StockSense: A Real-Time, AI-Powered Inventory Management System for SMEs Using Pretrained YOLOv8n and Streamlit.

MSc Research Project
Msc AI for Business

Dana Janeth Carranza Romero
Student ID: 23374900

School of Computing
National College of Ireland

Supervisor: Muslim Jameel

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Dana Janeth Carranza Romero
Student ID:	23374900
Programme:	Artificial Intelligence for Business **Year:** 2024-2025
Module:	MSCAIBUS
Supervisor:	Muslim Jameel
Submission Due Date:	11 August 2025
Project Title:	StockSense: A Real-Time, AI-Powered Inventory Management System for SMEs Using YOLOv8n and Streamlit.
Word Count:	4701 **Page Count** 17

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……………………………………

Date: 11 August 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

StockSense: A Real-Time, AI-Powered Inventory Management System for SMEs Using Pretrained YOLOv8n and Streamlit.

Dana Janeth Carranza
23374900

Abstract

Deployment of AI computer vision tool to detect inventory in limited technical and budget resources small businesses or SMEs. Implementing YOLO8n as the selected object detection model due to its lightweight and compatibility. As well as Camo Cam for real time detection with a smartphone. The project detects live inventory data each 5 seconds to provide information of the stock available and provide insights of the inventory. Later Streamlit provides a user-friendly interface for analyzing product counting, dashboards of stock composition, and provides recommendations generated by an LLM. Also, a custom trained YOLOv8n was tested but a pretrained model was the best option in the case for accessibility and low-cost solution for SMEs.

1 Introduction

Technological improvements and globalization, upgrades in supply chain management have become a fundamental base for the success of companies. Inventory management occupies a central place, essential to ensure product availability, minimize costs, and meet the delivery expectations of the final consumer or product (Zotov, V.B., Demin, S.S. and Glazkova, I.Y. 2019). The biggest challenge for small and medium companies and even small businesses is the lack of technical, time, efficiency resources. SMEs always seek opportunities to reduce costs and time. They often use manual processes where human error is unpredictable. Technical specialized software and employees to track inventory is not often available in cases where the SMEs centralize the resources in other areas.

The aim of StockSense is to provide an accessible solution for SMEs to detect inventory in real time as well as begin a tool easy to install and use by having the available resources to implement it. The system is composed of a computer vision model (YOLOv8n) which utilizes the camera of the smartphone (in this case iPhone) for image ingestion using the Camo Cam app available in the Apple Store for free. The real time recording with the smartphone records and provides the image ingestion for Streamlit to analyze the data and provide a translation of the products detected and provide a LLM suggestion depending on the needs of the SMEs which will be indicated with a prompt by the user on Streamlit. The insight will depend entirely on the prompt of the user, it can provide restocking suggestions, stock level recommendations, and decision-making support for SMEs KPIs. Also, a section with the image capture will be displayed as well as having the counting of the objects detected. In the last section it gives the option to download a PDF integrating the LLM recommendations, dashboards with the classification of the products providing the stock levels, composition, and last detection frame.

This project also analysed the possibility of using a custom trained model with the dataset SKU11K which later was discarded due to the inaccuracy, time consumption, and technical specialization resources which are not often available in SMEs. The research question is: Do pretrained YOLOv8n model offer a practical and accurate solution over a custom model for real-time inventory management and AI decision support for SMEs? This question is later explained and answered on a comparative analysis between a pretrained model and a custom trained model on the dataset SKU110K. Which accuracy wasn't eligible for a project, and it required a considerable use of resources like time and specialization technical skills which are limited on SMEs.

StockSense

2 Related Work

Manual inventory management has caused major costs due to common errors in the counting of objects. It is a labour intensive and time-consuming process which requires continuous monitoring of appropriate real-time monitoring in inventory (Verma, N.K., Sharma, T., Rajurkar, S.D. and Salour, A., 2016.) Despite the evolution of technologies, inventory management still demands a physical count or scan to update the internal system. For example, in the last decade over 80% of warehouses in Western Europe used the traditional "picker to-to-parts" method. This model required workers to physically count or pick one item at the time, which is not efficient for scalable business strategies. Nowadays, AI algorithms can analyse real-time insights into inventory levels and trends. This can benefit businesses to adapt swiftly to dynamic market conditions (Groenewald, E. and Kilag, O.K., 2024). For example, CNN-based object detection, such as the YOLO model, can deliver both high speed and accuracy, which make them very suitable for those kinds of dynamic environments where extremely fast decisions are essential. If we manage to deploy a YOLO-based system efficiency, the system will be able to continuously count the products, update the inventory records, and synchronize the digital stock counts with the actual stock on the inventory. Regular manual systems are widely practiced through the use of human power moving within the aisles to manually tally stocks and eventually, it becomes the only way they can make the internal log of the company meet with the one online though there might be the trouble in the form of gaining a much-needed vantage point on the movement of those stocks (Patil, H. and Divekar, B.R. 2014). inventory systems often tend to be fully manual, and the staff move away the items physically till the expiry of the items. Inventory management is essential for the informatization systems to improve the process efficiency (Wang, L.-p. 2009).

Limited resources have created an opportunity for Artificial Intelligence to solve problems and challenges of the traditional inventory management methods, which have been found to be ineffective. Today, companies that want to keep up with technological development and globalization must be able to effectively manage their supply chains to achieve high quality, increased efficiency, and low costs. Diversified customer needs, global competitors, and market competition have led companies to pay more attention to inventory management. (Albayrak Ünal, Ö., Erkayman, B. and Usanmaz, B 2023). AI technologies can automate the inventory management process in different ways such as stock monitoring, data improvement, and stock keeping so SMEs can speedily respond to any demand issues that arise. Management and tracking of stock inventory are a vital part of a company, laboratory, operation theater, hospitals, machine warehouse, storerooms, etc. Several researchers have carried CNN work in this area which includes systematic inventory management. (J. P. N., Prashanth, S., D., A. and J., M. 2023).

2.1 Applications of Pretrained Object Detection Models in Inventory Systems

The limitations on SMEs impact on the resource's allocation, which lead enterprises to seek cost-efficiency solutions. Customizing a model may take time and technical resources that SMEs often are limited to spare, also this solution is for larger enterprises that can invest on the technical specializations. Pretrained models provide benefits on time management and trained personnel to deploy it, as well as preventing having to maintain or update custom models. For example, a project was deployed as a lightweight model using YOLOv8n where a mAP50 score of 0.933 and an inference time of 13.4 ms was achieved by Arrie Kurniawardhani et al. (2024). This demonstrates that a pretrained model can achieve high accuracy as well as respecting the low-cost goal. The You Only Look Once (YOLO) object detection algorithm has gained popularity thanks to its combination of processing speed and accuracy, making it an attractive choice for real time object detection in urban environments (Ikmel, G. and Najiba, E.A.E.I. 2024). Also, a computer vision model was trained to demonstrate that inventory tracking can effectively work on minimal setup, which is great for small businesses and SMEs according to Villegas-Ch et al. (2024). Model capability is demonstrated in (Mupparaju, Thotakura & Ch. Venkata Rami Reddy, 2024) where an extensive review on YOLOv8 and nano variants were compared to other Yolo versions like v4, v5, v7 and other alternatives models like SSD and Faster R-CNN. It was implied that the YOLOv8n was ideal for real-time detection projects since it requires low power hardware such as smartphones and conventional computers. Also, it was highlighted that YOLOv8 offered better accuracy and speed than other models. It also confirmed that the pretrained already performed other versions like YOLOv5 in common scenarios. Counting products will be a light and smooth task for YOLOv8n. Limitations in other pretrained models can be found, for example CLIP and DINOv2 with zero-shot classification fail to detect subtle variations. Tur et al. (2024) proposed an ensembled method where CLIP and DINOv2 were combined to improve performance, but it was demonstrated that the model still lacked precision for real-world scenarios. It's implied that the project must have an extensive and emblematic dataset to custom a model to classify items accurately. The study suggested that pretrained models were ideal if it is a coarse detection like a product count rather than item differentiation.

Authors Ahmad, Rahimi and Hayat (2024) used a pretrained YOLOv8 model which is fine tuned from COCO dataset to detect chairs and workers in the manufacturing environment. The idea was to identify bottlenecks in real case scenarios. The project was tested over 6 months, and it was implied that the model was ideal for the project due to its properties of real time detection, also it supports the idea of significantly reducing time and resources by deploying a pretrained model for a low-cost application suitable for small business or SMEs.

2.2 Model Selection

In this section a comparison between YOLO models will be illustrated, as well as providing insights between pretrained and custom models to differentiate the benefits and allow an ideal deployment for each project needs. In the following chart, it can be implied that the YOLOv8 version provides a high speed-accuracy balance in low latency with 2.3ms and 4.2ms and a high mAP of 46.8 and 50.5%. Making it suitable for real-time applications. The nano version which is not provided in the chart is designed with the same benefits but lighter and suitable for deployment in conventional hardware devices such as smartphones and laptops. Below an image of the YOLO versions comparison is provided, the information was subtracted from the Ultralytics official website.

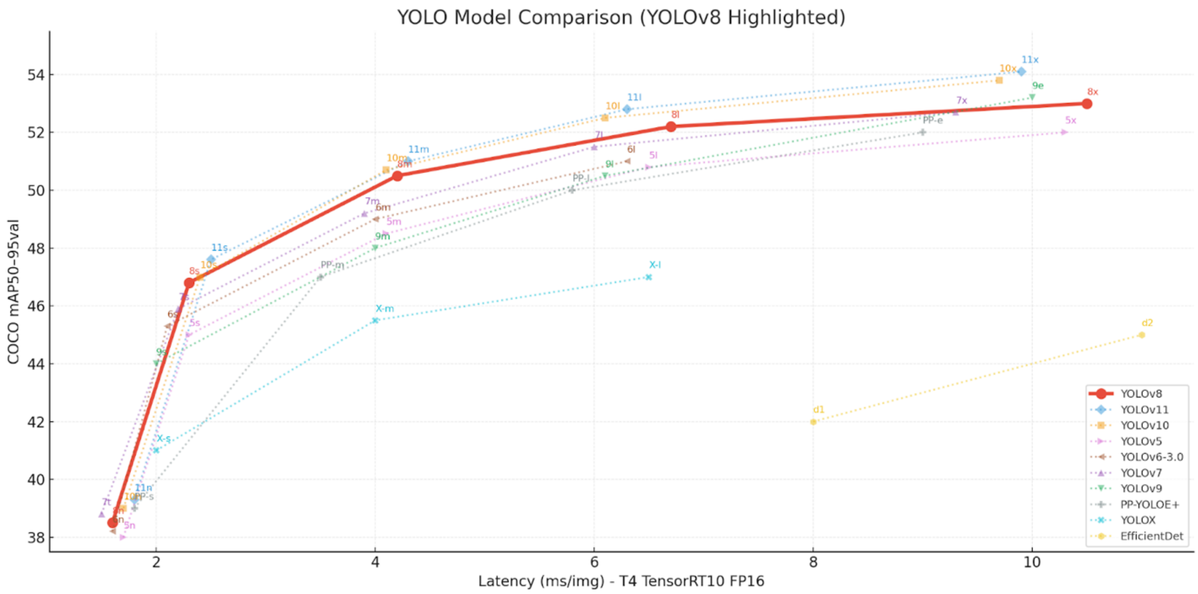


Figure 1: YOLO Model Comparison.

Below a comparative table is displayed to differentiate the benefits or disadvantages between previous use cases of similar projects where different model types, use cases, accuracy metrics, system set up and key limitations are discussed.

Table 1. Related Work study comparisons.

Study	Model Type	Use Case	Setup	Accuracy / mAP	Key Limitation
Kurniawardhani et al. (2024)	Pretrained YOLOv8n	On-shelf product detection	Edge device deployment	mAP50 = 0.933, Inference = 13.4 ms	Best in clean, structured retail environments
Villegas-Ch et al. (2024)	Pretrained CNN YOLO	Inventory tracking in warehouses	Minimal setup with pretrained weights	Qualitatively effective in practice	Lower performance in low-light or cluttered setups
Mupparaju et al. (2024)	YOLOv8 vs YOLOv5/SSD	General object detection benchmarks	Off-the-shelf YOLOv8 models	YOLOv8 outperformed all prior YOLO models	Not tailored for SKU-level tasks
Tur et al. (2024)	Zero-shot CLIP + DINOv2	Fine-grained retail product differentiation	Zero-shot; no training	Low accuracy (<50%top - 1)	Fails in detecting subtle SKU variations
Ahmad et al. (2024)	YOLOv8 (Transfer Learning)	Worker/chair detection in manufacturing	Pretrained on COCO + small dataset	~89% detection accuracy	Detection reduced with occlusion
Ikmel & Najiba (2024)	YOLO (General)	Real-time object detection in urban/logistic environments	Standard pretrained YOLO	High speed and accuracy noted	Lacks task-specific metrics for inventory
Groenewald & Kilag (2024)	AI in inventory decision-making	AI-based inventory trend forecasting	AI model inference layer	Not model-specific (qualitative)	Focuses on trend insights, not detection

AI and computer vision play a fundamental role in continuing to increase the benefits in logistics and inventory management. Inventory object detection with YOLOv8n facilitates locating and recognising items inside the warehouse of the inventory using deep learning and computer vision techniques (RA, V.S.K.,Kathrine, G.J.W. and Salaja, S. 2024). Small businesses or SMEs can benefit from pretrained models to develop solutions for their needs. It can be implied that applying computer vision with YOLOv8n is ideal for low-cost projects that are often required in SMEs. This version has shown an effective performance in low-cost devices compared to other versions, as well as confirming that a pretrained model works ideally for low cost projects and provides high accuracy detection. However, it was discussed that even though pretrained modes allow simple and fast deployment and lower technical resources, models can struggle with fine-grained object detection. It can be implied that for most cases a pretrained object detection model like YOLOv8n that is pretrained with COCO dataset can be ideal for real time counting detection for the following objects: person, bicycle, car, motorcycle, airplane, bus, train, truck, boat, traffic light, fire hydrant, stop sign, parking meter, bench, bird, cat, dog, horse, sheep, cow, elephant, bear, zebra, giraffe, backpack, umbrella, handbag, tie, suitcase, frisbee, skis, snowboard, sports ball, kite, baseball bat, baseball glove, skateboard, surfboard, tennis racket, bottle, wine glass, cup, fork, knife, spoon, bowl, banana, apple, sandwich, orange, broccoli, carrot, hot dog, pizza, donut, cake, chair, couch, potted plant, bed, dining table, toilet, TV, laptop, mouse, remote, keyboard, cell phone, microwave, oven, toaster, sink, refrigerator, book, clock, vase, scissors, teddy bear, hair drier, toothbrush. If the small business or SMEs has any of these objects or the need to count persons, this model is ideal for a cost-effective solution.

3 Research Methodology

This methodology aims to develop an accessible AI computer vision tool using conventional devices that are available in small business or SMEs. The methodology research is split in two sections since two model alternatives were tested as shown below.

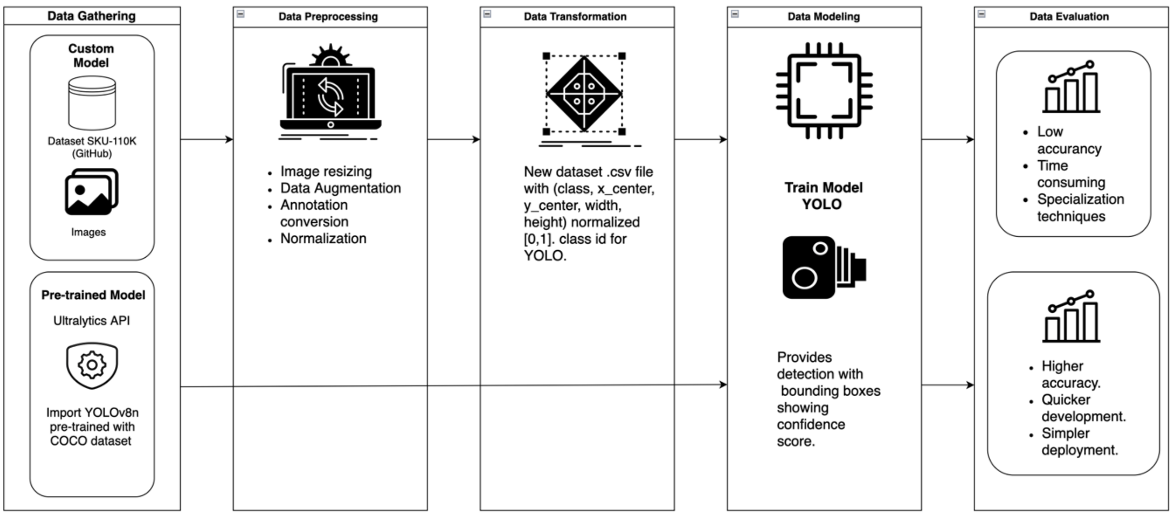


Figure 2. Model Pipeline.

Custom YOLO Model:

Data Gathering: Dataset SKU-110K was used to train the model with images in inventory environments.

Data Preprocessing:

- Resize Images for YOLO input.
- Convert annotations (class x_center, y_center, class id).
- Normalization: [0,1].

Data Transformation: New dataset created .csv file with the data preprocessing outputs (image_name, x1, y1, x2, y2, class,image_width, image_height columns) and (images/train), val and test, data.yaml.

Modeling: Model was trained on 100 epochs (72hrs training). Provides image detection output with bounding box showing confidence score.

Evaluation: The model can detect a product with .34 accuracy, but when it comes to more objects it triggers due to the many detections deployed.

Pre-trained YOLOv8n Model:

- **Data Gathering:** Import pretrained YOLOv8n with COCO dataset model from Ultralytics API. This allows to skip the data preprocessing, data transformation. The data goes directly to the model.
- **Data Modelling:** Data goes directly to data modeling creating the confidence bounding boxes score in the trained images.
- **Evaluation:** Provides higher accuracy, great for quick development and simpler deployment.

4 Design Specification

StockSense architecture is designed for proving real time detection and deploy a Streamlit interface for providing the the final product to the user where it can deploy the real time frame, counts of the products, LLM recommendations and dashboards in localhost. Five different layers were created: Data Source, Detection System, Data Logging, AI Recommendations and User Interface.

Data Source

The system begins by taking the data from the smartphone in this case iPhone camera that's connected with Camo Camera to later transfer the real time video detection to the computer in this case MacBook Air that also has Camo Camera which acts as a virtual webcam.

Detection System

The detection process begins with the python code importing YOLOv8n pre trained model with COCO dataset from Ultralytics along with other libraries to deploy the real time detection.

Data Logging

Detection results are log in a CSV log “inventory_history.csv”. The entries include a timestamp, SKU count, and other metrics.

LLM Recommendations

Groq API is integrated to the system to provide the inventory insights, it generates LLM recommendations based on the user’s prompt based on their business KPIs and metrics.

User Interface (Streamlit)

Python code deploys an interface deployed in Streamlit in which first section user puts a custom prompt to deploy the LLM recommendations, second section provide the real time detection along with bounding boxes in each object detected providing a confidence score. Last section includes a bottom in which a PDF document is downloaded providing dashboards with current stock counting, stock composition and last frame detection.

Deployment

The system (Streamlit) is deployed in Local Host.

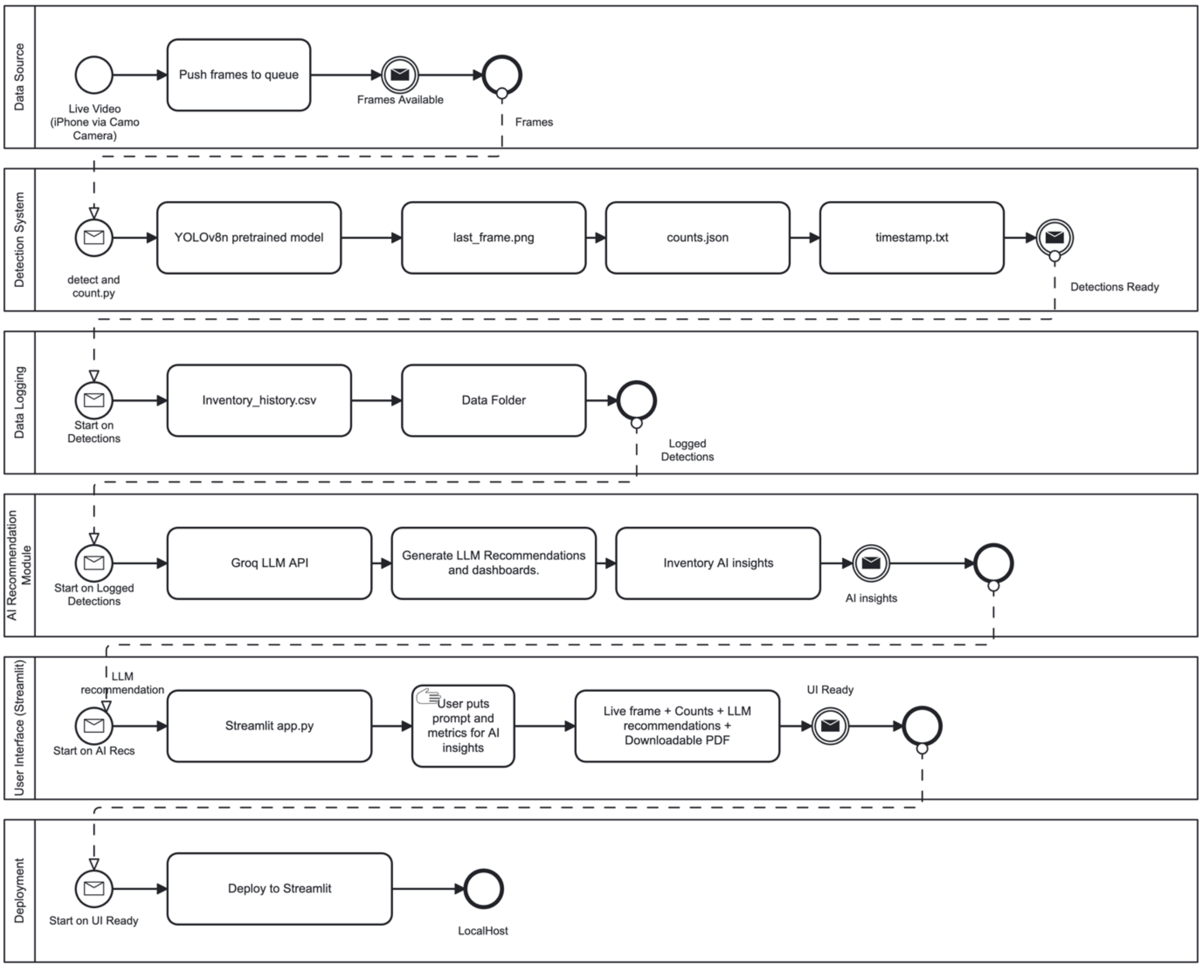


Figure 3: Model Design.

5 Implementation

The objective of this project was to provide comparative research and provide a demonstration on why a pretrained YOLO model is the best option for deploying solutions for small businesses and SMEs. The system was designed to:

- Run locally on conventional devices (in this case Macbook Air and iPhone 16 pro). Use a smartphone camera for real time detection using Camo Cam app available in IOS for free. Count and detect the objects (COCO dataset objects)
- Deploy the outputs in Streamlit as an interface to display the information insights on the objects detected.
- AI recommendations provided by Groq API with a custom prompt created by the user. Last frame detection with bounding boxes showing the confidence score as well as labelling the items and counting them in Streamlit.

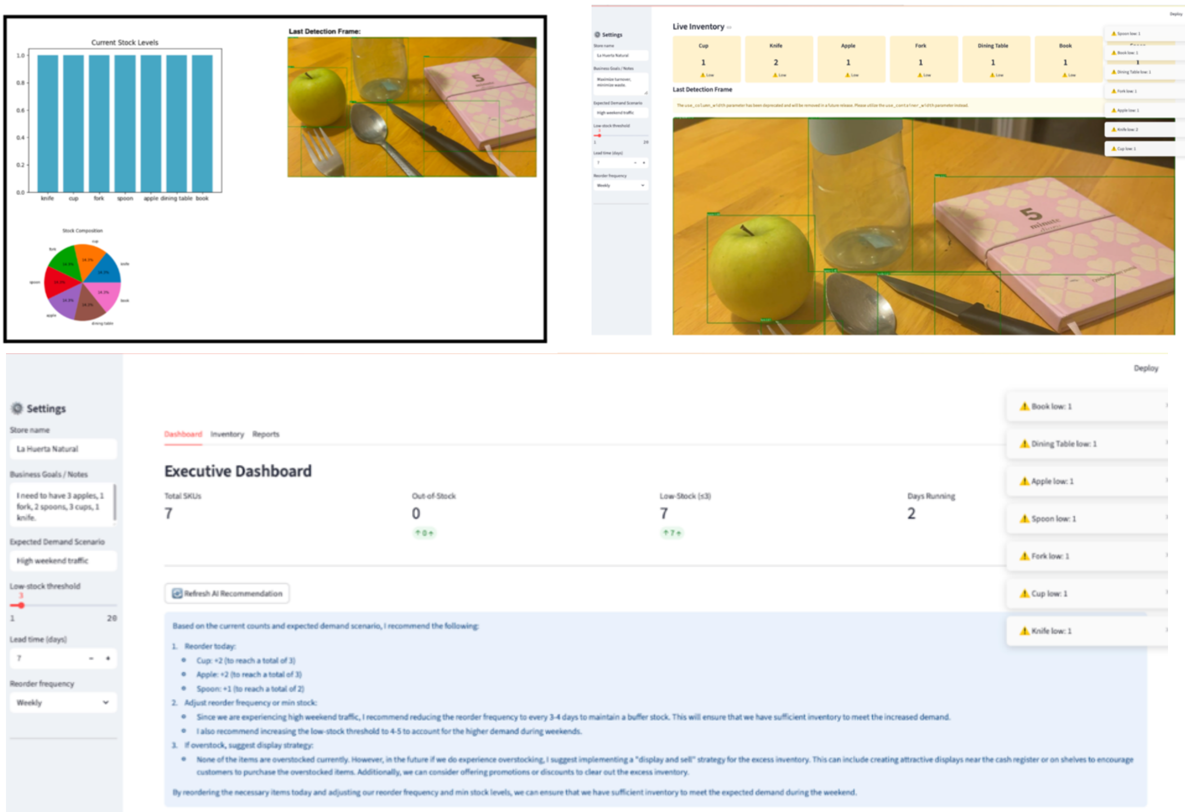


Figure 4: Model Interface.

6 Evaluation

A comparative evaluation was applied between the custom trained model (best.pt) and the pretrained model YOLOv8n by creating a dataset of 30 images to compare the MAE, MAPE, Avg inference and FPS of each model. In quick_eval.py the results demonstrated that the pretrained model YOLOv8n achieved a MAE of 1.17 which is slightly lower compared to the

1.33 achieved from the custom model. It can be implied that the pretrained model produced fewer absolute counting errors in the test. However, custom model archived a higher performance over the pretrained model in metrics MAPE by performing a 63.9% over the 72.5% produced by the pretrained model. It is inferred that the custom model had higher accuracy in scenarios with smaller object counts. This is also can be caused by the small test dataset training.

Inference speed was dominated by the pretrained model YOLOv8n which demonstrated a much higher performance over the custom model by deploying an average inference of 44.9 and 22.3 FPS over the 54.2 ms and 18.4 FPS of the custom model. These results align with literature findings of Ahmad, Rahimi and Hayat (2024) which stated that YOLOv8n has a lightweight architecture and is effective for real-time detection which is ideal for small businesses and SMEs with limited computational resources.

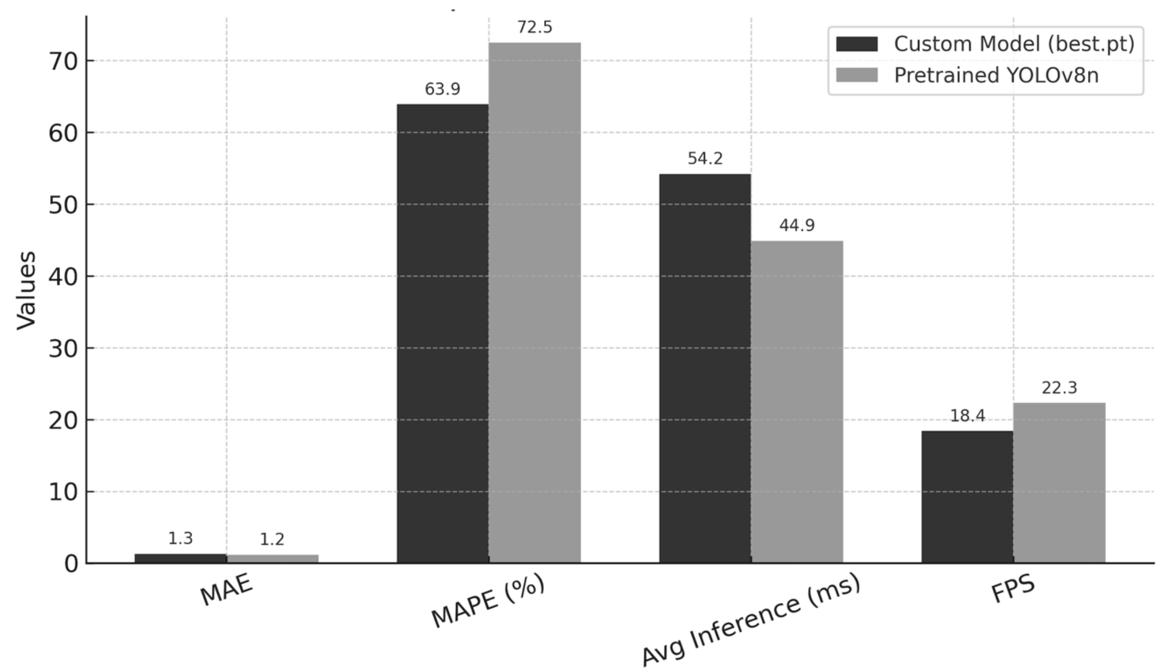
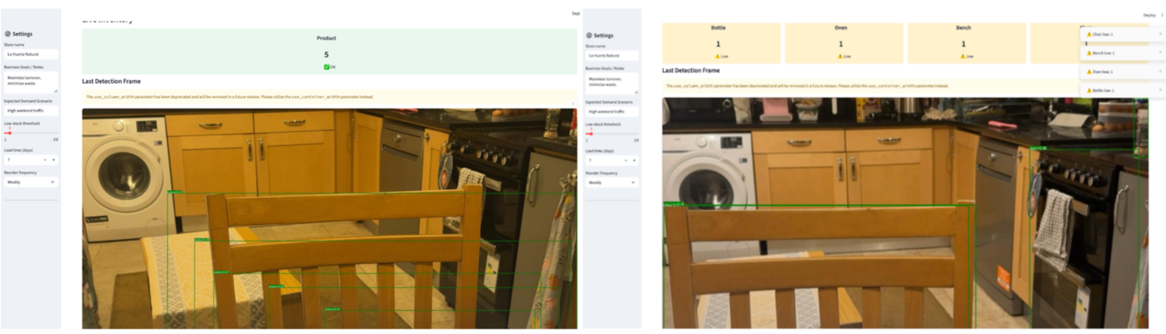
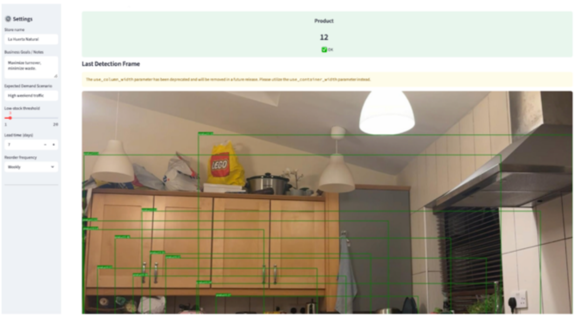


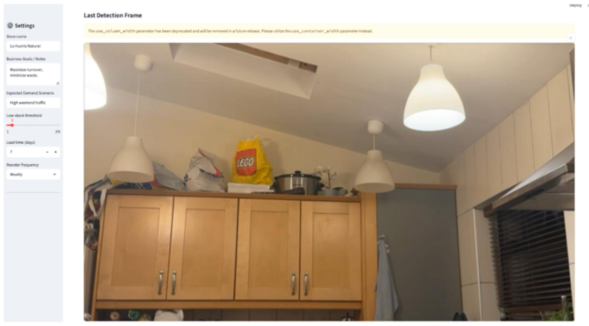
Figure 5: Model Performance Comparison

6.1 Experiment 1

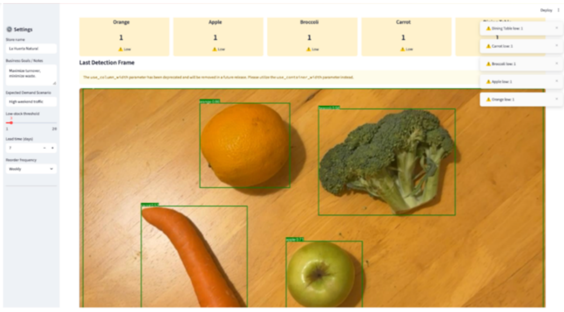


6.2 Experiment 2





6.3 Experiment 3



6.4 Discussion

In the experiments the two alternatives (custom and pretrained) models were for analysing the best of option for limited SMEs. Left image demonstrates custom model and right image the pretrained coco dataset model. 3 experiments were tested in which different positions to analyse the highest bounding box confidence score, objects detected and category accuracy. Also, it provided higher confidence score in its detections.

Experiment 1

Table 2: Experiment 1 Evaluation.

Evaluation	Custom Model	Pretrained Model
Bounding box confidence score	Highest confidence score was 41, detecting the chair.	Highest confidence score was 40 detecting the oven which is correct.
Objects Detected	5 objects were wrongly detected.	4 objects were detected
Category accuracy	Only 1 category was detected because the model was trained on just 1 category.	4 categories were detected chair, over, bottler and bench. Only the bench was wrong (it was detecting the chair).

Experiment 2

Table 3: Experiment 2 Evaluation.

Evaluation	Custom Model	Pretrained Model
Bounding box confidence score	54 showing as the highest confidence score.	No detection.
Objects Detected	12 objects were "detected"	No object was detected since pretrained model was trained on COCO dataset which doesn't include anything showed in the image.
Category accuracy	Only "product" was deployed due to model training.	No category detected.

Experiment 3

Table 4: Experiment 3 Evaluation.

Evaluation	Custom Model	Pretrained Model
Bounding box confidence score	No detection.	Highest confidence score showing 86 for orange which is correct.
Objects Detected	0	4 objects were detected with 100% accuracy.
Category accuracy	0	4 categories were detected (apple, broccoli, carrot and orange) were detected with 100% accuracy. COCO dataset includes these objects.

In conclusion, the deployment metrics showed that custom model can provide a slightly more proportional performance between accuracy and operational speed. Even though it may be inferred that the custom model provides a preferred high value over small inventory counting, pretrained YOLOv8n model demonstrates a higher performance for real-time monitoring with low latency and ease of integration, also in the findings that experiments provide, it is implied that the pretrained model have better accuracy for detecting and classifying objects since the custom model only counts one category which is "product" whereas the pretrained model can detect of 80 different products accurately. Since the project objective is to deliver an accessible and low-cost computational integration management tool for small business or

SMEs, it was demonstrated that pretrained with COCO dataset YOLOv8n was the best option for the final implementation for StockSense.

7 Conclusion and Future Work

In this project two real-time object detection model alternatives were tested with the objective to create an accessible AI tool for small business and SMEs. Custom YOLO model and pre trained with COCO dataset were explored. The model that was created and customized with dataset SKU-110K was trained with retail scenarios with a preprocessing pipeline to improve accuracy. Pre-trained model was imported with COCO dataset detection implementation to detect 80 different objects. It was demonstrated that pre-trained YOLO model (YOLOv8n) was a better option to deploy a solution for small business and SMEs due to its capability of accurately detect objects and simple integration to the code solution (import from Ultralytics). In the experiments it is implied that the pre-trained model had better accuracy demonstrating higher confidence score with the bounding boxes in each product.

The final system integration StockSense provides high accuracy for object detection in 80 different objects using conventional hardware such as a laptop (MacBook) and smartphone (iPhone). Providing a localhost app deploying LLM recommendations with a custom prompt entered by the user, as well as business metrics context for the LLM insights. A second section where the last frame detected of the camera (smartphone) is shown as well as the product counting, identification, and notifications of the products detected in real time. Last section with a download bottom, that downloads a PDF document providing the LLM recommendation, a dashboard for current stock levels, a pie chart for the stock composition and the last frame detection with the confidence score bounding boxes detecting the products available. This project answers the question "Do pretrained YOLOv8n model offer a practical and accurate solution over a customer model for real-time inventory management and AI decision support for SMEs?" Since it was demonstrated that customizing a model requires much higher consumption for the training, research and deployment, as well as requiring specialised technical skills that are often limited in small business or SMEs.

The limitations and future work for this project is the possibility of custom train the model with different techniques, or different dataset for similar projects. Custom trained model can be ideal for specific inventory detection which pre-trained YOLOv8n lacks diversity of objects since it only detects 80 different products. Limitations start when the SMEs inventory items are not included in this list. Custom train can be effective if the small business has the resources to invest in a specialized solution tailored to their needs. Future work may focus on expanding the dataset with synthetic data, integrate multimodal inputs like sales APIs and aim for larger scalability. Also adding more AI features to the app, as well as deploying it cloud to provide quicker accessibility and installation.

References

Verma, N.K., Sharma, T., Rajurkar, S.D. and Salour, A., 2016. “Object identification for inventory management using convolutional neural networks”. In: “2016 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)”, Washington, DC, USA, pp.1-6. doi:10.1109/AIPR.2016.8010578.

Groenewald, E. and Kilag, O.K., 2024. “E-commerce Inventory Auditing: Best Practices, Challenges, and the Role of Technology”. “International Multidisciplinary Journal of Research for Innovation, Sustainability, and Excellence (IMJRISE)”, 1(2), pp.36-42.

J.P.N., Prashanth, S., D, A. and J, M., 2023. “Design and Evaluation of a Real-Time Stock Inventory Management System”. In: “2023 IEEE 5th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA)”, Hamburg, Germany, pp.180-185. doi:10.1109/ICCCMLA58983.2023.10346665.

Boysen, N., de Koster, R. & Weidinger, F. (2019) ‘Warehousing in the e-commerce era: A survey’, European Journal of Operational Research, 277(2), pp. 396–411. DOI: 10.1016/j.ejor.2018.08.023.

Patil, H. and Divekar, B.R. (2014) ‘Inventory Management Challenges for B2C E-commerce Retailers’, Procedia Economics and Finance, 11, pp. 561–571. DOI: 10.1016/S2212-5671(14)00221-4.

Wang, L.-p. (2009) ‘Study on the System Optimization of Inventory Management of SME’, 2009 International Joint Conference on Artificial Intelligence, Hainan, China, pp. 413–416. DOI: 10.1109/JCAI.2009.49.

Albayrak Ünal, Ö., Erkayman, B. and Usanmaz, B. (2023) ‘Applications of artificial intelligence in inventory management: A systematic review of the literature’, Archives of Computational Methods in Engineering, 30(4), pp. 2605–2625.

J. P. N., Prashanth, S., D., A. and J., M. (2023) ‘Design and Evaluation of a Real-Time Stock Inventory Management System’, 2023 IEEE 5th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA), Hamburg, Germany, pp. 180–185. DOI: 10.1109/ICCCMLA58983.2023.10346665.

Ikmel, G. and Najiba, E.A.E.I. (2024) ‘Performance Analysis of YOLOv5, YOLOv7, YOLOv8, and YOLOv9 on Road Environment Object Detection: Comparative Study’, 2024 International Conference on Ubiquitous Networking (UNet), Marrakech, Morocco, pp. 1–5. DOI: 10.1109/UNet62310.2024.10794724.

RA, V.S.K., Kathrine, G.J.W. and Salaja, S. (2024) 'Dynamic Inventory Object Detection Using YOLOv8 Deep Learning Model', 2024 International Conference on Advances in Modern Age Technologies for Health and Engineering Science (AMATHE), Shivamogga, India, pp. 1–6. DOI: 10.1109/AMATHE61652.2024.10582192.

Zotov, V.B., Demin, S.S. and Glazkova, I.Y. (2019) 'Features of material flow accounting for the efficient supply chain management', International Journal of Supply Chain Management, 8(3), pp. 483–492.

Fudholi, D., Kurniawardhani, A., Andaru, G., Alhanafi, A. & Najmudin, N., 2024. YOLO-based small-scaled model for on-shelf availability in retail. Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), 8(2), pp.265–271. doi:10.29207/resti.v8i2.5600.

Villegas-Ch, W., Maldonado Navarro, A. & Sanchez-Viteri, S., 2024. Optimization of inventory management through computer vision and machine learning technologies. Intelligent Systems with Applications, 24, p.200438. Available at: https://doi.org/10.1016/j.iswa.2024.200438.

Mupparaju, S., Thotakura, S. R. & Ch. Venkata Rami Reddy, C., 2024. A Review on YOLOv8 and Its Advancements.En: Data Intelligence and Cognitive Informatics, pp. 529–545. Springer. doi:10.1007/978-981-99-7962-2_39

Tur, A.O., Conti, A., Beyan, C., Boscaini, D., Larcher, R., Messelodi, S. & Ricci, E., 2024. Exploring fine-grained retail product discrimination with zero-shot object classification using vision-language models. In: Proceedings of the 2024 IEEE 8th Forum on Research and Technologies for Society and Industry Innovation (RTSI), Milano, Italy, pp. 97–102. IEEE. doi:10.1109/RTSI61910.2024.10761839.

Ahmad, H.M., Rahimi, A. & Hayat, K., 2024. Capacity constraint analysis using object detection for smart manufacturing. Automation, 5(4), pp.545–563.

AI Acknowledgement Supplement

StockSense: A Real-Time, AI-Powered Inventory Management System for SMEs Using Pretrained YOLOv8n and Streamlit.

MSc Research Project

Your Name/Student Number	Course	Date
Dana Carranza/23374900	Msc AI for Business	11/08/2025

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click here.

AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

Tool Name	Brief Description	Link to tool
Chat GPT	Code error fixing and explanation	https://chat.openai.com/

Description of AI Usage

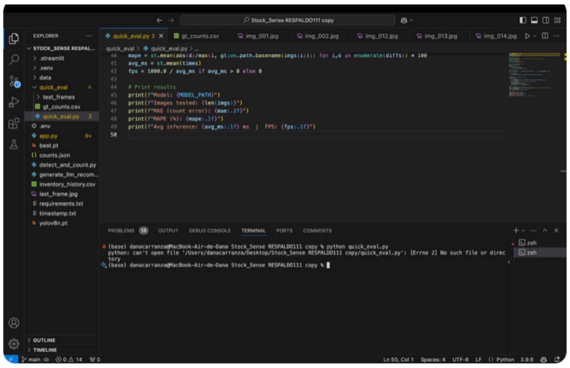
This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. **One table should be used for each tool used**.

Chat GPT	
Understand graphics.	
That means the terminal can't find the file you're trying to run because you're not in the same directory as quick_eval.py.	I provided a image with the error code.

Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

Evidence A:

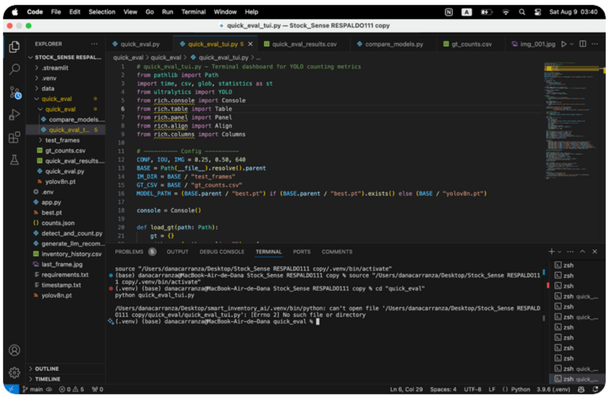


I see the issue — the error says:

swift

```
python: can't open file '/Users/danacarranza/Desktop/Stock_Sense RESPALD
```

That means the terminal can't find the file you're trying to run because you're **not in the same directory** as `quick_eval.py`.



Thought for 11s

You're super close—two little gotchas in that screenshot:

1. **Tiny typo**: the error path shows `quickx_eval_tui.py` (with an extra **x**). The file is `quick_eval_tui.py`.
2. You're in the right folder now (`…/quick_eval`), so a **relative run** will work.

Do this in the Terminal pane: