.
  - Using Denavit-Hartenberg + inverse kinematics, the arm is moved toward the rough position.

- **Grasp Execution**
  - The end-effector (gripper) positions itself grasps the object.

- **Object Retrieval**
  - Once the object is secured, the arm lifts it.
  - The system calculates the path back (simple return path).

6.1 Block Diagram



6.2 Component specifications

1) Raspberry Pi 5

Processor	Broadcom BCM2712 2.4GHz quad-core 64-bit Arm Cortex-A76 CPU,, 512KB per-core L2 caches and a 2MB shared L3 cache
Memory	16 GB
Connectivity	Dual-band 802.11ac Wi-Fi®, Bluetooth 5.0 / Bluetooth Low Energy (BLE), 2 × USB 3.0 & 2 × USB 2.0 ports
Video/Audio	VideoCore VII GPU, supporting OpenGL ES 3.1, Vulkan 1.3 Dual 4Kp60 HDMI® display output with HDR support 4Kp60 HEVC decoder
Storage	Micro-SD card slot
Power	5V/5A DC power via USB-C, with Power Delivery support
GPIO	40-pin header
Temperature	0–50°C

2) MG996R Servo Motor

Type	High-torque metal-geared servo
Torque	11 kg-cm @ 6V
Rotation Angle	~360°
Voltage	4.8 – 7.2 V

3) MG90S Micro Servo

Type	Metal-geared micro servo
Torque	2.2 kg-cm @ 6V
Rotation	180°
Voltage	4.8 – 6 V

4) Raspberry Pi 5MP camera

Video	1080p@30fps, 720p@60fps, 480p@100fps
Field of View	75° (Standard)
Interface	CSI connector (200 mm ribbon)
Size	20 x 25 x 9mm

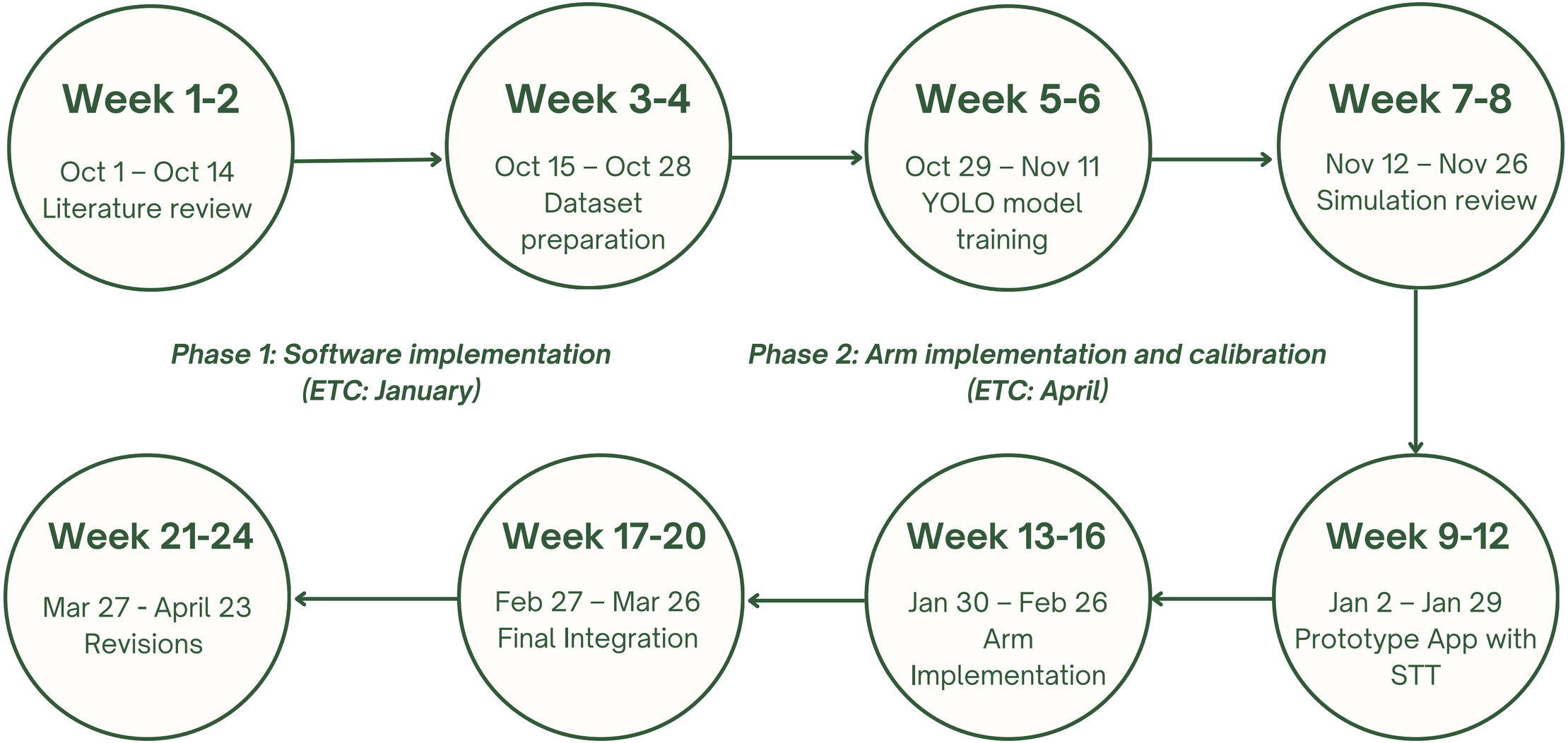
5) PCA9685 driver

Channels	16 independent PWM outputs.
Interface	I²C (Fast-mode Plus, up to 1 MHz).
Resolution	12-bit (4096 steps) per channel
Supply voltage	2.3V to 5.5V
Output current	25mA sink, 10mA source (at 5V)

6.3 Estimated B.O.M.

Sr. No	Components	Quantity	Estimated Price
1	Raspberry PI 5	1	9000 Rs
2	PCA9685 driver	1	219 Rs
3	MG996R Servo Motor	3	1,119 Rs
4	SG90S Micro Servo	2	378 Rs
5	Raspberry PI 5MP Cam	2	500 Rs
		Total Price	11,216 Rs

7. Project Timeline



8. References

- **[1]** J. U.K., I. V., K. J. Ananthakrishnan, K. Amith, P. S. Reddy and P. S., Voice Controlled Personal Assistant Robot for Elderly People, 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2020, pp. 269-274, doi: 10.1109/ICCES48766.2020.9138101.
- [**2**] R. A. John, S. Varghese, S. T. Shaji and K. M. Sagayam, Assistive Device for Physically Challenged Persons Using Voice Controlled Intelligent Robotic Arm, 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2020, pp. 806-810, doi: 10.1109/ICACCS48705.2020.9074236.
- **[3]** K. Kawamura and M. Iskarous, Trends in service robots for the disabled and the elderly, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94), Munich, Germany, 1994, pp. 1647-1654 vol.3, doi: 10.1109/IROS.1994.407636.
- **[4]** R. K. Megalingam, R. S. Reddy, Y. Jahnavi and M. Motheram, ROS Based Control of Robot Using Voice Recognition, 2019 Third International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 2019, pp. 501-507, doi: 10.1109/ICISC44355.2019.9036443.
- **[5]** Al Tahtawi, Adnan & Agni, Muhammad & Hendrawati, Trisiani. (2021). Small-scale Robot Arm Design with Pick and Place Mission Based on Inverse Kinematics. Journal of Robotics and Control (JRC). 2. 10.18196/26124.
- **[6]** D. A. Jimenez-Nixon, M. C. Paredes-Sánchez and A. M. Reyes-Duke, Design, construction and control of a SCARA robot prototype with 5 DOF, 2022 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT), Soyapango, El Salvador, 2022, pp. 1-6, doi: 10.1109/ICMLANT56191.2022.9996479.
- **[7]** L. B. Duc, M. Syaifuddin, T. T. Toai, N. H. Tan, M. N. Saad and L. C. Wai, Designing 8 Degrees of Freedom Humanoid Robotic Arm, 2007 International Conference on Intelligent and Advanced Systems, Kuala Lumpur, Malaysia, 2007, pp. 1069-1074, doi: 10.1109/ICIAS.2007.4658549.

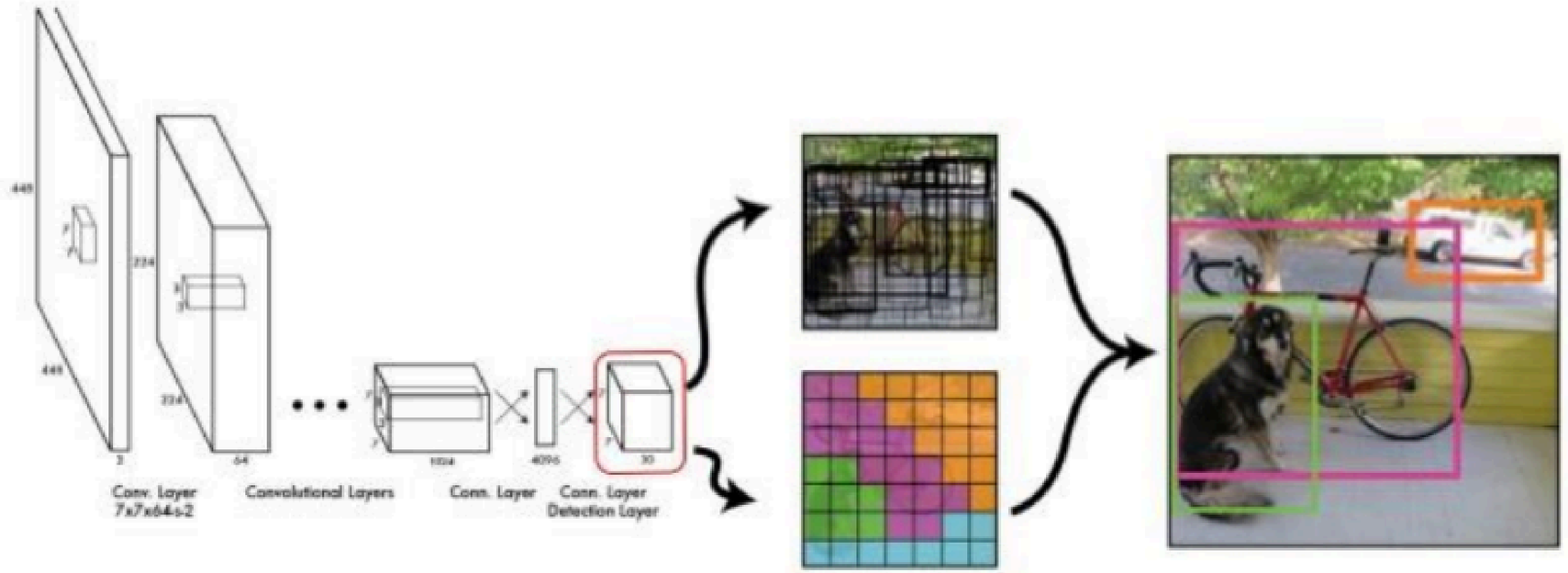
- **[8]** Panicker, John & Jain, Divy & N, Vibodh & Yeshwanth, T. & Ramaiah, Swetha. (2023). SwabBot-An oral Swabbing Robotic arm. 1-6. 10.1109/ICECCME57830.2023.10252564.
- **[9]** Y. A. Wubet and K. -Y. Lian, Voice Conversion Based Augmentation and a Hybrid CNN-LSTM Model for Improving Speaker-Independent Keyword Recognition on Limited Datasets, in IEEE Access, vol. 10, pp. 89170-89180, 2022, doi: 10.1109/ACCESS.2022.3200479.
- **[10]** S. F. Ahmed, R. Jaffari, M. Jawaid, S. S. Ahmed and S. Talpur, An MFCC-based Secure Framework for Voice Assistant Systems, 2022 International Conference on Cyber Warfare and Security (ICCWS), Islamabad, Pakistan, 2022, pp. 57-61, doi: 10.1109/ICCWS56285.2022.9998446.
- **[11]** B. Balakrishnan, R. Chelliah, M. Venkatesan and C. Sah, Comparative Study On Various Architectures Of Yolo Models Used In Object Recognition, 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 2022, pp. 685-690, doi: 10.1109/ICCCIS56430.2022.10037635.
- **[12]** S. Rafsan, S. Arefin, A. H. M. M. R. Hasan and M. M. Hoque, Design a human-robot interaction framework to detect household objects, 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV), Dhaka, Bangladesh, 2016, pp. 973-978, doi: 10.1109/ICIEV.2016.7760144.
- **[13]** B. Karthika, M. Dharssinee, V. Reshma, R. Venkatesan and R. Sujarani, Object Detection Using YOLO-V8, 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, 2024, pp. 1-4, doi: 10.1109/ICCCNT61001.2024.10724411.
- **[14]** Nantzios, G.; Baras, N.; Dasygenis, M. Design and Implementation of a Robotic Arm Assistant with Voice Interaction Using Machine Vision. Automation 2021, 2, 238-251.
- **[15]** J. Nannda, L. V. Dora, T. Gupta, A. Chouhan and A. Verma, A Robotic Arm Vehicle using Voice Recognition for Physically Challenged People, 2024 2nd International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT), Faridabad, India, 2024, pp. 858-862, doi: 10.1109/ICAICCIT64383.2024.10912202.

Review-I Summary:

- COCO dataset → 80 classes
  - Did not contain Medicine box class & Remotes
    - Custom dataset was made.

- ESP32 vs PCA9685 Driver
  - Arm design → 5 DOF
    - 5 PWM pins required from the Pi 5, but it only has 4(GPIO 13&19 PWM(1) and GPIO 12&18 PWM(0))
    - 2 Hardware PWMs and 2 Software PWMs
  - Due to this insufficiency the PCA9685 Driver (16 channels) is used.

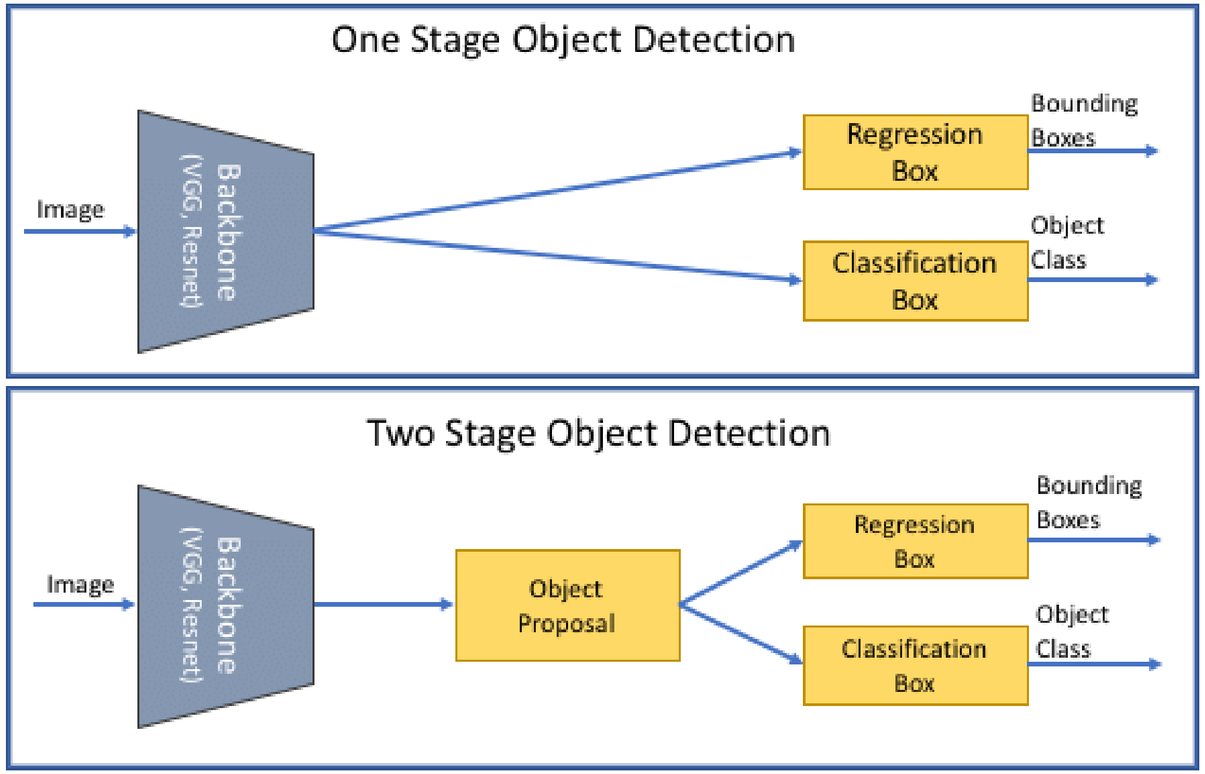
YOLO Architecture:



- Input Image
  - A full image is given to a single Convolutional Neural Network (CNN).
- Feature Extraction (Backbone)
  - Convolution layers extract important features like edges, shapes, and objects.
- Grid Division
  - The image is divided into an S × S grid.
  - Each grid cell is responsible for detecting objects whose center lies inside it.
- Prediction Head
  - Each grid cell predicts:
    - Bounding Box → (x, y, width, height)
    - Objectness Score → confidence that an object exists
    - Class Probabilities → object category (person, bottle, phone, etc.)

YOLO vs Other models:

- Single-Stage Detection:
  - YOLO uses one CNN and one forward pass
  - Other models use multiple stages (region proposals + classification)
  - ➜ Much faster detection
- Real-Time Performance:
  - Runs at 30–60 FPS
  - Suitable for live video, robotics, and surveillance
- Global Image Understanding:
  - YOLO looks at the entire image at once
  - Reduces false detections compared to region-based methods
- End-to-End Simplicity:
  - Directly predicts bounding boxes + class labels
  - Easier to train, deploy, and optimize
- Edge & Embedded Friendly:
  - Lightweight versions work on low-power devices
  - Other CNN detectors are computationally heavy



Dataset Preparation:

- To evaluate and compare YOLO models for our Helping Hand Bot, we created a custom dataset tailored to our use case rather than relying on generic datasets.
  - **COCO dataset → 80 classes limitation**
- The dataset focuses on **common household items (8 classes)** that the bot is expected to identify (cup, banana, apple, orange, medicines, remotes, glass, water bottle).
- Around 1740 images were collected and **annotated using Roboflow** in YOLO format.
- The dataset was split into **70% training, 20% validation, and 10% testing** subsets **(1387 Training, 181 Validation, 119 Test)**
- This setup enables us to accurately measure precision, recall, and mAP for each YOLO variant and identify which model performs best for real-world deployment.

All Splits | Train | Valid | Test

Sort

Remote 838

Glass 459

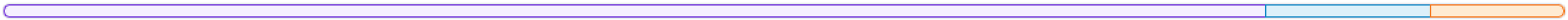
Water Bottle 444

Medicine 213

Cup 152

Apple 147

Banana 104



Orange 82

Model Training Overview:

- We trained five YOLO variants — **YOLOv8n, YOLOv8m, YOLOv9c, YOLOv11s, and YOLOv11m** — using a traditional supervised training pipeline.
- Training included standard steps:
  - Loading labeled images (train/val/test split)
  - Learning bounding boxes + class predictions
  - Evaluating accuracy using **mAP, precision** and **recall**
- This allowed a fair, controlled comparison across all YOLO versions using identical data and hyperparameters.
- **YOLO-E was not trained on image labels**. Instead, it was trained using **text prompts**, following its open-vocabulary / zero-shot pipeline to evaluate how well it can identify objects without supervised training.
- This dual-approach comparison helped us understand how traditional supervised YOLO models differ from prompt-driven models in accuracy, generalization, and practical usability for our robot assistant.

Model Validation Overview:

```
Validating /content/runs/detect/yolov8n_custom/weights/best.pt...
Ultralytics 8.3.223 🚀 Python-3.12.12 torch-2.8.0+cu126 CUDA:0 (Tesla T4, 15095MiB)
Model summary (fused): 72 layers, 3,007,208 parameters, 0 gradients, 8.1 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ━━━━━━━━━━━━ 3/3 1.4it/s 2.2s
                   all        181        196      0.946       0.94      0.964      0.851
                 Apple         17         17      0.935          1      0.992      0.824
                Banana         11         11      0.984          1      0.995      0.832
                   Cup         17         22          1      0.782      0.942      0.876
                 Glass         21         36      0.841      0.881      0.877      0.754
              Medicine         20         20      0.882          1      0.988      0.933
                Orange         17         17      0.984          1      0.995      0.921
                Remote         17         21      0.993      0.952       0.96       0.86
           Water Bottle         37         52      0.953      0.904      0.966      0.807
Speed: 0.2ms preprocess, 1.6ms inference, 0.0ms loss, 3.8ms postprocess per image
Results saved to /content/runs/detect/yolov8n_custom
YOLOv8n completed in 13.6 minutes
```

YOLO 8n

```
Validating /content/runs/detect/yolov8m_custom/weights/best.pt...
Ultralytics 8.3.223 🚀 Python-3.12.12 torch-2.8.0+cu126 CUDA:0 (Tesla T4, 15095MiB)
Model summary (fused): 92 layers, 25,844,392 parameters, 0 gradients, 78.7 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ━━━━━━━━━━━━ 6/6 1.7it/s 3.5s
                   all        181        196      0.954      0.935      0.968      0.857
                 Apple         17         17       0.99          1      0.995       0.83
                Banana         11         11       0.98          1      0.995      0.852
                   Cup         17         22          1      0.812      0.951      0.877
                 Glass         21         36      0.779      0.784      0.855      0.737
              Medicine         20         20      0.948          1      0.993      0.955
                Orange         17         17      0.976          1      0.995      0.914
                Remote         17         21          1      0.942      0.986      0.868
           Water Bottle         37         52      0.961      0.941      0.971      0.824
Speed: 0.3ms preprocess, 7.3ms inference, 0.0ms loss, 5.3ms postprocess per image
Results saved to /content/runs/detect/yolov8m_custom
YOLOv8m completed in 32.0 minutes
```

YOLO 8m

```
Validating /content/runs/detect/yolo11m_custom/weights/best.pt...
Ultralytics 8.3.223 🚀 Python-3.12.12 torch-2.8.0+cu126 CUDA:0 (Tesla T4, 15095MiB)
YOLO11m summary (fused): 125 layers, 20,036,200 parameters, 0 gradients, 67.7 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ━━━━━━━━━━━━ 6/6 1.9it/s 3.1s
                   all        181        196      0.952      0.925      0.962      0.851
                 Apple         17         17      0.974          1      0.995      0.812
                Banana         11         11      0.985          1      0.995      0.845
                   Cup         17         22          1      0.801       0.91      0.856
                 Glass         21         36      0.822      0.833      0.851      0.761
              Medicine         20         20      0.855          1      0.995      0.956
                Orange         17         17      0.984          1      0.995      0.929
                Remote         17         21          1      0.842      0.993      0.862
           Water Bottle         37         52          1      0.923      0.963      0.786
Speed: 0.3ms preprocess, 8.2ms inference, 0.0ms loss, 4.2ms postprocess per image
Results saved to /content/runs/detect/yolo11m_custom
YOLOv11m completed in 36.0 minutes
```

YOLO 11m

```
Validating /content/runs/detect/yolo11s_custom2/weights/best.pt...
Ultralytics 8.3.223 🚀 Python-3.12.12 torch-2.8.0+cu126 CUDA:0 (Tesla T4, 15095MiB)
YOLO11s summary (fused): 100 layers, 9,415,896 parameters, 0 gradients, 21.3 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ━━━━━━━━━━━━ 4/4 1.3it/s 3.0s
                   all        181        196      0.945      0.943      0.968      0.862
                 Apple         17         17      0.932          1      0.992      0.824
                Banana         11         11          1      0.989      0.995      0.841
                   Cup         17         22          1      0.837      0.946      0.873
                 Glass         21         36      0.822      0.896      0.873      0.791
              Medicine         20         20      0.915       0.95      0.985      0.938
                Orange         17         17      0.978          1      0.995      0.948
                Remote         17         21      0.973      0.952       0.98      0.868
           Water Bottle         37         52       0.94      0.923      0.977      0.811
Speed: 0.2ms preprocess, 4.2ms inference, 0.0ms loss, 3.1ms postprocess per image
Results saved to /content/runs/detect/yolo11s_custom2
YOLOv11s completed in 17.3 minutes
```

YOLO 11s

```
Validating /content/runs/detect/yolov9c_custom/weights/best.pt...
Ultralytics 8.3.223 🚀 Python-3.12.12 torch-2.8.0+cu126 CUDA:0 (Tesla T4, 15095MiB)
YOLOv9c summary (fused): 156 layers, 25,325,416 parameters, 0 gradients, 102.4 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ──────────── 6/6 1.7it/s 3.6s
                   all        181        196      0.931      0.949      0.962       0.85
                 Apple         17         17      0.929          1      0.995      0.825
                Banana         11         11      0.969          1      0.995      0.857
                   Cup         17         22          1      0.891      0.949      0.867
                 Glass         21         36      0.764      0.806      0.834      0.732
              Medicine         20         20        0.9          1      0.993      0.961
                Orange         17         17      0.976          1      0.995      0.949
                Remote         17         21      0.964      0.952      0.961      0.803
          Water Bottle         37         52      0.942      0.942      0.974      0.806
Speed: 0.2ms preprocess, 11.6ms inference, 0.0ms loss, 3.2ms postprocess per image
Results saved to /content/runs/detect/yolov9c_custom
YOLOv9c completed in 55.4 minutes
```

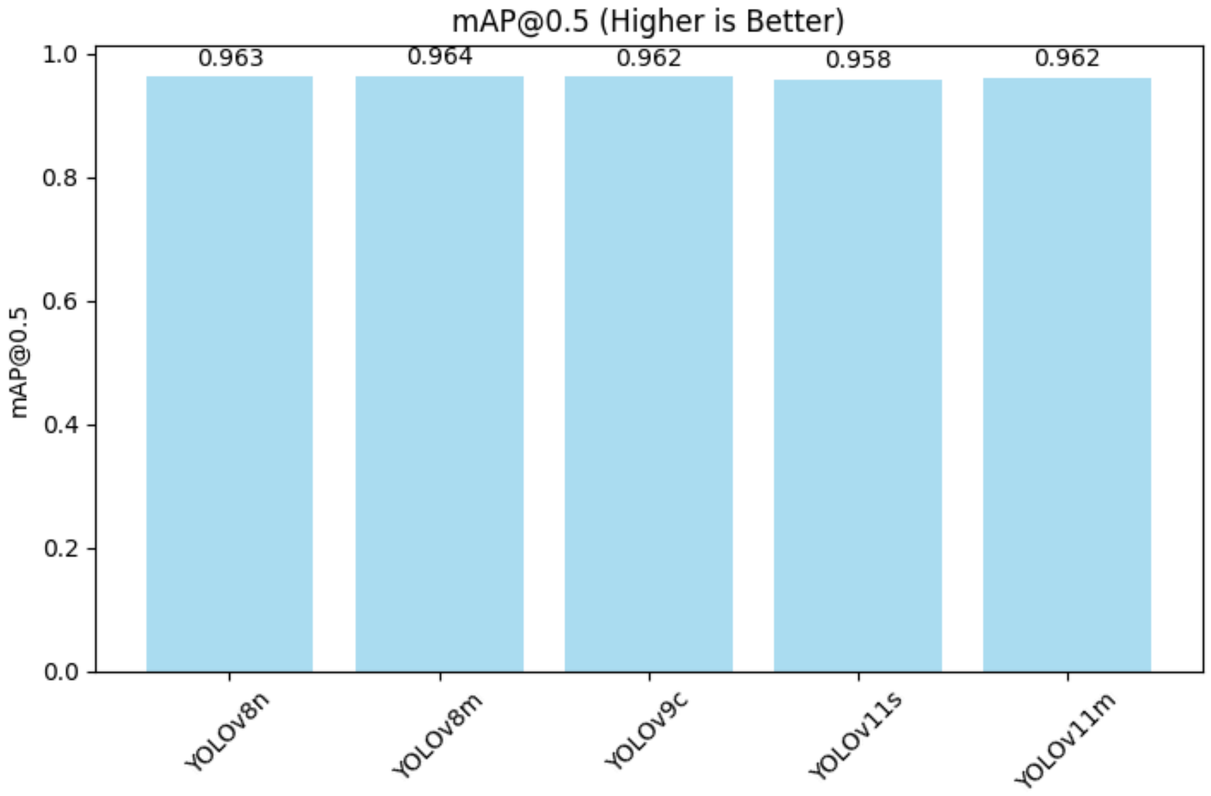
YOLO 9c

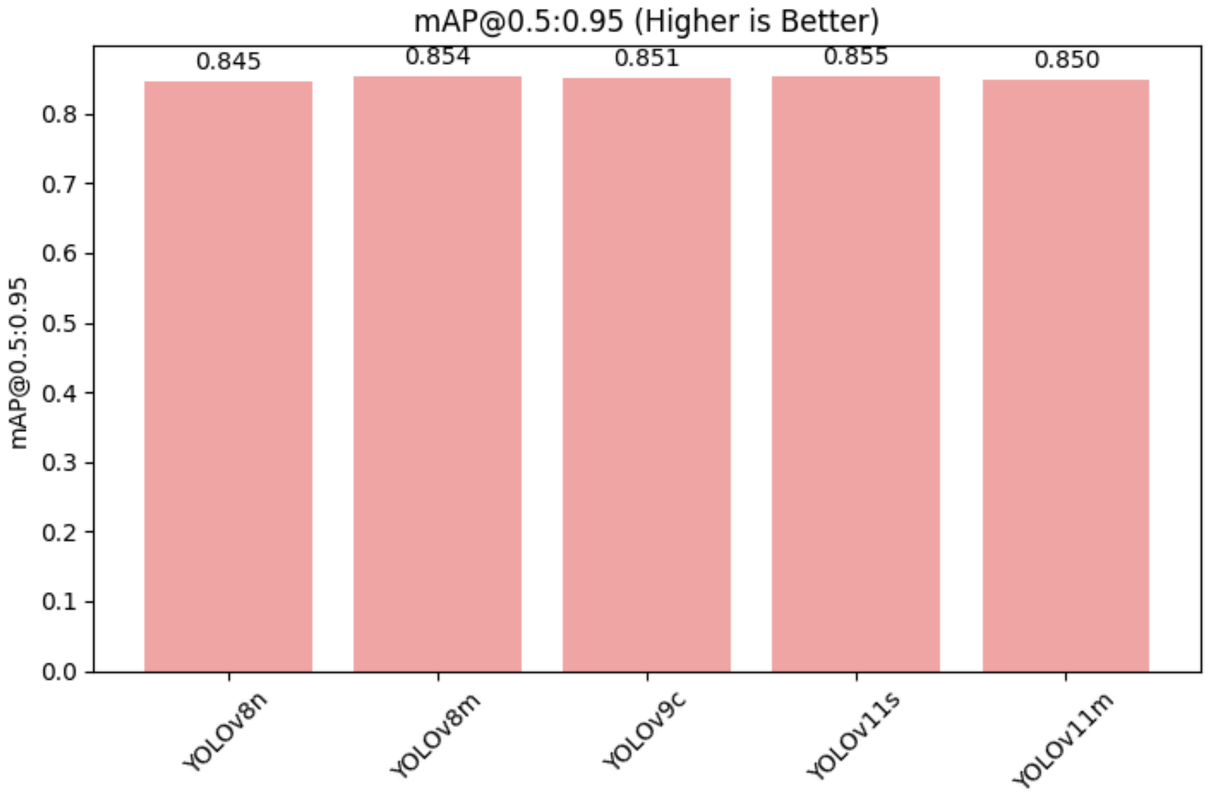
Final validation results:

	Model	mAP50	mAP50_95	Precision	Recall	Model_Size_MB	Inference_Speed_ms
0	YOLOv8n	0.963	0.845	0.935	0.955	5.944	99.001
1	YOLOv8m	0.964	0.854	0.950	0.943	49.609	350.255
2	YOLOv9c	0.962	0.851	0.933	0.952	49.192	406.088
3	YOLOv11s	0.958	0.855	0.941	0.936	18.275	270.905
4	YOLOv11m	0.962	0.850	0.952	0.925	38.629	276.637

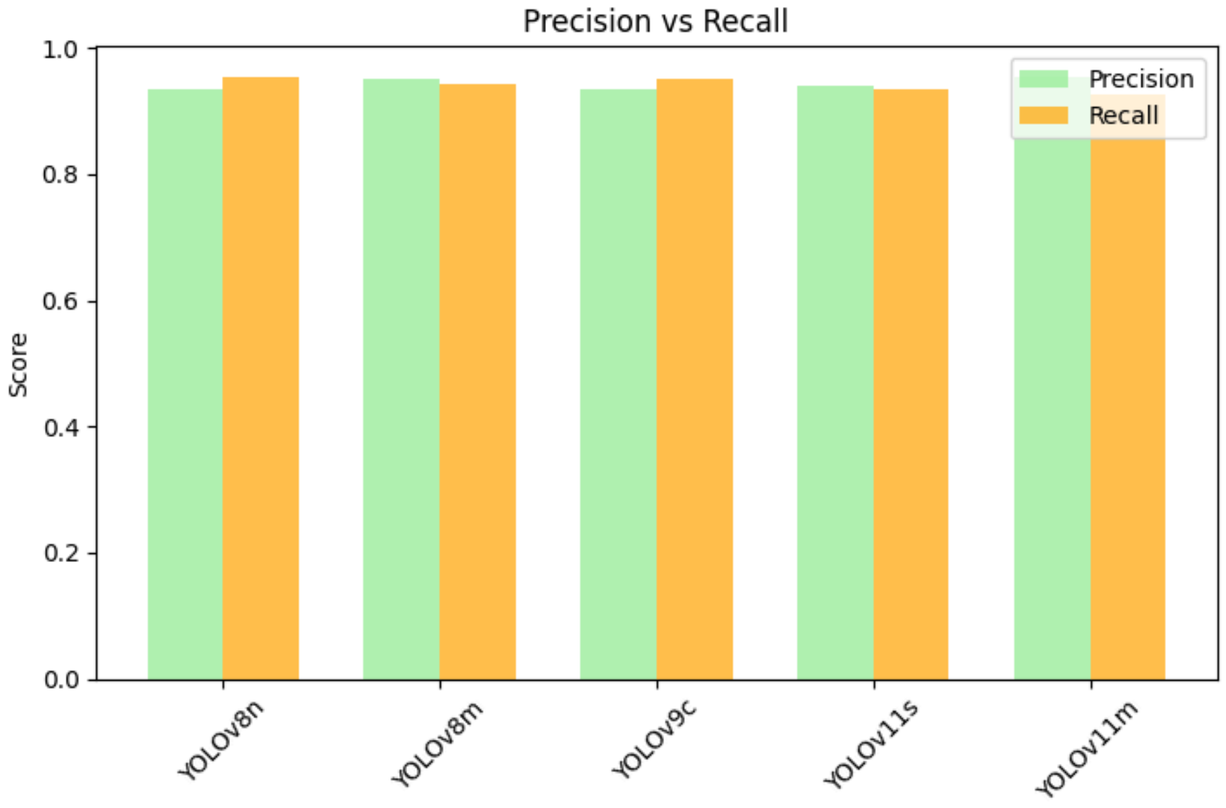
Model Comparison Results:

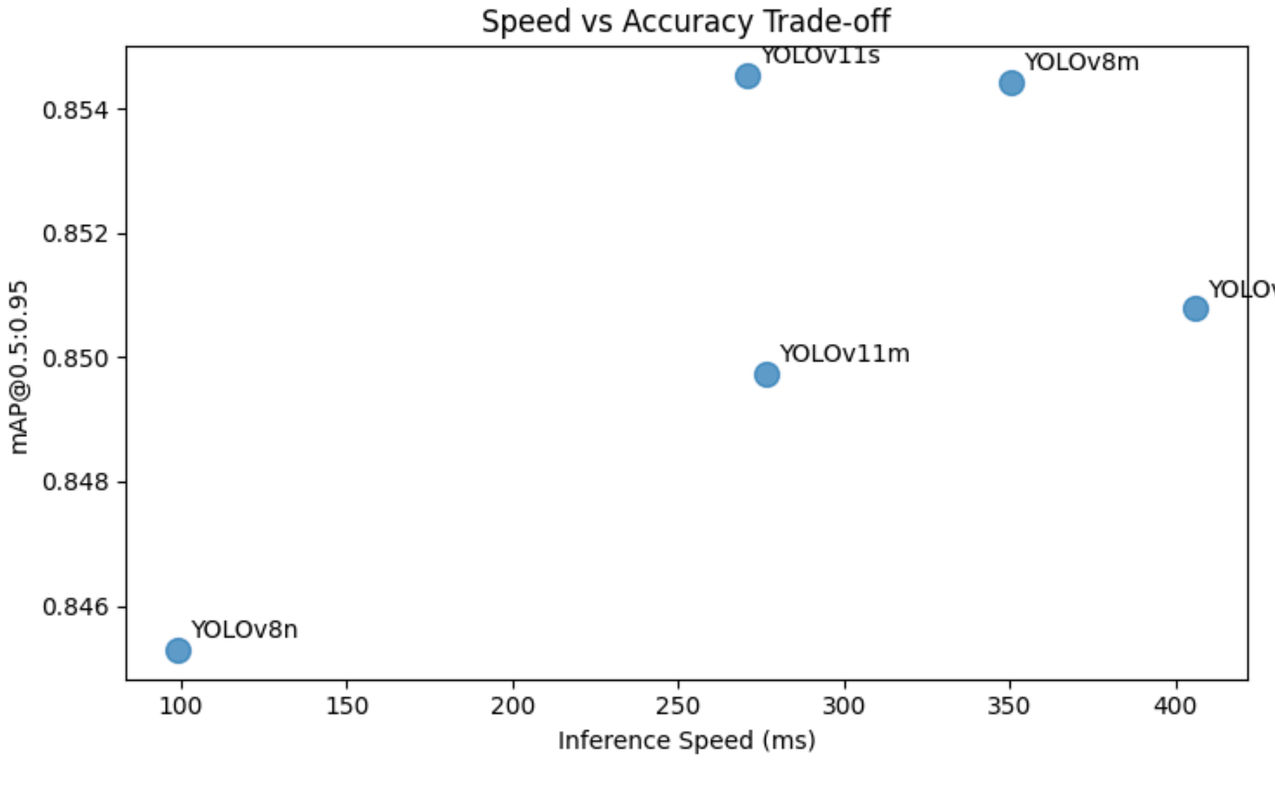
Based on validation results:

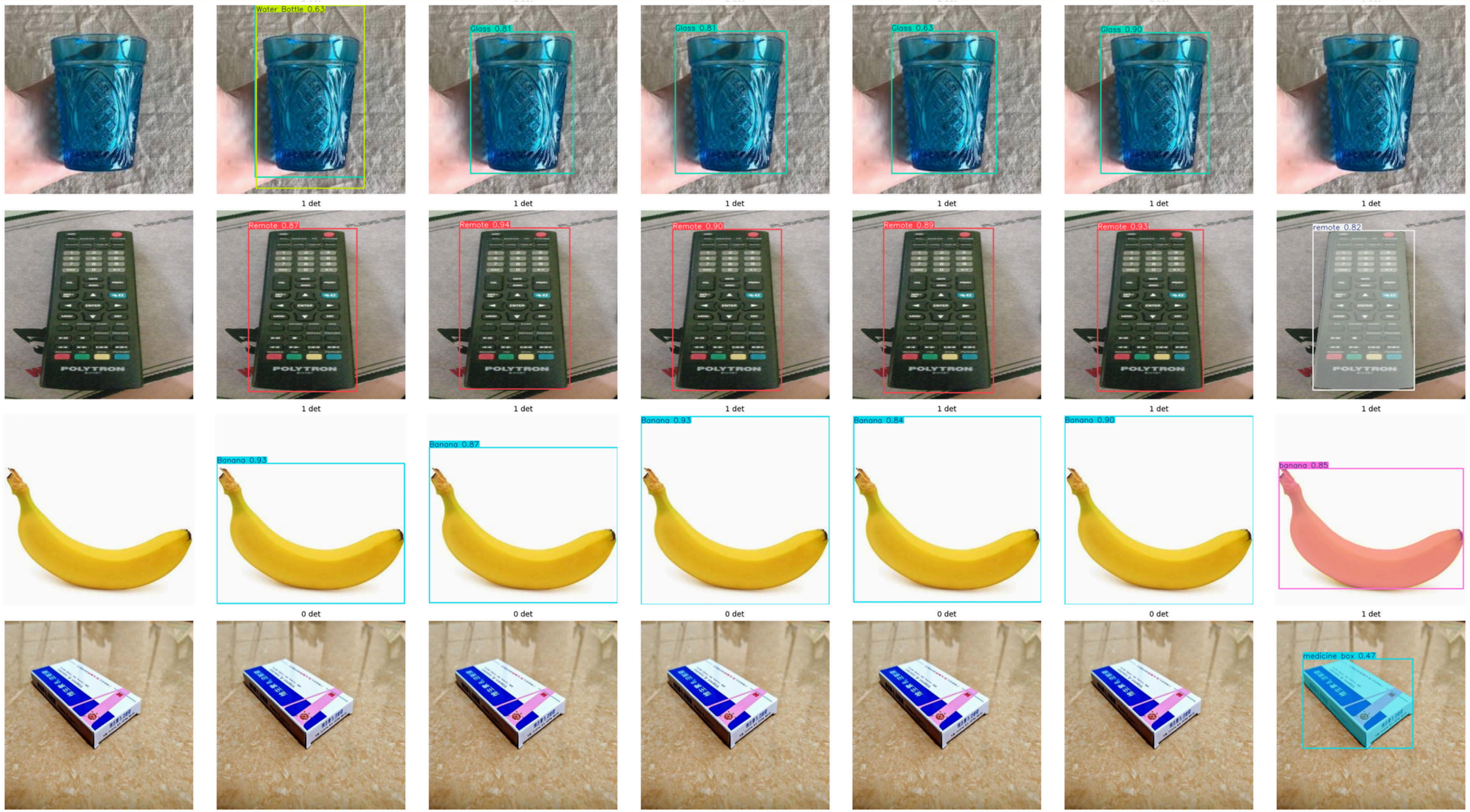


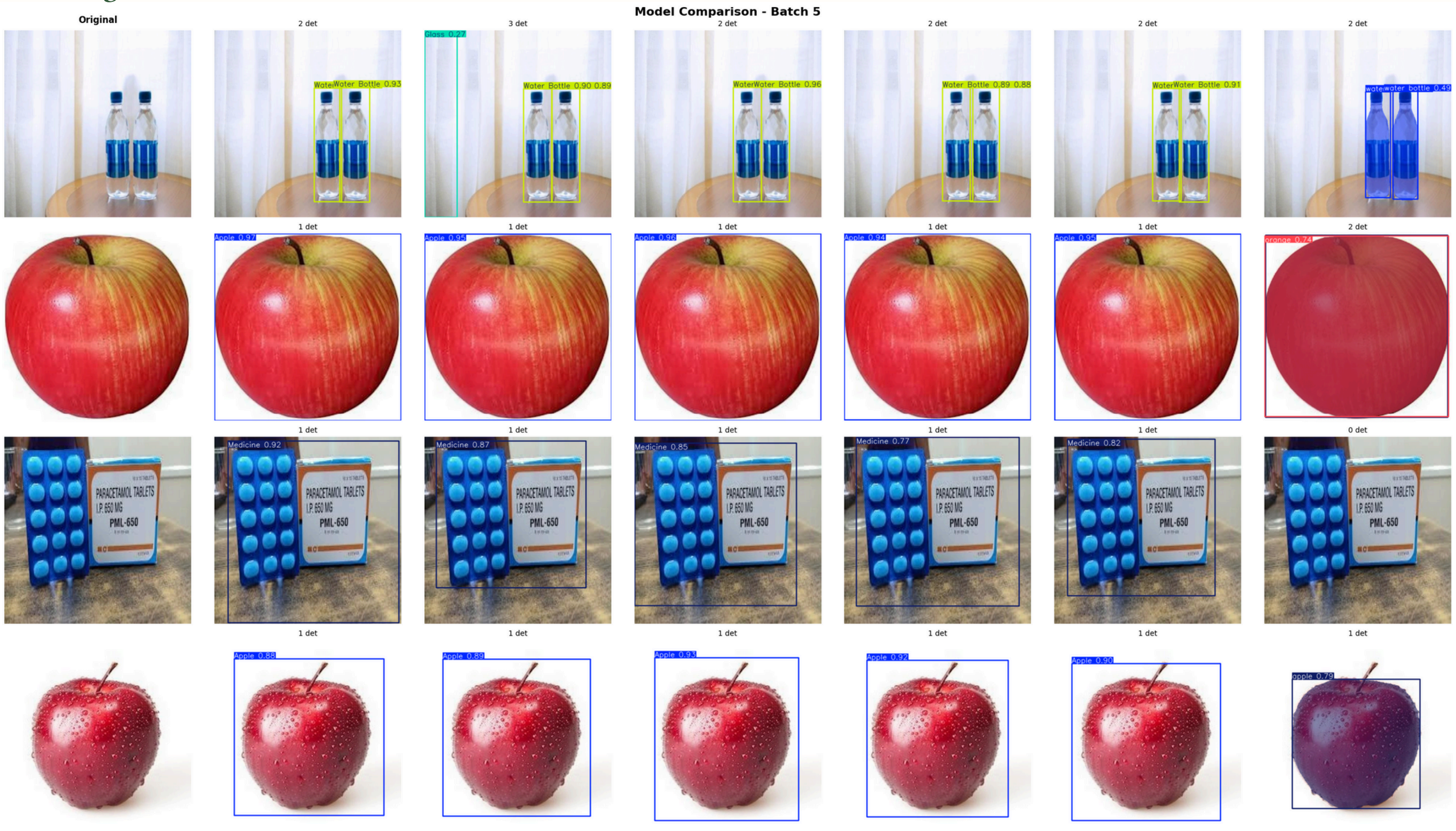


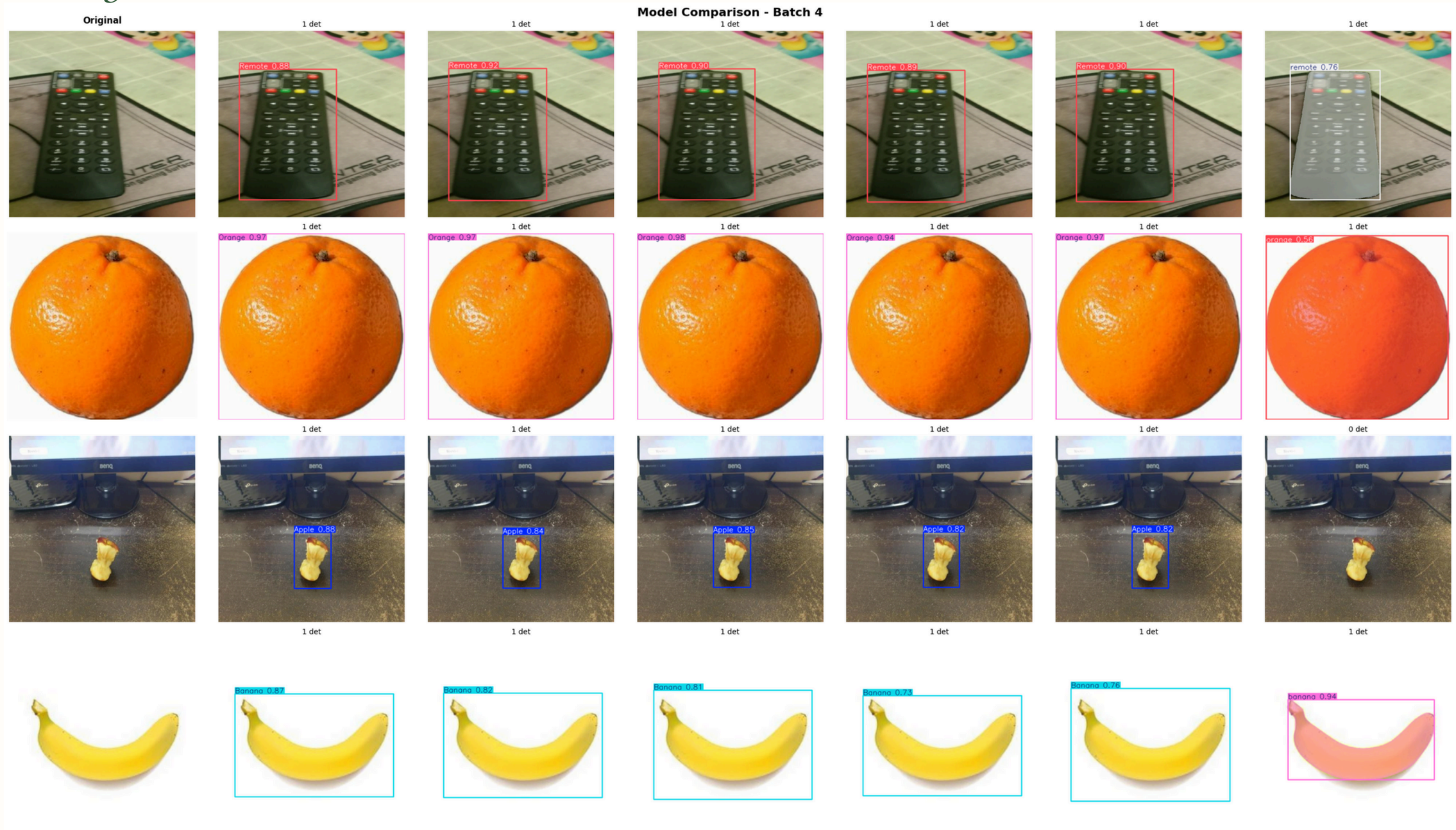
Model Comparison Results:

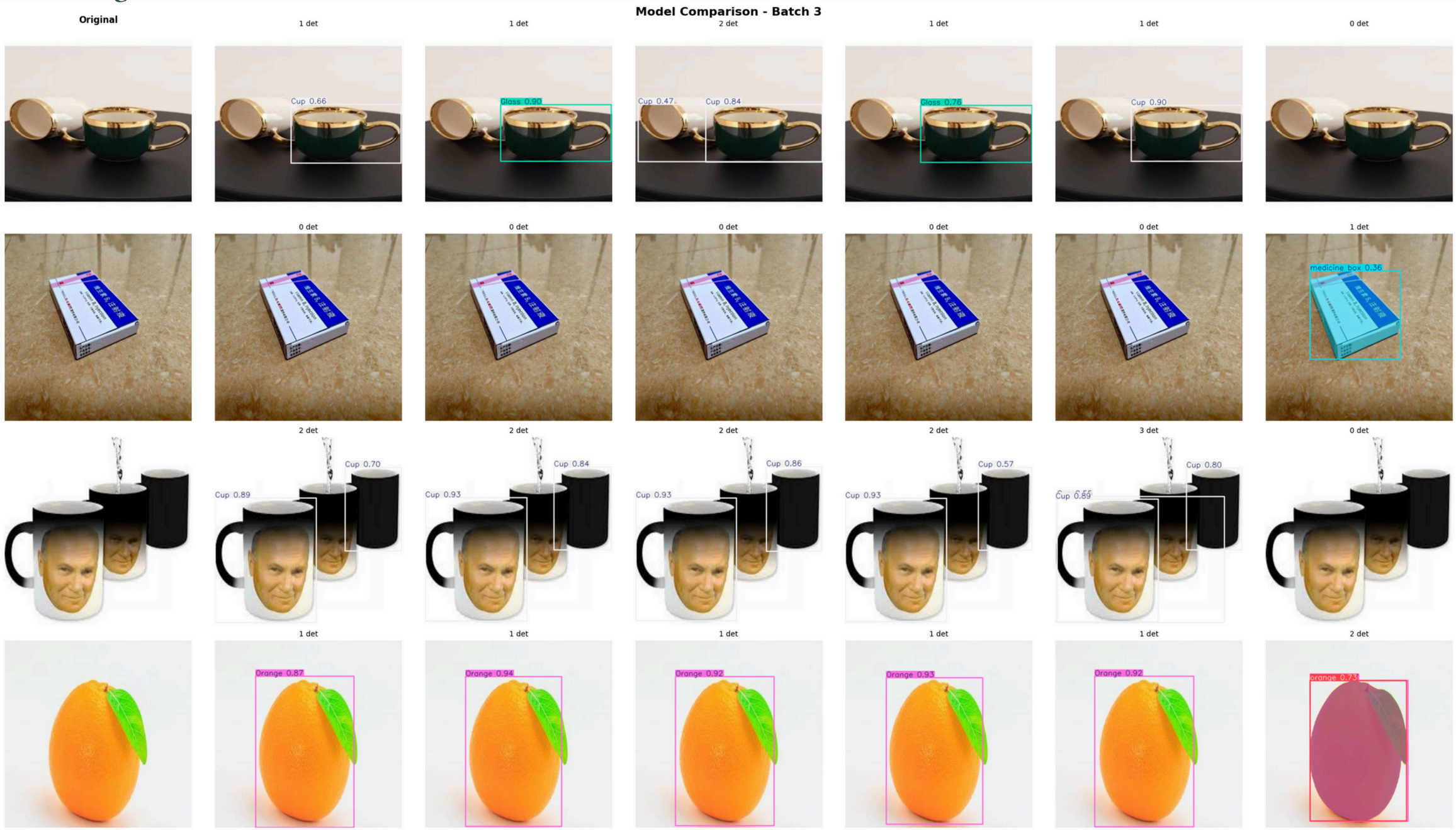


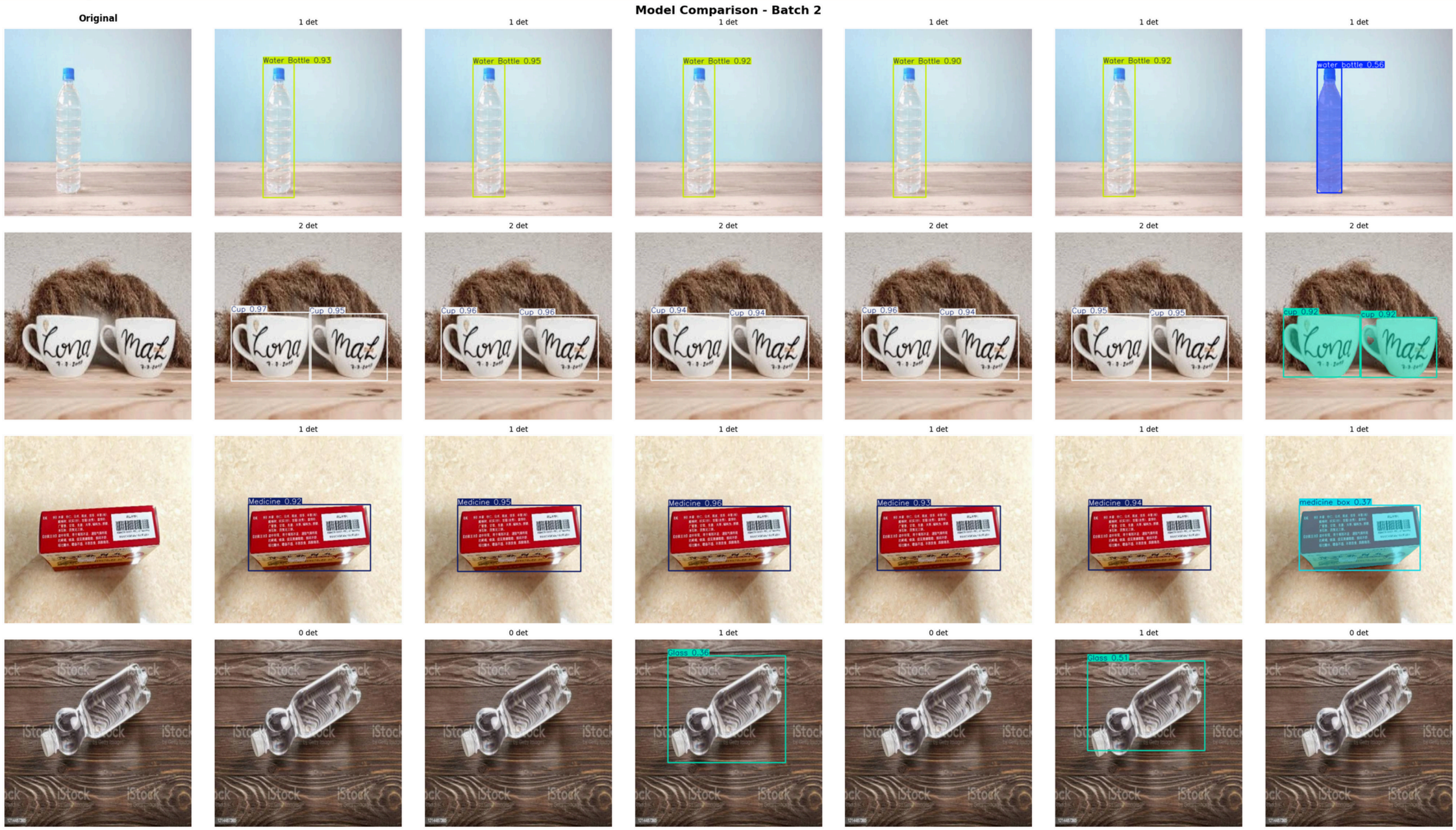



Original
YOLOv8n
YOLOv8m
YOLOv9c
YOLOv11s
YOLOv11m
YOLOe
Water_Bottle 0.63
Glass 0.81
Glass 0.81
Glass 0.63
Glass 0.90
1 det
Remote 0.87
Remote 0.94
Remote 0.90
Remote 0.89
Remote 0.93
remote 0.82
Banana 0.93
Banana 0.87
Banana 0.93
Banana 0.84
Banana 0.90
banana 0.85
0 det
medicine box 0.47

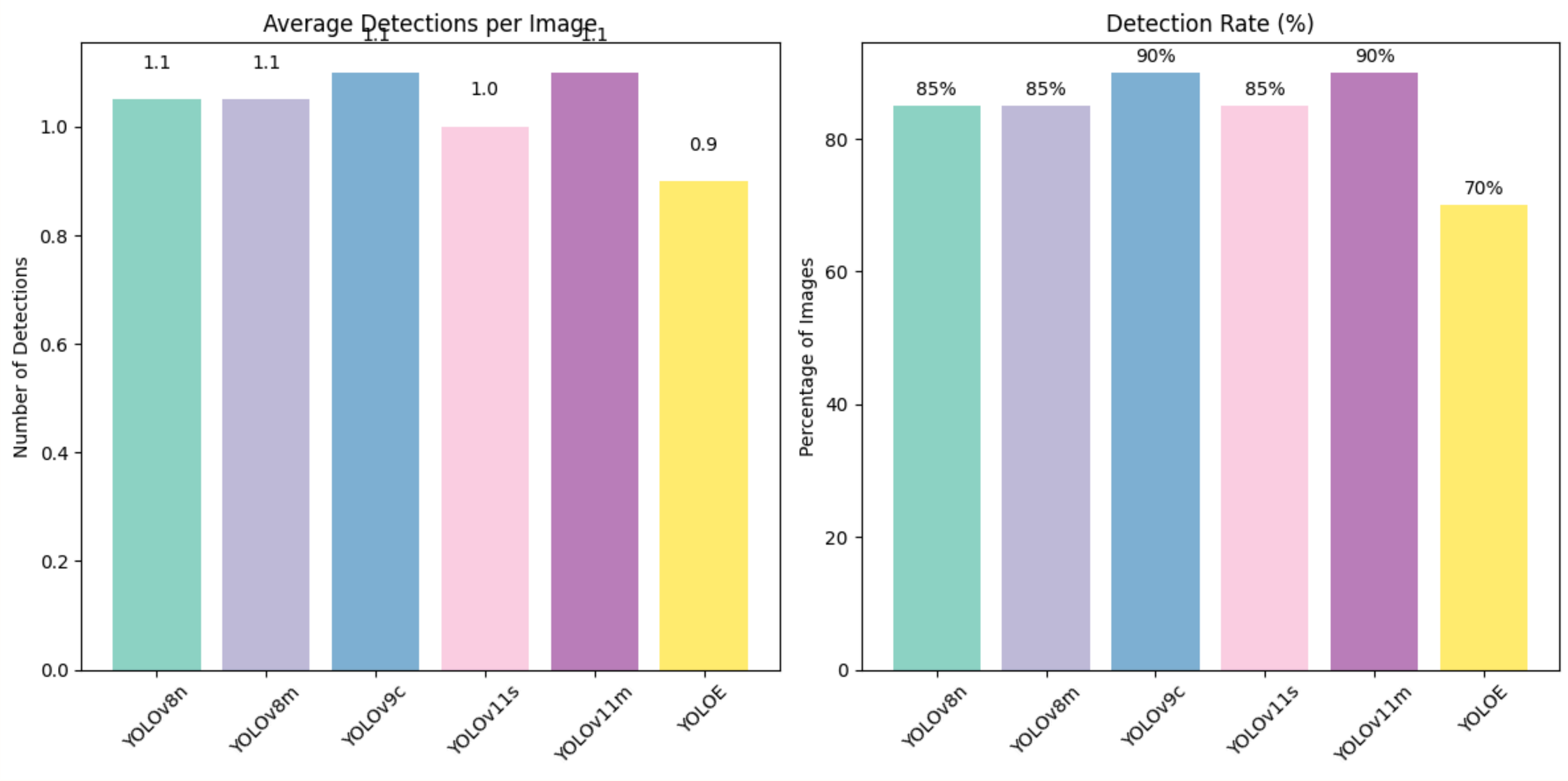

Original
YOLOv8n
YOLOv8m
YOLOv9c
YOLOv11s
YOLOv11m
YOLOe
Model Comparison - Batch 5
Original
2 det
Water Bottle 0.93
3 det
Glass 0.27
Water Bottle 0.90 0.89
2 det
Water Bottle 0.96
2 det
Water Bottle 0.89 0.88
2 det
Water Bottle 0.91
2 det
water bottle 0.49
1 det
Apple 0.97
1 det
Apple 0.95
1 det
Apple 0.96
1 det
Apple 0.94
1 det
Apple 0.95
2 det
orange 0.74
1 det
Medicine 0.92
1 det
Medicine 0.87
1 det
Medicine 0.85
1 det
Medicine 0.77
1 det
Medicine 0.82
0 det
PARACETAMOL TABLETS
I.P. 650 MG
PML-650
1 det
Apple 0.88
1 det
Apple 0.89
1 det
Apple 0.93
1 det
Apple 0.92
1 det
Apple 0.90
1 det
apple 0.79


Original
YOLOv8n
YOLOv8m
YOLOv9c
YOLOv11s
YOLOv11m
YOLOe
Model Comparison - Batch 4
Original
1 det
Remote 0.88
Remote 0.92
Remote 0.90
Remote 0.89
Remote 0.90
remote 0.76
Orange 0.97
Orange 0.97
Orange 0.98
Orange 0.94
Orange 0.97
orange 0.56
0 det
Apple 0.88
Apple 0.84
Apple 0.85
Apple 0.82
Apple 0.82
Banana 0.87
Banana 0.82
Banana 0.81
Banana 0.73
Banana 0.76
banana 0.94


Original
YOLOv8n
YOLOv8m
YOLOv9c
YOLOv11s
YOLOv11m
YOLOe
Model Comparison - Batch 3
Original
1 det
Cup 0.66
1 det
Glass 0.90
2 det
Cup 0.47
Cup 0.84
1 det
Glass 0.76
1 det
Cup 0.90
0 det
0 det
0 det
0 det
0 det
0 det
0 det
1 det
medicine box 0.36
2 det
Cup 0.89
Cup 0.70
2 det
Cup 0.93
Cup 0.84
2 det
Cup 0.93
Cup 0.86
2 det
Cup 0.93
Cup 0.57
3 det
Cup 0.89
Cup 0.80
0 det
1 det
1 det
Orange 0.87
1 det
Orange 0.94
1 det
Orange 0.92
1 det
Orange 0.93
1 det
Orange 0.92
2 det
orange 0.73


Original
YOLOv8n
YOLOv8m
YOLOv9c
YOLOv11s
YOLOv11m
YOLOe
Model Comparison - Batch 2
Original
1 det
Water Bottle 0.93
Water Bottle 0.95
Water Bottle 0.92
Water Bottle 0.90
Water Bottle 0.92
water bottle 0.56
2 det
Cup 0.97
Cup 0.95
Cup 0.96
Cup 0.96
Cup 0.94
Cup 0.94
Cup 0.96
Cup 0.94
Cup 0.95
Cup 0.95
cup 0.92
cup 0.92
1 det
Medicine 0.92
Medicine 0.95
Medicine 0.96
Medicine 0.93
Medicine 0.94
medicine box 0.37
0 det
1 det
Glass 0.36
0 det
1 det
Glass 0.51
0 det

Detection Results:



Conclusions:

- **Reference Paper [13] Object Detection using YOLOv8**
  - **Precision: 85%**
  - **Recall: 80%**
  - **mAP: 82%**
- Our Model Evaluation (YOLOv9c/YOLOv11m) -Based on validation metrics and visual testing on our custom 8-class household dataset, YOLOv9c/11m achieved:

YOLOv9c

- **Precision: 0.933 (93.3%)**
- **Recall: 0.952 (95.2%)**
- **mAP50: 0.962 (96.2%)**
- **mAP50–95: 0.851 (85.1%)**

YOLOv11m

- **Precision: 0.952 (95.2%)**
- **Recall: 0.925 (92.5%)**
- **mAP50: 0.962 (96.2%)**
- **mAP50–95: 0.85 (85%)**

- These results outperform the reference YOLOv8 metrics in both precision and recall, indicating YOLOv9c/11m are better at both correctly identifying objects and detecting all instances present in the images.

Thank you!