Department of Electronic & Telecommunication Engineering
University of Moratuwa

EN3160 - IMAGE PROCESSING AND MACHINE VISION

PROJECT- DENSELY PACKED PRODUCT DETECTION

MANIMOHAN T. *200377M*
SAIRISAN R. *200552V*

This report is submitted as the fulfillment of project of module EN3160

28^{th} October 2023

ABSTRACT

This report describe about the Project EN3160 - Image Processing and Machine Vision module, which conducted by Dr.B.K.R.P. Rodrigo

Retail inventory management is a pivotal component of the retail industry, and the adoption of efficient object detection methods holds the potential to revolutionize this sector. This research paper explores the application of the YOLOv8 object detection algorithm to the challenging task of detecting densely packed objects on supermarket shelves. The SKU-110K dataset, representative of retail environments, serves as the foundation for this study, with a focus on dataset characteristics, data preprocessing, automatic annotation, and model training. Performance evaluation is based on key metrics, and the broader implications of automated inventory management for the retail industry are discussed, including cost reduction, improved customer experiences, and operational efficiency. The study lays the groundwork for future research directions, including hyperparameter optimization, generalization assessments, and real-world implementation.

This research delves into the transformative potential of efficient object detection in retail inventory management. By applying YOLOv8 to densely packed supermarket shelves, the study addresses critical challenges in the retail industry. Leveraging the SKU-110K dataset, the research explores dataset characteristics, data preprocessing, and automatic annotation, leading to model training with a focus on hardware and hyperparameters. The evaluation of model performance based on key metrics underscores the significance of this approach in enhancing inventory management. The research extends beyond technical aspects to highlight the broader implications for the retail sector, paving the way for future investigations in this evolving domain.

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Background and Significance

The significance of this project(Densely packed product detection) report extends to the core of retail inventory management. Within the retail industry, the efficient handling of inventory is indispensable. Supermarket shelves laden with an extensive variety of products present a unique challenge for retailers. The complex task of detecting and locating objects in densely packed shelves can be arduous and labor-intensive. Automation offers a solution to this age-old challenge, promising significant benefits to retailers, customers, and the industry as a whole.

Efficient inventory management is more than a logistical exercise. It has a direct impact on a retailer's operational costs, profit margins, and, perhaps most importantly, customer satisfaction. An empty shelf can result in missed sales opportunities, while overstocking can lead to wastage. Consequently, precise and real-time inventory management is a strategic necessity.

The foundation of this project rests on the shoulders of YOLOv8, a state-of-the-art object detection algorithm. This report represents a pioneering application of YOLOv8 to the domain of retail inventory management. The synergy between this advanced algorithm and the complex reality of supermarket shelves filled with products exemplifies an innovative and practical use case, capable of transforming the way retailers manage their inventory.

1.2 YOLOv8: A State-of-the-Art Object Detection Algorithm

YOLO (You Only Look Once) has earned its place as a benchmark in real-time object detection. With each iteration of the YOLO series, a new standard in accuracy, efficiency, and speed is set. The evolution culminated in the introduction of YOLOv8 by Glenn Jocher in June 2020, representing a significant leap forward. YOLOv8 is characterized by its reduced size and increased accuracy, making it a pivotal choice for real-time applications.

In the context of this project, a deep understanding of YOLOv8 is essential to appreciate its potential in detecting densely packed objects on supermarket shelves. The architecture of the model, the intricacies of the training process, and its key features are all pivotal elements that will be dissected further in the ensuing sections.

GitHub Repository for the Project:- https://github.com/Manimohan05/Project--Densely-packed-product-detection
Demonstration video :- https://youtu.be/fmcZLSlWT60?si=qpLB5wKunmdxAZT4

2 DATASET DESCRIPTION

2.1 The SKU-110K Dataset

At the heart of this project lies the SKU-110K dataset, a rich repository of images captured in real-world retail settings. The dataset mimics the densely packed supermarket shelves found in everyday shopping scenarios. Familiarizing ourselves with the dataset's origin, attributes, and distribution is a prerequisite for conducting this project.

The dataset is more than just a collection of images; it is a reflection of the real-world complexities of retail environments. Understanding the scale, structure, and unique challenges presented by the dataset equips us to make informed decisions throughout the project process. In the following sections, we delve into the dataset's finer details.

2.2 Dataset Characteristics

A meticulous analysis of the SKU-110K dataset unveils insightful characteristics that are integral to shaping the project methodology. The images within the dataset vary in size and aspect ratio, mirroring the diverse nature of retail settings. The spectrum of object dimensions and layout influences critical decisions regarding data preprocessing, model architecture, and the selection of hyperparameters.

Understanding these dataset characteristics not only allows us to tailor our approach effectively but also prepares us for the challenges presented by real-world retail images.

Name	#Img.	#Obj./img.	#Cls.	#Cls./img.	Dense.	Idnt.	BB
UCSD (2008) [8]	2000	24.9	1	1	✓	✗	✗
PACAL VOC (2012) [13]	22,531	2.71	20	2	✗	✗	✓
ILSVRC Detection (2014) [12]	516,840	1.12	200	2	✗	✗	✓
COCO (2015) [28]	328,000	7.7	91	3.5	✗	✗	✓
Penguins (2016) [2]	82,000	25	1	1	✓	✗	✗
TRANCOS (2016) [34]	1,244	37.61	1	1	✓	✓	✗
WIDER FACE (2016) [49]	32,203	12	1	1	✗	✗	✓
CityPersons (2017) [51]	5000	6	1	1	✗	✗	✓
PUCPR+ (2017) [22]	125	135	1	1	✓	✓	✓
CARPK (2018) [22]	1448	61	1	1	✓	✓	✓
Open Images V4 (2018) [25]	**1,910,098**	8.4	600	2.3	✗	✓	✓
Our SKU-110K	11,762	**147.4**	**110,712**	**86**	✓	✓	✓

Figure 1 — sample image from provided as reference

3 METHODOLOGY

3.1 Preprocessing

Data preprocessing forms the foundation of our research methodology. It is a cornerstone of model training and plays a pivotal role in ensuring the success of our project. The preprocessing stage encompasses a series of actions aimed at transforming raw images into standardized inputs suitable for the object detection model.

One of the key steps in preprocessing is the resizing of images to a uniform 416x416x3 dimension. This step is fundamental as it standardizes the input size for our model, ensuring it can effectively process images of varying dimensions. Additionally, data augmentation and normalization are important preprocessing steps. Data augmentation introduces variety into the training dataset by applying transformations such as rotations, flips, and color adjustments. This diversification enhances the model's ability to generalize across different shelf configurations and product layouts.

3.2 Automatic Annotation

The annotation of data is a pivotal part of preparing the dataset for object detection. The SKU-110K dataset provides annotations in the form of CSV files, which specify object bounding boxes, class labels, and image dimensions. However, to streamline the annotation process, we leverage the automation capabilities of Roboflow.ai. This section provides a comprehensive overview of the annotation process, elucidating the use of CSV files and the automation of annotation through CVAT

Automation brings efficiency to the annotation process, saving valuable time and reducing the risk of human error. However, it also presents its own set of challenges, which we will address.

3.3 Data Generation

The generation of structured datasets for training and evaluation is indispensable. It involves the division of the dataset into training, validation, and test sets, each serving a specific purpose. The process of data generation is multi-faceted and crucial to the project.

The rationale behind our chosen data split, which comprises 70% for training, 20% for validation, and 10% for testing, warrants discussion. This split ensures that the model is trained on a representative portion of the data and evaluated on distinct datasets, fostering robust assessments of its performance.

Furthermore, this section will delve into the mechanics of data generation using CVAT, an efficient tool that simplifies the structuring of datasets for training and evaluation.

3.4 Hardware Requirements

* **GPU:** NVIDIA GeForce RTX 2060 with 6144MiB of memory.

– A GPU is essential for accelerating deep learning tasks such as object detection.

* **Memory:** 6144MiB of GPU memory.

– Sufficient for many deep learning tasks, though more memory is beneficial for handling larger datasets and complex models.

3.5 Environment Setup

* **Ultralytics YOLOv8:**

– Install YOLOv8 using pip:

```
pip install yolov8
```

– Download pre-trained weights if available for your specific project.
– Ensure you install the exact version YOLOv8.0.43 if needed for reproducibility.

Figure 2 — YOLOv8

* **Python 3.11.4:**

– Ensure Python 3.11.4 is installed. You can use tools like Anaconda or pyenv for Python version management.

* **Torch 2.1.0+cu121 (CUDA:0):**

– Torch is the deep learning framework.
– CUDA:0 indicates GPU support.
– Verify the correct installation with GPU support for your NVIDIA GPU.

* **CUDA:0 (NVIDIA GeForce RTX 2060, 6144MiB):**

– Ensure your environment is set up to utilize the NVIDIA GeForce RTX 2060 GPU with 6144MiB of memory.

* Consider creating a virtual environment using tools like `virtualenv` or `conda` to manage dependencies and isolate your project environment.

3.6 Model Training

Model training forms the heart of our research, as it is the point at which our chosen algorithm, YOLOv8, learns to detect objects in densely packed retail images. However, this process is resource-intensive and reliant on specialized hardware.

The choice of hardware is a critical factor influencing training times and the model's capacity to handle substantial datasets.Model training encompasses several nuanced aspects, including the selection of hyperparameters, the determination of the number of training iterations, and the calculation of various performance metrics. These factors directly impact the quality of the model's predictions. Thus, they require comprehensive exploration, and this section will detail their implications for the success of the research.

Additionally, the challenges and considerations specific to training YOLOv8 are worthy of discussion. This may involve configuring anchor boxes for object detection, optimizing the learning rate, and the benefits of transfer learning in this context.

4 RESULTS

4.1 Model Performance

The results of our project unveil the capabilities of our model in detecting and accurately bounding products in densely packed supermarket shelf images. The evaluation of the model's performance hinges on key metrics, including mean Average Precision (mAP), precision, recall, and F1 score.

This section will offer a detailed analysis of the model's performance. Visualizations of detection results on sample images will provide a vivid representation of the model's accuracy and efficiency. It is imperative to discuss how hyperparameters, training strategies, and architectural choices influence these metrics. For instance, the significance of anchor scale adjustments, learning rate, and the number of training iterations in achieving high mAP scores cannot be understated.

4.2 Implications for Retail Inventory Management

Beyond the technical evaluation, our research extends to the broader implications of efficient object detection in the context of retail inventory management. The automation of product identification and localization in supermarket shelves carries profound consequences for the retail industry. This section will delve into how precise and real-time object detection can revolutionize the management of inventory.

Automated inventory management optimizes stock levels, reducing the occurrence of empty shelves and minimizing overstocking. This directly affects operational costs and profit margins. Moreover, customers benefit from a more efficient shopping experience, where products are readily available and easy to locate. However, this section will also highlight potential challenges and considerations for implementing the model in real-world retail settings. These may encompass hardware requirements, scalability, and the need for periodic model updates to adapt to evolving retail environments.

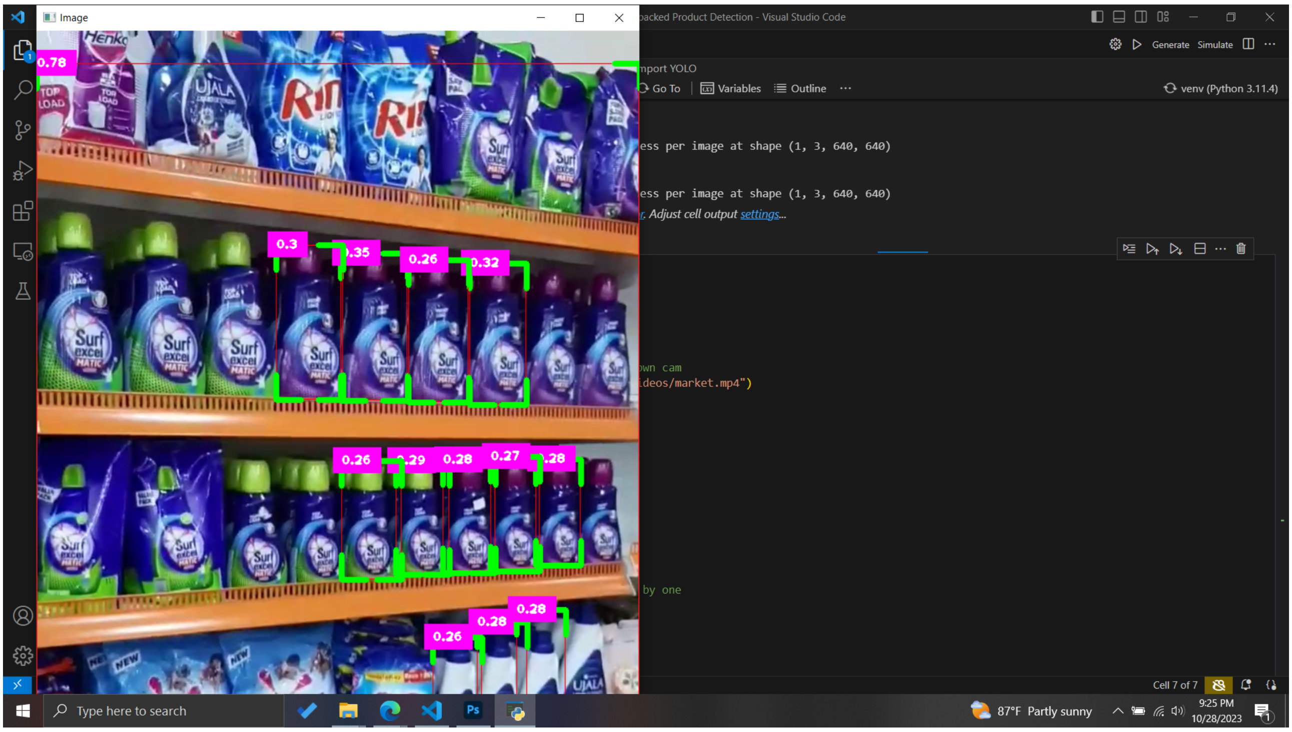

Figure 3 — Results-image 1

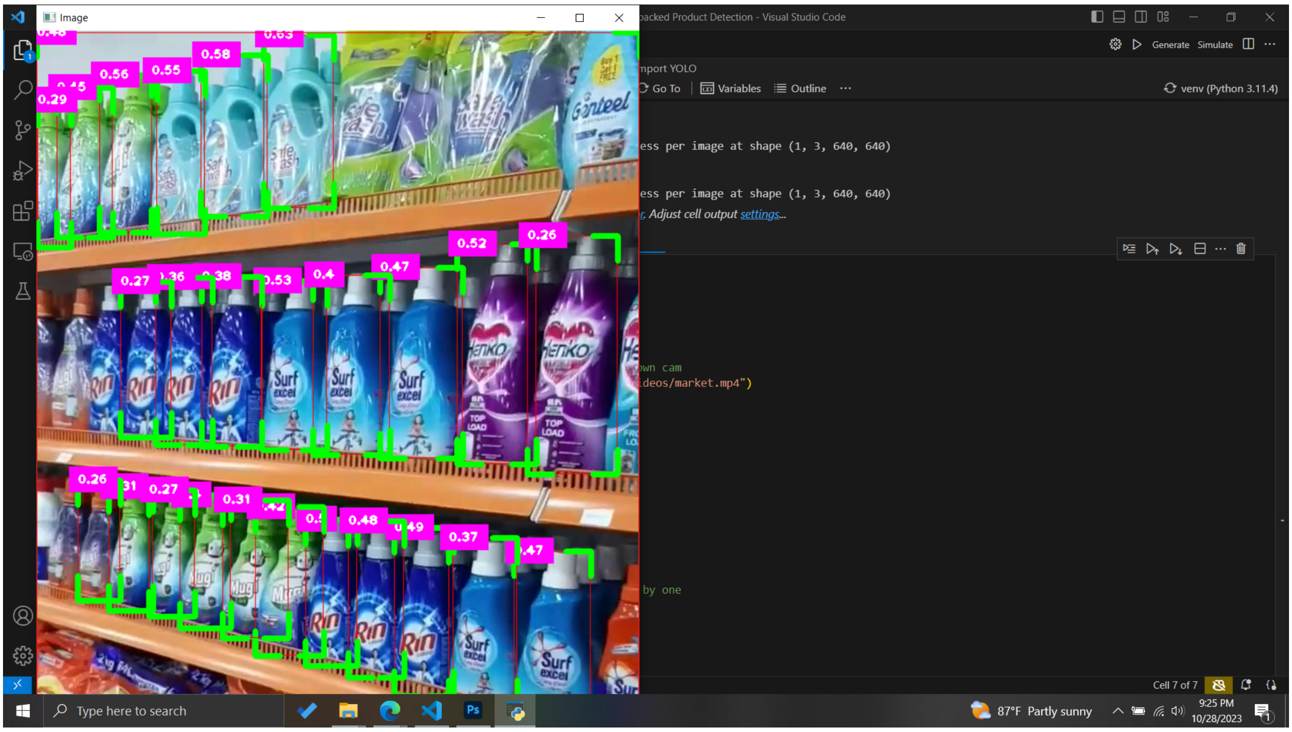

Figure 4 — Results-image 2

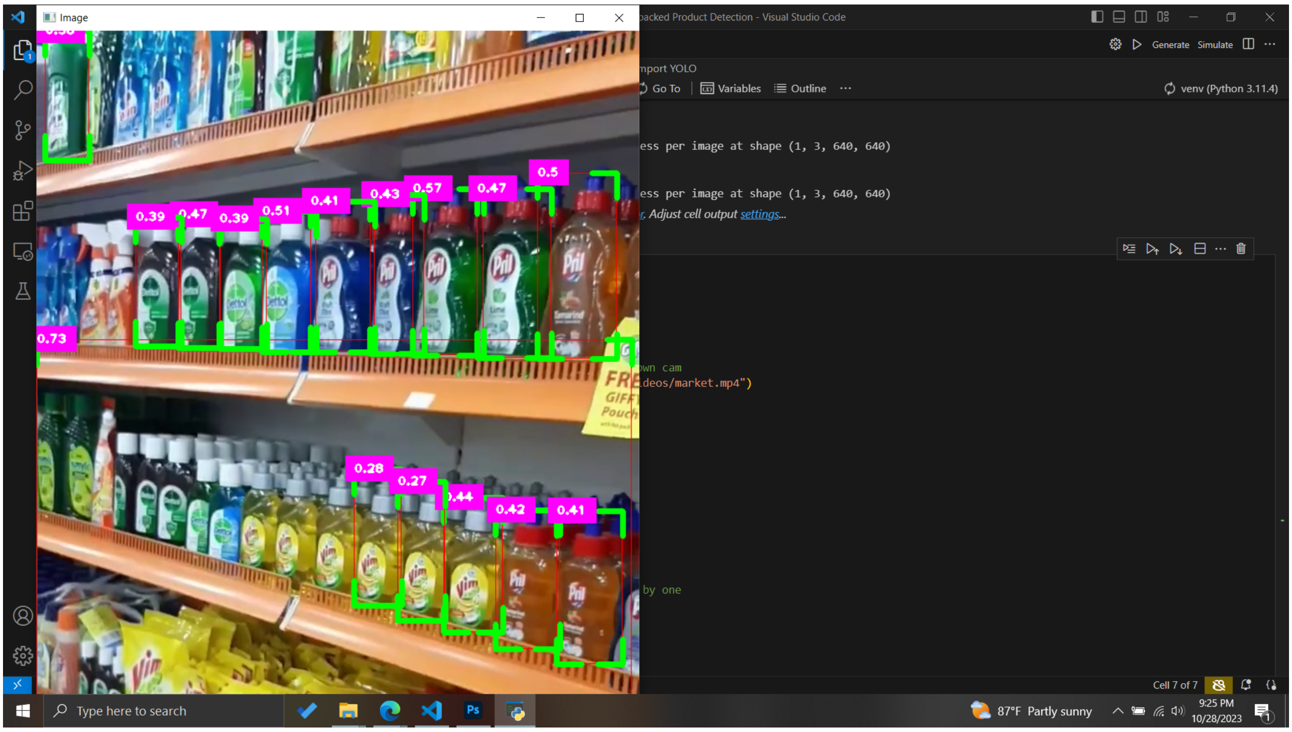

Figure 5 — Results-image 3

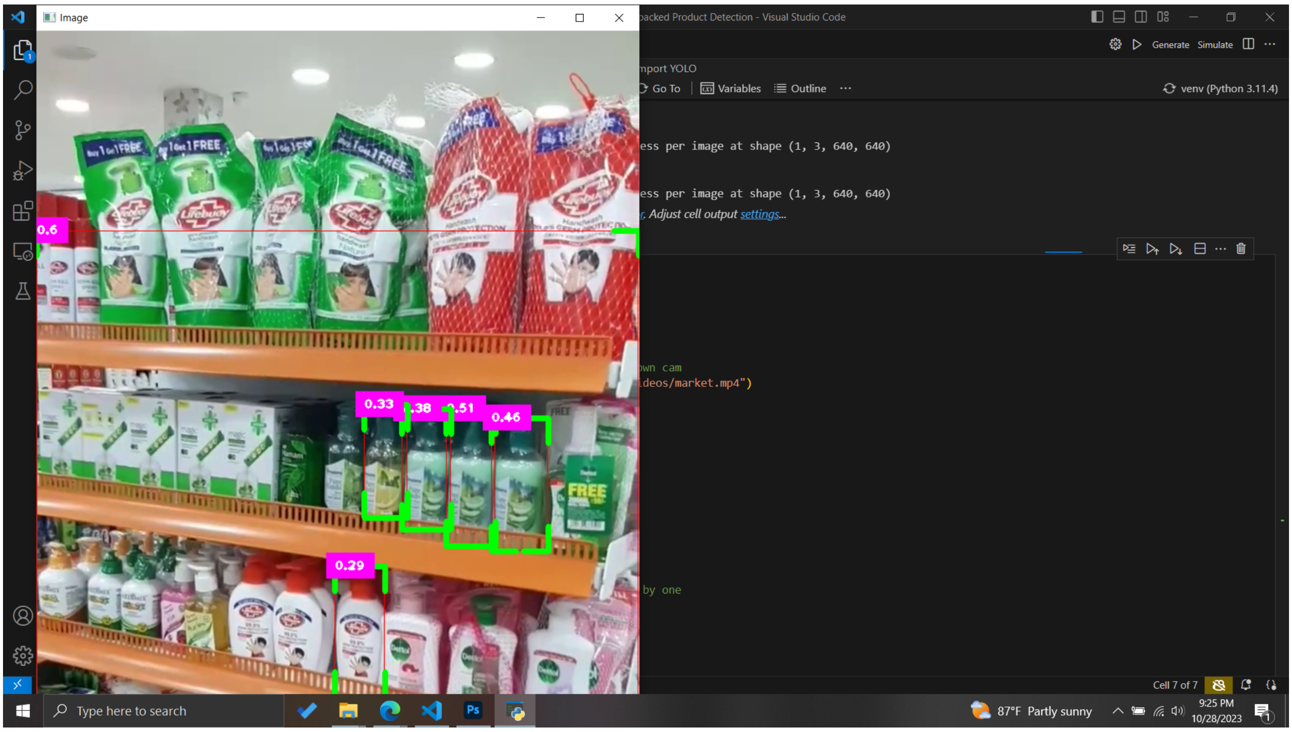

Figure 6 — Results-image 4

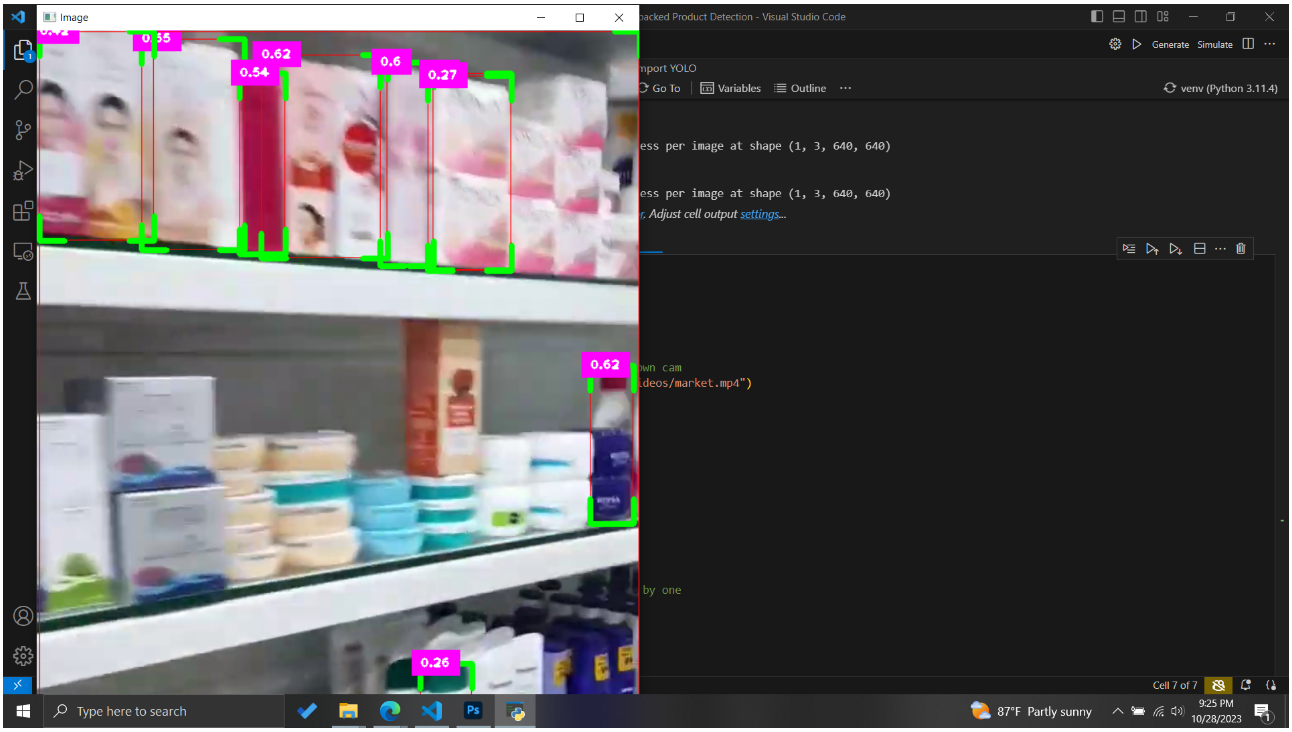

Figure 7 — Results-image 5

5 CONCLUSION

5.1 Key Takeaways

This project report embodies an innovative approach to object detection in densely packed supermarket shelves using YOLOv8. The journey commenced by addressing the multifaceted challenges within retail inventory management. This was followed by an introduction to the YOLOv8 algorithm and its successful application to the SKU-110K dataset. The key takeaways from our research encompass the adaptability of YOLOv8 to challenging real-world scenarios, the significance of dataset characteristics in shaping research methodology, and the transformative potential of efficient object detection in the realm of retail inventory management.

5.2 Future Directions

Our research provides a robust foundation for future investigations in the field of retail inventory management. Subsequent research may delve deeper into hyperparameter optimization, evaluating YOLOv8 on additional datasets to assess its generalization capabilities, and exploring real-world deployments in diverse retail environments.

Ongoing studies may focus on tackling the challenges presented by object occlusion, variations in lighting, and the real-time deployment of the model in extensive retail settings. As technology continues to evolve, so do the opportunities and challenges in the domain of retail inventory management. Our project opens doors to further exploration and innovation in this dynamic field.

5.3 Acknowledgments

This section expresses our gratitude to individuals (Dr.B.K.R.P. Rodrigo & Mr.Tharindu Wickremasinghe) who contributed to the success of this project. Their expertise and support were instrumental in advancing our understanding and knowledge in this domain.

6 REFERENCES

The references section presents a comprehensive list of citations, including research papers, articles, and other resources that were consulted and referenced in the course of this research. Proper citation is an essential component of maintaining academic integrity and facilitating further exploration for readers.

* CVPR 2020 challenge https://retailvisionworkshop.github.io/detection_challenge_2020/
  https://arxiv.org/pdf/2006.07825.pdf
* Redmon, J., Farhadi, A. (2020). YOLOv8: An Integrated Curriculum for Object Detection. . https://retailvisionworkshop.github.io/detection_challenge_2020/
* YOLOv8 https://youtu.be/m9fH9OWn8YM?si=ojcD1eQNt8igbLv_
* Retail Store Item Detection
  https://shorturl.at/cjnHI
* Ultralytics YOLOv8 GitHub Repository. https://github.com/ultralytics/yolov8
* Python Official Website. https://www.python.org/
* PyTorch Official Website. https://pytorch.org/
* NVIDIA CUDA Toolkit. https://developer.nvidia.com/cuda-toolkit

Listing 7.1 — Densely packed product detection

```
from ultralytics import YOLO

import cv2
import cvzone
import math

#cap = cv2.VideoCapture(0)   #turn camera 0- laptop own cam
cap = cv2.VideoCapture("Object tracking images and videos/market.mp4")
wid=640
hei=480
cap.set(3,wid)
cap.set(4,hei)

model_weights = 'YOLOweights/yolov8s.pt'

model = YOLO(model_weights)

while True:
    success, img = cap.read()   # read img frame one by one
    results = model(img, stream=True)

    for r in results:
        boxes =r.boxes
        for box in boxes:
            x1,y1,x2,y2= box.xyxy[0]
            x1,y1,x2,y2=int(x1),int(y1),int(x2),int(y2)
            #print(x1,y1,x2,y2)

            w,h=x2-x1,y2-y1
            #cv2.rectangle(img,(x1,y1),(x2,y2),(0,0,255),2)
            cvzone.cornerRect(img, (x1, y1, w, h),colorR=(0,0,255),
            colorC=(0,255,0),t=5,rt=1)
            conf=math.ceil(box.conf[0]*100)/100
            #print(conf)

            cvzone.putTextRect(img, f'{conf}' , (x1, y1+4),
            scale=1,thickness=2)
            cls=int(box.cls[0])

    cv2.imshow("Image",img)
```

```
    key=cv2.waitKey(1)
    # Check if the key pressed is '0'
    if key == ord('0'):
        break  # Exit the loop if '0' is pressed

# Release the video capture object and close all OpenCV windows
cap.release()
cv2.destroyAllWindows()

# Perform inference on the image
```

***