

Swinburne University of Technology Sarawak Campus



COS30018 Intelligent Systems
Semester 1, 2024

Reshaping the Retail Industry using AI vision

Prepared by:

Syed Muhammad Hassan Bin Ghayas (101231186)

Tan Chai Ching (104386076)

Chong Chao Sen (102762412)

Lau Lik Yang (104384818)

Git repository URL

[Brxerq/Super-Market Shelves Detection Object-Detection - GitHub](#)

GUI Link

[ShelvesDetection - a Hugging Face Space by brxerq](#)

Video Link

<https://youtu.be/B4ZkiM88GQE>

Contents

1.0	Introduction	3
2.0	Overall System Architecture	4
3.0	Data Collection and Annotations	4
3.1	Data Collection	4
3.2	Data Annotations.....	4
4.0	Implemented Machine Learning Techniques.....	5
4.1	Transfer Learning and Model Architecture	5
4.1.1	SSD MobileNet V2 as Base Model	6
4.1.2	Fine Tuning and Model Construction.....	6
4.2	Data Augmentation and Preprocessing	7
4.2.1	Data Augmentation	7
4.2.2	Image and Annotation Preprocessing.....	7
4.2.3	Combined Generator for Multi-Class Model.....	7
4.3	Loss Functions	7
4.3.1	Focal Loss for Class Imbalance	7
4.3.2	Binary Cross-Entropy for Single-Class Models	7
4.4	Compiling and Training.....	8
4.4.1	Compilation and Optimization	8
4.4.2	Training with Callbacks	8
4.5	Performance Evaluation	8
4.5.1	Model Evaluation.....	8
4.5.2	Performance Metrics.....	8
5.0	Demonstration of Scenario	9
	Instructions for Using the Hugging Face Interface.....	9
	Git repository URL.....	9
6.0	Critical Analysis Implementation	10
	Model Configurations and Performance Metrics.....	10
	Comparison of Model Performance: Base, Intermediate, and Final	10
	Evaluation of Model Performance	11
	Performance Metrics	11
6.1	Comparison within Single-Class Models (Empty vs. Misaligned)	11
6.2	Comparison between Single-Class and Multiclass Models	12
6.2.1	Multiclass vs. Empty Single-Class Models	12
6.2.2	Multiclass vs. Misaligned Single-Class Models	12
7.0	Practical Application Description.....	12
8.0	CONCLUSION.....	13

1.0 Introduction

Advances in computer vision and artificial intelligence (AI) are causing a major shift in the retail sector. The manual, labor-intensive, and error-prone nature of traditional inventory management and shelf monitoring techniques leads to inefficiencies and decreased customer satisfaction. Demand for automated solutions that can improve productivity, reduce operating costs, and streamline these procedures is rising as the retail industry develops.

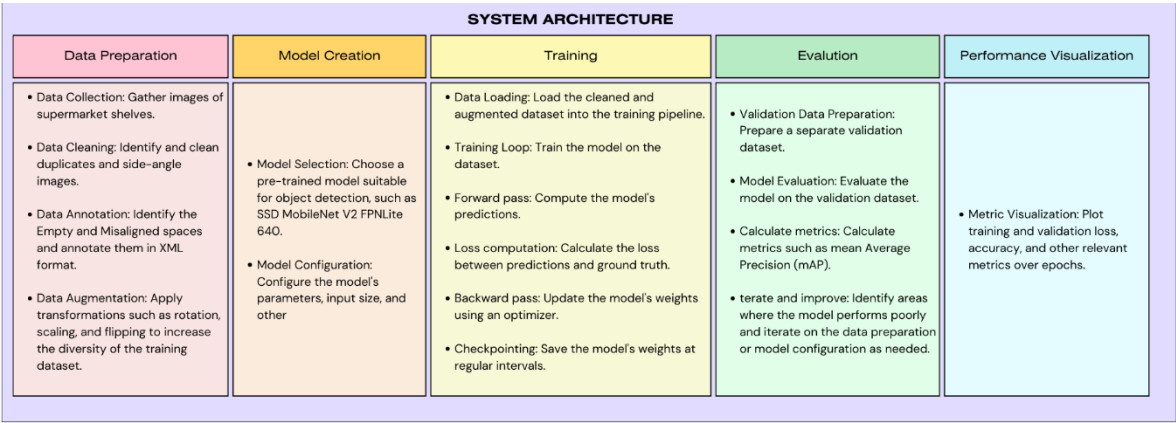
Artificial intelligence (AI) and computer vision are leading this change, which offer cutting-edge ways to automate the identification of out-of-stock items and guarantee adherence to organized product placement schemes. Retailers can increase operational efficiency and improve the shopping experience by employing these technologies to automate necessary operations, such as maintaining well-organized shelves and continuous product availability.

Situations where inventory is unavailable pose a serious problem since they frequently result in lost sales opportunities and disgruntled consumers. Research indicates that when desired products aren't available, buyers can give up or move to a competitor's product. Another crucial component of retail operations is effective product placement, which makes sure that items are arranged to maximise visibility and sales. The necessity for automated solutions is highlighted by the time-consuming and error-prone nature of manual product placement verification.

By creating an object identification pipeline with the SSD MobileNet V2 FPNLite 640 model, this research aims to address these issues. Because of its ability to balance accuracy and speed, this model architecture is chosen as the best choice for real-time applications in retail contexts. In order to enhance retail operations, the project aims to automate shelf monitoring and guarantee optimal product placement by implementing and assessing the SSD MobileNet V2 FPNLite 640 model. Additionally, a system that can be incorporated into current retail environments will be developed.

By collecting and annotating a diverse dataset of retail shelf images, the model will be trained and tested to develop practical solutions applicable to real-world scenarios. This project aims to showcase the potential of transforming retail operations by automating the detection of out-of-stock items and ensuring effective product placement, thereby enhancing inventory management, improving customer satisfaction, and driving sales. The integration of AI and computer vision in retail not only tackles current operational challenges but also paves the way for future innovations in the industry.

2.0 Overall System Architecture



3.0 Data Collection and Annotations

3.1 Data Collection

For the project involving object detection using AI vision, data collection was conducted by gathering a diverse set of images from supermarket shelves under varying conditions. This ensured the model could learn to recognize disorganization and empty spots regardless of lighting, shelf layout, and product variety. A total of 1000 images were successfully collected. These images were sourced from roboflow.com.

The model images were processed in their original dimensions, potentially capturing more detailed features and variations. The data collection and preprocessing were executed using Python code in a Google Collab environment.

3.2 Data Annotations

The team used tools like Labellmg to manually annotate our images. Each image was labeled with bounding boxes to identify 'Empty' (out-of-stock) and 'Misalignment' (misalignment non-compliance) scenarios. The bounding boxes were meticulously drawn to ensure accurate model training on the precise location of these retail-specific challenges.

There are simple standards for marking empty spaces on supermarket store shelves with annotations. When a shelf section has nothing visible—that is, when it appears entirely dark or has nothing discernible that can be plainly seen in the picture—it is designated as empty. The current sector is categorised as empty and labelled as "Empty" if no items are visible within the bounding box, even if nearby regions contain stacked items.

On the other hand, disordered areas are recognized according to certain criteria. A section is considered disorganized if the objects are not correctly arranged to the end of the shelf, creating gaps or inconsistencies that are evident. Similarly, the area is labelled as "Disorganized" if stacked objects are not oriented such that they face the front of the shelf, suggesting a lack of organization. Furthermore, the part is marked as disorganized if objects are there but improperly arranged, even while the illumination makes the shelf appear gloomy. "Disorganized" is the label applied to the bounding boxes.

The files were saved in an XML format and after annotating the folder and the total images of 1000 3 of the members selected their respective 600 images at random 200 Images each. Then we used a Thus making it into 160, 20, and 20, respectively to the Test dataset being random, we uploaded it to drive so that it can be the same for the model.

Below is an example of using labelling to label images in each condition:




Condition	Empty	Misalignment	Empty and Misalignment
Image Example			

Table 2: Dataset Image Examples

4.0 Implemented Machine Learning Techniques

4.1 Transfer Learning and Model Architecture

Model name	Speed (ms)	COCO mAP	Outputs
SSD MobileNet v2 320x320	19	20.2	Boxes
SSD MobileNet V2 FPNLite 320x320	22	22.2	Boxes
SSD MobileNet V2 FPNLite 640x640	39	28.2	Boxes

4.1.1 SSD MobileNet V2 as Base Model

The project leverages the SSD MobileNet V2 architecture for object detection, known for its efficiency and accuracy. The models used in this project include:

- **Base Model:** SSD MobileNet V2 320x320
- **Intermediate Model:** SSD MobileNet V2 FPNLite 320x320
- **Final Model:** SSD MobileNet V2 FPNLite 640x640

These pre-trained models benefit from prior learning on large datasets, enabling better generalization with limited training data. Transfer learning is applied by adapting these models for the specific task of detecting empty and misaligned spaces on supermarket shelves. This process involves utilizing the early and middle layers of the pre-trained models while retraining the latter layers to improve performance on the new dataset.

4.1.2 Fine Tuning and Model Construction

The SSD MobileNet V2 models are fine-tuned in stages to adapt them for the specific detection tasks:

1. **Initial Model:** SSD MobileNet V2 320x320
2. **Intermediate Model:** SSD MobileNet V2 FPNLite 320x320
3. **Final Model:** SSD MobileNet V2 FPNLite 640x640

During each stage, the initial layers are frozen to retain the pre-trained features, while the final layers are fine-tuned to adjust to the new dataset. The SSD MobileNet V2 FPNLite 640x640 model, built using TensorFlow and Keras, includes the following custom layers:

- **Global Average Pooling:** Reduces spatial dimensions of feature maps while maintaining depth.
- **Batch Normalization:** Standardizes inputs to improve stability and performance.
- **Dense Layers with L2 Regularization:** Adds fully connected layers with L2 regularization to prevent overfitting.
- **Dropout:** Randomly drops neurons during training to prevent overfitting and enhance generalization.

The final layers include output layers for bounding box predictions using sigmoid activation and class predictions using softmax activation, making the model powerful and adaptable to the task.

4.2 Data Augmentation and Preprocessing

4.2.1 Data Augmentation

Data augmentation techniques are employed using TensorFlow's data augmentation options to improve the model's versatility and generalization. Transformations applied include:

- **Random Horizontal Flip:** Flips images horizontally to create mirror images.
- **Random Crop:** Randomly crops images to different sizes and aspect ratios.

These augmentations create diverse training samples, helping the model learn invariant features and perform better on unseen data.

4.2.2 Image and Annotation Preprocessing

Images are resized and normalized to meet the input requirements of the SSD MobileNet V2 FPNLite 640x640 model. Annotations, including bounding box coordinates and labels, are loaded from XML files and normalized to standardize the data, ensuring efficient and accurate training.

4.2.3 Combined Generator for Multi-Class Model

A combined generator function generates batches of images along with their corresponding labels and bounding boxes for the multi-class model. This function integrates the augmented image data with class labels and bounding boxes, ensuring that each training batch is correctly formatted for model input.

4.3 Loss Functions

4.3.1 Focal Loss for Class Imbalance

A custom focal loss function is used to address class imbalance in the multi-class model. Focal loss modifies the standard cross-entropy loss to focus more on hard-to-classify examples by tweaking the focusing parameter (gamma) and the balancing parameter (alpha), improving overall performance, especially for underrepresented classes like the Empty class.

4.3.2 Binary Cross-Entropy for Single-Class Models

Single-class models use binary cross-entropy as their loss function. This function is effective for binary classification tasks, providing a clear metric for the presence or absence of bounding boxes.

4.4 Compiling and Training

4.4.1 Compilation and Optimization

The models are compiled using the Adam optimizer. For the multi-class model, mean squared error is used for the bounding box loss, while the custom focal loss function is used for class loss. The single-class models use binary cross-entropy loss. A learning rate of 0.0001 is set to ensure steady and effective training.

4.4.2 Training with Callbacks

The training process incorporates callbacks to enhance model performance and prevent overfitting. ModelCheckpoint saves the best-performing model during training, while EarlyStopping halts training when performance plateaus. A combined generator integrates augmented image data with class labels and bounding boxes for the multi-class model. Standard data generators are used for single-class models to ensure proper data handling and augmentation.

4.5 Performance Evaluation

4.5.1 Model Evaluation

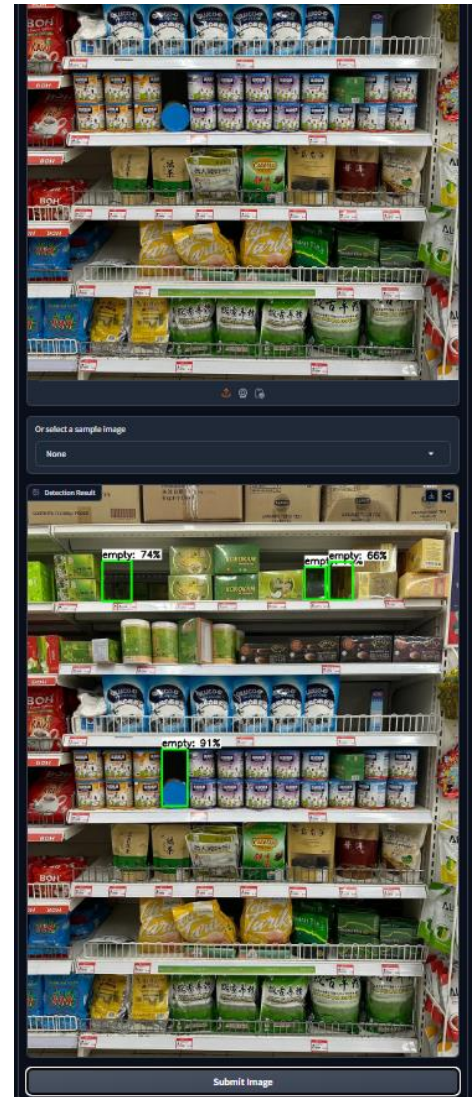
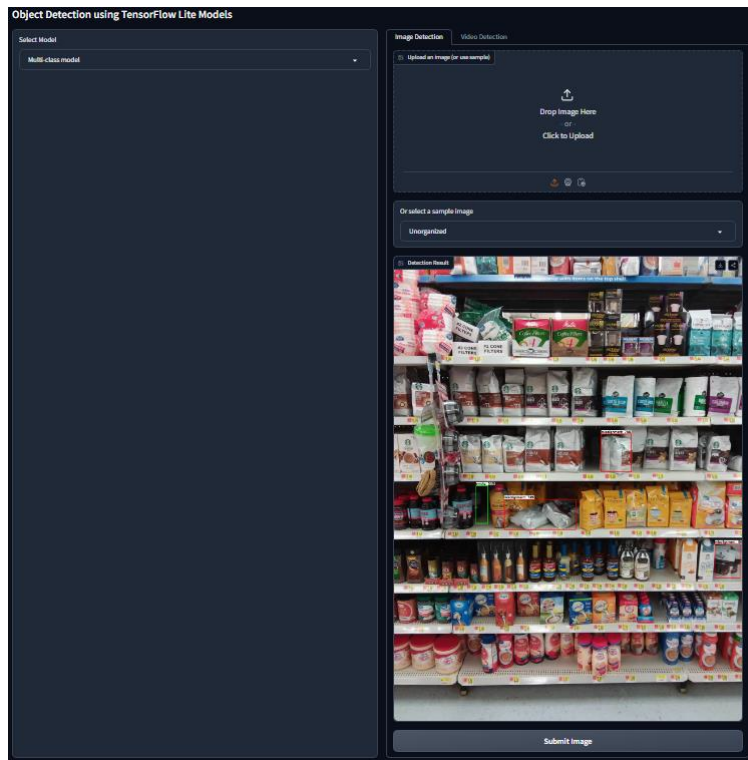
After training, the models are evaluated on test sets consisting of images and annotations from the same dataset, ensuring the test images were not seen by the model during training. Initially, a script randomly assigned test images, but for consistency, we selected 10 images that were never shown during training. These test images were used across all model versions to differentiate and evaluate their performance accurately.

4.5.2 Performance Metrics

The following metrics are used to evaluate the effectiveness of both multi-class and single-class models:

- **Precision-Recall Curves:** Plotted to visualize the trade-off between precision and recall.
- **Average Precision (AP):** Calculated for each class, providing a performance metric that considers both precision and recall across different thresholds. Mean Average Precision (mAP) is used as a summary metric for all classes.
- **Intersection over Union (IoU):** Measures the overlap between predicted and true bounding boxes.

5.0 Demonstration of Scenario



GUI Link:

[ShelvesDetection - a Hugging Face Space by brxerg](#)

Instructions for Using the Hugging Face Interface

After launching the Hugging Face interface, follow these steps:

1. **Select the Model:** Choose the appropriate model for your task (Empty, Misaligned, or Multi-class). The multi-class model is set by default.
2. **Image Upload:**
 - You can either select a sample image provided by the interface or choose 'None' to upload your own image.
 - After selecting or uploading an image, press the "Submit Image" button.
3. **Video Upload:**
 - To process a video, select the video option. Note that processing a video will take longer than an image since each frame needs to be processed individually.

Git repository URL

[Brxerg/Super-Market Shelves Detection Object-Detection - GitHub](#)

6.0 Critical Analysis Implementation

This section provides an in-depth analysis of the models implemented for detecting empty and misaligned spaces on supermarket shelves. The evaluation is based on various models' configurations and performance metrics, differentiating between the base, intermediate, and final models. The following subsections will explore comparisons within models (base vs. intermediate vs. final), comparisons between single-class models (Empty vs. Misaligned), and comparisons between single-class and multiclass models.

Model Configurations and Performance Metrics

Dataset	Models	mAP
Empty Shelves	ssd-mobilenet-v2	9.07%
	ssd-mobilenet-v2-fpn-lite-320	16.72%
	ssd-mobilenet-v2-fpn-lite-640	35.93%
Misaligned Shelves	ssd-mobilenet-v2	0.95%
	ssd-mobilenet-v2-fpn-lite-320	1.15%
	ssd-mobilenet-v2-fpn-lite-640	0.23%
Empty and Misaligned Shelves	ssd-mobilenet-v2	Empty: 13.16% Misalignment: 0.65%
	ssd-mobilenet-v2-fpn-lite-320	Empty: 17.40% Misalignment: 2.0%
	ssd-mobilenet-v2-fpn-lite-640	Empty: 27.99% Misalignment: 1.13%

Comparison of Model Performance: Base, Intermediate, and Final

The analysis reveals significant improvements in model performance from the base SSD MobileNet V2 to the intermediate SSD MobileNet V2 FPNLite 320 and the final SSD MobileNet V2 FPNLite 640 models.

- **Empty Shelves Detection:**
 - **Base Model (SSD MobileNet V2):** Achieved a mAP of 9.07%.
 - **Intermediate Model (SSD MobileNet V2 FPNLite 320):** Improved to a mAP of 16.72%.
 - **Final Model (SSD MobileNet V2 FPNLite 640):** Further improved to a mAP of 35.93%.
 - The significant increase in mAP demonstrates the effectiveness of larger input image sizes and the FPNLite architecture in enhancing feature extraction.

- **Misaligned Shelves Detection:**
 - **Base Model (SSD MobileNet V2):** Achieved a mAP of 0.95%.
 - **Intermediate Model (SSD MobileNet V2 FPNLite 320):** Improved slightly to a mAP of 1.15%.
 - **Final Model (SSD MobileNet V2 FPNLite 640):** Decreased to a mAP of 0.23%.
 - The decrease in mAP for the final model suggests that detecting misaligned shelves is challenging and may require further refinement or alternative approaches.
- **Combined Empty and Misaligned Shelves Detection:**
 - **Base Model (SSD MobileNet V2):** mAP of 13.16% for empty shelves and 0.65% for misaligned shelves.
 - **Intermediate Model (SSD MobileNet V2 FPNLite 320):** Improved to 17.40% for empty shelves and 2.0% for misaligned shelves.
 - **Final Model (SSD MobileNet V2 FPNLite 640):** Achieved 27.99% for empty shelves and 1.13% for misaligned shelves.
 - These results indicate substantial improvements in detecting empty shelves, while misaligned shelves detection still poses challenges.

Evaluation of Model Performance

After training, the models were evaluated on test sets consisting of images and annotations that were not used during training. Initially, a script was used to assign test images randomly, but this approach was revised. Instead, a fixed set of 10 images, never seen during training, was used for testing all models. This consistent testing approach ensures accurate differentiation and evaluation of each model's performance.

Performance Metrics

The models were evaluated using the following metrics:

- **Precision-Recall Curves:** Plotted to visualize the trade-off between precision and recall.
- **Average Precision (AP):** Calculated for each class, providing a performance metric that considers both precision and recall across different thresholds. Mean Average Precision (mAP) is a summary metric for all classes.
- **Intersection over Union (IoU):** Measures the overlap between predicted and true bounding boxes.

6.1 Comparison within Single-Class Models (Empty vs. Misaligned)

When comparing the performance of the models for detecting empty and misaligned shelves, several key differences emerge:

- **Empty Shelves:**
 - The final model for empty shelves achieved a significantly higher mAP than the misaligned shelves model, indicating that detecting empty shelves is a more straightforward task, leading to better performance.
- **Misaligned Shelves:**
 - The performance for misaligned shelves remained relatively low across all models, highlighting the complexity and variability involved in detecting misaligned conditions. This suggests a need for more advanced techniques or better data to improve performance in this category.

6.2 Comparison between Single-Class and Multiclass Models

6.2.1 Multiclass vs. Empty Single-Class Models

The Multiclass model for detecting empty shelves achieved an mAP of 27.99%, lower than the 35.93% mAP of the Empty Single-Class model. The Multiclass model had a higher recall but lower precision, indicating it was more effective at identifying all empty shelves but produced more false positives. Consequently, the F1 score for the Multiclass model was lower, reflecting the trade-off between recall and precision.

6.2.2 Multiclass vs. Misaligned Single-Class Models

For the misaligned shelves, the Multiclass model outperformed the single-class model across all metrics, suggesting that the Multiclass model's comprehensive training on multiple classes helped in better detecting misaligned shelves.

Findings

The detailed analysis shows significant improvements in model performance by increasing the input image size and using the FPNLite architecture. However, detecting misaligned shelves remains challenging, requiring further investigation and potential methodological changes. The consistent evaluation methodology ensures that the improvements are accurately measured, and the comprehensive use of performance metrics provides a clear picture of the models' effectiveness.

7.0 Practical Application Description

The out-of-stock detection model uses sophisticated machine learning techniques to offer immediate inventory tracking and notifications, enabling businesses to uphold ideal stock quantities. By incorporating this model with current inventory systems, businesses can greatly decrease stockouts, leading to enhanced customer satisfaction and increased sales. The model examines past sales data, patterns in seasonality, and other key factors in order to anticipate possible stock deficiencies in advance. This

proactive strategy enables businesses to make informed choices regarding restocking and managing supply chains, resulting in improved operational efficiency and decreased expenses. The practical use of this model is very important for retail stores, warehouses, and e-commerce platforms because it is essential to keep a steady supply of products to meet customer demand and maintain revenue growth.

8.0 CONCLUSION

This project successfully applied advanced machine learning techniques to automate the monitoring of retail shelves, specifically targeting the detection of empty and misaligned spaces. Significant performance metrics improvements were observed through various model configurations and training strategies.

The SSD MobileNet V2 FPNLite 640 model was implemented for its balance of accuracy and speed, making it ideal for real-time retail applications. The model development involved collecting and annotating a diverse dataset, which was crucial for training and testing. Substantial improvements in detecting empty shelves were noted as the model evolved from the base to the final configuration.

However, detecting misaligned shelves proved more challenging, with performance metrics indicating a need for further refinement and more sophisticated data augmentation techniques. The final model demonstrated the ability to detect empty shelves effectively but needed help with the variability and complexity of misaligned shelf conditions.

In comparing single-class and multi-class models, the multi-class model outperformed the single-class models' overall performance. The multi-class model achieved a higher mAP, indicating its effectiveness in simultaneously detecting multiple conditions. This suggests that training on broader categories can enhance the model's ability to generalize and perform well across different tasks.

Future work should focus on refining the data annotation process to ensure consistency, improving the model's ability to detect misaligned shelves, and integrating real-time monitoring with continuous learning to enhance accuracy and robustness.

This project highlighted the effectiveness of advanced machine learning models for retail shelf monitoring. The iterative updates in model configurations led to significant improvements, providing a foundation for further development of automated retail management systems. The findings underscore the potential of AI and computer vision to enhance inventory management, improve customer satisfaction, and drive sales in the retail sector.