SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY

A MINI-PROJECT REPORT ON

"ON-SHELF AVAILABILITY DETECTION USING YOLOv7"

Submitted in partial fulfilment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION SCIENCE AND ENGINEERING

Submitted by

Prajna Dutt M D (R20EK023)

Under the guidance of

Ms.Nandhini
Assistant Professor, School of C&IT

REVA University

Rukmini Knowledge Park, Kattigenahalli, Yelahanka, Bengaluru-560064
www.reva.edu.in

May 2023

DECLARATION

I, Prajna Dutt M D (R20EK023) student of B.Tech. CSSE, VI Semester, School of Computing and Information Technology, REVA University declare that the Mini-Project Report entitled **“On-Shelf Availability Detection using YOLOv7”** conducted under the guidance of **Ms. Nandhini** , **Assistant Professor**, School of Computing and Information Technology, REVA University.

I am submitting the Mini-Project Report in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Information Science and Engineering by the REVA University, Bengaluru during the academic year 2022-23.

I further declare that the Mini-Project or any part of it has not been submitted for award of any other Degree of REVA University or any other University / Institution.

Prajna Dutt M D (R20EK023)

SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY

CERTIFICATE

This is to certified that the Mini-Project entitled **"On-Shelf Availability Detection using YOLOv7"** carried out under my guidance for Prajna Dutt M D (R20EK023) bonafide student of REVA University during the academic year 2022-23. The above-mentioned student is submitting the Mini-Project report in partial fulfilment for the award of Bachelor of Technology in Information Science and Engineering during the academic year 2022-23**.** The Mini-Project report has been approved as it satisfies the academic requirements in respect of Mini-Project work prescribed for the said degree.

Signature with date	**Signature with date**
Guide	**Deputy Director**

Name of the Examiner	**Signature with Date**
1.	
2.	

ABSTRACT

The retail sector has been undergoing a significant transition for many years to improve customers' shopping experiences and to make services as seamless as possible. In the same light, artificial intelligence is revolutionizing retail operations by automating tasks that were previously handled by human labour. There are numerous innovative seamless checkout retail stores have emerged in the market such Amazon Go. The optical detection techniques are used in critical processes like restocking methodology, item interaction, consumer movement and gesture detection, and product recognition on shelves.

Restocking the products in empty places on the shelves is one of the important stages among the sub-components in a seamless system. Lack of or delayed out-of-stock detection may have a substantial negative impact on sales and customer satisfaction. In this study, a deep neural network model is created to help determine a product's availability by identifying vacant places on the shelf using a supervised learning technique and deep neural network algorithm. A model is trained using the YOLOv7 detection technique to recognize empty spaces in shelves to alert the assistant to replenish the out-of-stock items on the shelf. It also addresses key requirements and related issues that must be overcome during model training to create a high-performance and accurate model.

Key words: Deep Neural Network, Optical Detection, Yolov7, supervised learning.

ACKNOWLEDGEMENT

I like to express our deepest gratitude to **Dr. P. Shyama Raju,** Founder Chancellor for providing us the necessary facilities for the successful completion Mini-Project. I would also like to acknowledge **Mr. Umesh Raju**, Pro Chancellor, **Dr. M. Dhanamjaya**, Vice-Chancellor, **Dr. R C Biradar,** Pro Vice-Chancellor and **Dr. N. Ramesh,** Registrar for their constant support and endorsement through invaluable administration.

Also, I would also like to mention my sincere gratitude to **Dr. Mallikarjun Kodabagi,** Director, School of Computing and Information Technology for his constant support and encouragement. I would like to express our deepest gratitude to **Prof. Nandhini** -Program Coordinator for giving me an opportunity to take up the Mini-Project.

I wish to record my deep sense of gratitude and profound thanks to my guide, School of Computing and Information Technology, REVA University for her keen interest, inspiring guidance, constant encouragement with my work during all stages, to bring the Mini-Project report into fruition.

Table of Contents

List of Figures

1. Introduction

The first introductory chapter begins with describing the motivation and working environment behind this mini project. Subsequently, it formulates the primary objective for this study and finishes with an overview of the project structure.

1.1 Objective

Every retail industry in the world has been working diligently in recent years to offer more innovative shopping experiences to customers. Service providers and retailers are beginning to use an omnichannel strategy to deliver a top-notch consumer experience throughout the checkout process. Currently, numerous new solutions are being developed to ensure smooth checkout both in-store and on digital platforms. Due to the dynamic nature, instant results, and limitless possibilities, today technology is promising and experimenting with them can result in outcomes that were unimaginable a few years ago. As a result, “Grab and Go”, “Buy Now, Pay Later”, and “Just Walk Out” technologies have come into the picture.

On-shelf availability (OSA) is the most significant factor of seamless checkout stores, or any retail space with quickly moving consumer products. Since every component of the store is controlled by machine learning and computer vision techniques, the replenishment should also be handled automatically with little human effort. Empty shelf detection and automatic alerting of the assistant for replenishment play a significant role in seamless checkout operations. If a consumer can't find the item on the appropriate shelf or it's out of stock, there's a good chance they won't be interested in going back and will instead choose another store. This in turn has an effect on the current sales and future demand of the stores. Retail businesses must consequently place more of an emphasis on on-shelf availability given that it has a significant impact on business earnings and performance in order to retain long-term permanent customers.

Employees at retail enterprises typically manually check the availability of shelves to address the problem of product out-of-stocks, disorganized inventory, and misplaced items. However, this continuous human labour makes the solution unsustainable and monotonous. Because of this, these issues are more likely to require an expensive solution and are susceptible to errors and poor management. There are two forms of on-shelf availability. One is when the product is unavailable locally, and the other is when it is misplaced in the store and the customer cannot locate it. Both times, retailers struggled to retain their most valued customers, which damaged their bottom line.

1.2 Problem Statement

One of the most important elements of not only a seamless checkout system but of every retail environment is to detect On-Shelf Availability (OSA), commonly known as Out-of-Stock (OOS) detection. Retail revenues are impacted directly by a lack of attention to empty shelf space. Consequently, this study examines various machine learning and deep neural network models as well as the results of various approaches in order to address this issue. The project also summarizes the performance of various pre-trained models and demonstrates empty space detection on real-time data samples.

1.3 Research Questions

1. Which Deep Learning Model works best for detecting Empty Spaces on the Shelf?
2. How are data set collected and prepared for the supervised learning approach for this research?
3. What are the reasons for selecting YOLOv7 algorithm for performing On-shelf Availability Detection?
4. How efficient is the trained neural network model compared to already existing solutions?

1.4 Research Outline

Before proceeding into further specifics, the following paragraphs provide a concise explanation of the research structure:

Chapter 2 describes all the background works and related works done on empty space detection. Chapter 3 (Empty Shelf Space Detection) begins with presenting all the existing techniques that are used to detect vacant spaces followed by analysis and selection of algorithm to start the implementation. This chapter also includes the hardware requirements and few installation procedures required by the chosen algorithmic technique.

Chapter 4 (Preparation of Dataset for Model Training) deals with the importance of dataset and its attributes for neural network model training. It also describes the recommended criteria for data collection and guidelines followed during annotating process.

Chapter 5 (Model Training) is dedicated to illustration of model training for empty space detection. The chapter describes various training configurations and demonstrates fine tuning and inference trails using various YOLOv7 pre-trained models along with result metrics.

In Chapter 6 (Performance Evaluation) usability study that was performed to assess the developed approach. Both the setting of the study and its findings are reviewed. The outcomes of this thesis are summarized in Chapter 7 (Conclusion). Chapter 8 (Significance and Future Works) explains importance and offers a preview of potential improvements and future work.

1.5 Research Scope

In this experiment, using YOLOv7 single shot detector, the deep neural network model is trained employing a supervised learning approach. The focus of the study is to enhance model performance and accuracy by investigating several neural network models with various run-time modifications. Also, this research prioritizes on the reliance of appropriate training and test data required for modelling and aims at demonstrating how crucial it is to prepare accurate annotated data before training the model.

The scope of this research is extensive in the field of improvising user and retail store experiences. This is the significant element in any retail store that are transitioning to seamless checkout systems. The optical detection techniques are the future of new retail world and adopting automatic detection of out-of-stock items on the shelf is the major criteria to save and improve business sales.

2. Literature Survey

The retailers are adopting a variety of strategies to lessen the problems caused by supply shortages while maintaining high in-store shelf availability. Some of the methods employed include manual audits and supervision, RFID readers with integrated weight sensors, and consumer-grade depth sensors. Computer-based procedures are required since it is either difficult or expensive to integrate these methods with the store's present systems.

One of the methods employed for the automatic verification of OSA is to utilize inventory data of the store to estimate stock-outs as a substitute for OOS detection. But the on-shelf availability detection cannot rely solely on inventory data since it is also unlikely to have up-to-date and correct inventory information.

A study [1] showed that in 30% of cases, the product can be in the stores but not on the shelf (thus, not a full stock-out). RFID (Radio-Frequency Identification) technology is being used by certain well-known retailers to maximize shelf availability, inventory accuracy, out-of-stock replenishment and to enable real-time tracking of products or objects. The cost of the RFID tags, on the other hand, has historically been a major barrier to their broad usage [2]. It is frequently used on clothing retail shelves, but not typically on shelves for groceries. For shelf verification, the use of weight-sensing has been standard practice [3]. However, the system's basic infrastructure comes at a significant expense (sensors powered by wire). Also, some inaccurate readings could happen since other units of mixed products might be on the shelf. Additionally, it is necessary to configure the area to weight ratio for each adjustment.

Rosado suggested a supervised machine learning methodology [4] for empty shelf detection by employing Support Vector Machines. A computer vision and machine learning based method for identifying empty shelves from camera images was presented by Priyanwada et al. [5].

Deep learning algorithms have recently been applied in several research to address the OOS problem. Faster R-CNN algorithm was utilized by Chen et al. [6] to gather location data, which was then followed by different empty shelf identification methods. To identify or count items, Moorthy et al.'s project [7] used reference images and the SURF approach. With an optimized Cascade RCNN, Rong et al. [8] developed a variation of the random cropping technique to address the problem of densely packed scene detection in major retailers and obtained a mean average precision of 58.7% on SKU-110k [10]. Recently, Yilmazer and Birant [9] recursively trained their models using pseudo-labelled data using YOLOv4 utilizing 1500 images from the WebMarket dataset [11] to detect empty, virtually empty, and three product classes.

Likewise, there are numerous systems that employ supervised learning with SIFT (Scale Invariant Features Transform), Support Vector Machines, SURF and color histogram techniques, blob identification with machine learning, 3D shelf monitoring using RGB-D sensors with computer vision and conventional machine learning algorithms employing an image processing approach. Nevertheless, despite using a large, annotated dataset to train the model, the conventional machine learning and computer vision techniques produce low accuracy. Hence, deep neural network approaches are being utilized to increase the model's effectiveness, and several research using deep learning models has produced results that are more precise and accurate than those from conventional machine learning models.

3. Tools and Methodology

This research demonstrates an end-to-end deep neural network methodology for real time empty shelf detection using YOLOv7 algorithm. The objective of this research is a little different from usual object detection—in fact, it's the exact opposite, which is to identify empty spaces between the objects instead of objects itself.

3.1 Selection of Algorithm

Object detection is the process through which an algorithm can discriminate between different items. The act of detecting an object is referred to as detection, while the act of recognizing an object's nature is referred to as recognition. The detection phase makes use of bounding boxes, while the recognition phase makes use of the class probability, which predicts the output's class name. There are numerous deep neural network pre-trained models that can be customized to tackle the on-shelf availability challenge and convolution neural networks are the best for all object detection tasks. These neural networks have high performance metrics, are simple to use, and are inexpensive to compute. One such family of object detection techniques that uses convolutional neural networks for identification tasks in real-time is YOLO. As a result of their significantly better performance compared to other target identification algorithms, the YOLO family of object recognition systems is frequently used for recognition tasks. The success of the algorithm is due to its precision and quickness. YOLO established itself as the go-to remedy for object detecting problems thanks to its more advanced functionality and effective working procedures.

The YOLO model was first presented by Joseph Redmon in his paper "You Only Look Once: Unified, Real Time Object Detection" from 2015 [12]. CNN models were the most advanced object detection models up until that moment. Despite being precise, the RCN family of models required a long time to execute because they had to compete against proposed regional bounding boundaries, classify these regions, and then do post-processing to enhance the outcome. Since YOLO V1's 2015 release, the algorithm has gained a lot of support from both the community and competitors. There have been various versions released, starting with version 2 and ending with version 7. After YOLOv3, Joseph abandoned the project, stating that the computer vision was too powerful and would be manipulated. Only a few days after YOLOv4 was released, the company Ultra lyrics unveiled YOLOv5. The release dates of YOLO versions 6 and 7 in 2022 are very near together. Additionally, there have been a few modifications created in the interim, such as PP-YOLO and YOLOR, both of which offer advancements over the earlier iterations.

The recently released version of YOLOv7 offers the highest accuracy of all real-time object detectors, with an average precision of 56.8%, and can operate at 30 frames per second or more when utilizing a GPU V100. The performance of YOLOv7 is superior to that of all of its predecessors, including YOLO X, scaled YOLOv4, YOLOv5, and transformer and convolution-based object detectors. YOLOv7 is a better option than its predecessors because its average precision is 1.5 times higher than that of prior YOLO models.

3.2 Hardware Requirements for Model Training

Deep learning and artificial intelligence model training works best with GPUs because of their multi-tasking capacity. Multiple concurrent tasks can be computed more quickly on these machines because of their many cores. For our study, the YOLOv7 model training with smaller weights was performed on an NVIDIA Quadro P2000 with 4 GB RAM on a local machine. However, more complex models were trained using Google Colab, whose sessions begin with an NVIDIA K80 GPU and 12 GB of RAM.

4. Preparation of Dataset for Training

In supervised learning, a set of training data is given to the algorithm, and supervised models learn from ground truth data that data scientists explicitly annotate. The model uses this information to train itself to predict outcomes based on new inputs (inferencing). The dataset has to be properly annotated and labelled before being fed into the system in order to train the models using supervised learning technique. Real-world data collection, cleaning, annotation, and pre-processing pose a major obstacle for the model training. There are many issues with the images that are captured during data collection due to improper camera positioning and settings. For this study, we utilize the SKU-110k annotated dataset, which consists of 11,762 densely packed shelf photos with 110k classes that were captured in a normal retail environment. And these images are manually annotated using the VGG annotator to produce the yolov7 dataset.

There are numerous services where we can upload the image dataset, label the bounding box, and then download the annotated file for free. It is important to mark every empty space with one of the above tools and label the empty space as "vacant". For labelling the empty space, a variety of shapes including rectangles, squares, and polygons can be employed depending on the shape of the area. In this study, there is just one class, "vacant," since it only focuses on detecting empty spaces. The Figure 1 shows the annotation of vacant space in the image using image annotator.

Figure 1: Labelling ‘vacant’ class using Image Annotator

After the images have been annotated, the folder structure under the /yolov7/data/ directory must be organized as indicated in Figure 6. The dataset is split into the three subfolders train, validation, and test. In this experiment, 20% of the dataset is reserved for validation sets and 80% for the training set. Images for executing personalized tests and conducting inference can be found in the test folder. This is brand-new data that is not part of the training or validation dataset. The test images can be applied to the trained model to check the model performance. All of the images for the corresponding operations are placed in the images subfolder respectively, while the labels folder has the corresponding bounding box labels. The folder name has to be train, val, test images, and labels in accordance with the YOLOv7 model regulations. The pictorially representation of folder structure is shown in Figure 2.

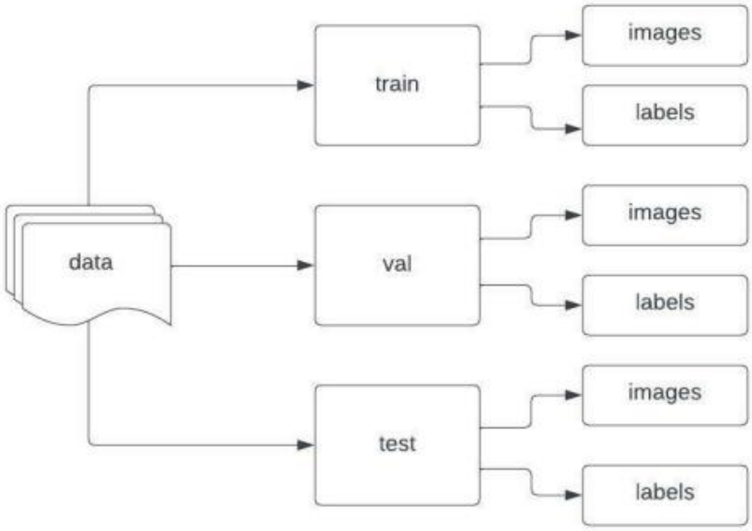

Figure 2: Dataset folder organisation in YOLOv7 repository

5. Model Training

After the initial installations and set-up, the most importantly dataset preparation the experiment can be started using YOLOv7 pre-trained weights. There are many versions of YOLOv7 models which can be utilized based on the requirements. Each model has a set of default parameters and a standard image size that it will accept. The YOLOv7, YOLOv7-tiny, and YOLOv7-W6 are some of the fundamental YOLOv7 models. YOLOv7 is the basic model that works the best for typical GPU computation. When compared to other versions of YOLOv7, the yolov7.pt version is the smallest model and has the fewest parameters. It accepts images up to 640 pixels in size. Faster processing times are achieved with smaller parameter.

For our experiment, we consider basic model and proceed to train the model using different hyperparameters. The detailed comparison of different YOLOv7 pre-trained models that is illustrated in the official YOLOv7 article is shown in Figure 3. It provides a detailed explanation of the necessary image size for each model during model training. As we get to higher models, it is also evident that processing times grow along and provides higher performance, or average precision on test data.

Model	Test Size	AP^{test}	AP_{50}^{test}	AP_{75}^{test}	batch 1 fps	batch 32 average time
YOLOv7	640	51.4%	69.7%	55.9%	161 *fps*	2.8 *ms*
YOLOv7-X	640	53.1%	71.2%	57.8%	114 *fps*	4.3 *ms*
YOLOv7-W6	1280	54.9%	72.6%	60.1%	84 *fps*	7.6 *ms*
YOLOv7-E6	1280	56.0%	73.5%	61.2%	56 *fps*	12.3 *ms*
YOLOv7-D6	1280	56.6%	74.0%	61.8%	44 *fps*	15.0 *ms*
YOLOv7-E6E	1280	56.8%	74.4%	62.1%	36 *fps*	18.7 *ms*

Figure 3: YOLOv7 Pre-trained Models

In this experiment, for the model training for empty space detection, we begin with the smallest model yolov7.pt on a local machine. The following code snippet must be added to the custom_data.yaml file in the /yolov7/data directory to indicate the repository's location for the train, val, and test data. The parameter nc = 1 shows how many classes are used for detection;

in our situation, there is only one, which is the 'vacant' class. The class name used for detection must be provided in the class names section, and this new file is saved in /yolov7/data.

```
train: ./data/train
val: ./data/val
test: ./data/test

# number of classes
nc: 1

# class names
names: [ 'vacant' ]
```

After the above update, under the \yolov7\cfg\training directory, a new .yaml configuration file has to be created 'yolov7_custom.yaml.' This is to configure the yolov7 model for vacant space detection. There are many configuration files in \yolov7\cfg\training directory which contain model configurations for different pre trained model weights. Following the initial setup, either locally on a system or remotely utilizing cloud services like Google Colab, the model training started by executing the following command. Each setting in the training command is configured for this experiment depending on the running environment, dataset size, hardware constraints, and desired outcome. To prevent overfitting and to establish baseline, the training is initially conducted with a less epochs.

Fine tuning p5 models for custom dataset:

! python train.py --device 0 --batch-size 4 --epochs 100 --img 640

--data data/custom_data.yaml

--hyp data/hyp.scratch.custom.yaml –

-cfg cfg/training/yolov7_custom.yaml

--weights yolov7.pt

--name yolo7x-custom

The description of all important flags in the training command is given below:

1. - - device: This is the GPU Id or number used for training. The experiment is conducted only on one GPU hence its 0. If the model training is running on CPU, instead of 0, the value has to be provided as cpu.

2. - - data: This represents the path to the dataset YAML file. In our case it is, data/custom_data.yaml.
3. - -img: This is explicitly mentioning image size for the training based on the yolov7 model version. For our model, image size is 640*640. By default, all the images will be resized to required resolution before being fed to the network. But it's still compulsion to mention.
4. - - cfg: This accepts the path for the model configuration file which we created before. It's required to load the yolov7 model architecture for training. cfg/training/yolov7_custom.yaml --weights This is to mention the pretrained model path. - - name There is subdirectory yolov7/runs, which stores the results of all the training, validation, detection and testing as individual sub directory after the respective process is complete. Here we can specify the name of the sub directory which will get created as a String name. --name yolo7x-custom .
5. -- hyp: Each yolov7 model has different parameters and hyperparameters. These parameters contain information such as learning rate, data augmentation and pooling techniques, and intensity of augmentations. All these are defined in hyperparameters files under data sub directory: data/hyp.scratch.custom.yaml. We have to specify an appropriate hyperparameter YAML file based on the pretrained model we are using.
6. --batch size: Batch size for parallel execution. It's recommended to train the model with the highest batch size (16 / 32) that is possible for the hardware.
7. – epoch: The number of iterations an algorithm makes around a training dataset.
8. --train.py: This is the training file that comes with the code when we clone the yolov7 git repository. All the training's operating procedures are present in this file.
9. –workers: This shows how many cores or threads will be required for training.

There are two key issues in any machine learning model training, Overfitting and underfitting that might arise throughout the learning process. Overfitting occurs when a model is too complicated for the data it is meant to model. One of the most common reasons for this is simply the model trying to learn too much from the input. When this occurs, the model memorizes the training data rather than identifying generalized patterns. Because of this, the model performs well on the training dataset but has trouble adjusting to new data. For this study for detecting vacant space on the shelf, the experiment is conducted using all YOLOv7 smaller pre-trained weights model. This pre-trained model yields different mean average precision depending on image size, the number of epochs and batch size. Once the training is completed, the results are stored in the ‘yolov7\runs\train’ directory with folder name specified in the respective training command using --name. The results of YOLOv7 contains a weights folder, confusion matrix, R curve, P curve, PR curve and each epoch results text file and summarized result metrics. Weights folder contains the weights during each epoch. But the file named ‘best.pt' is the best weights recorded during training and last.pt is the weights from the last epoch of training. The best.pt is saved whenever a new best fitness is observed during training on the validation set. The fitness of a model is defined as a weighted combination of metrics of precision, recall, mAP@0.5 and mAP@0.5:0.95]

The model training experiment started by considering the smallest YOLOv7 pre-trained model to build a model that can detect vacant places in the shelf. The training hyperparameters employed to carry out the first model training are displayed below, along with the corresponding outcomes. This training was carried out on a local machine that supported CUDA with 4GB memory.

```
! python train.py
--device 0
--batch-size 4
--epochs 600
--img 640
--data data/custom_data.yaml
--hyp data/hyp.scratch.custom.yaml
--cfg cfg/training/yolov7_custom.yaml
--weights yolov7.pt --name yolo7x-custom
```

During training, the model trained with 600 epochs achieved highest mAP compared all the previous training trails varying hyperparameters. The Figure 4 shows the confusion matrix for model training, which indicates that the model has 93% mAP on detecting vacant space in the shelf. A confusion matrix helps invisualizing the results of a classification task by providing a table layout of the variousoutcomes of the prediction and results. It creates a table with all of a classifier's predicted and actual values. In the final confusion matrix, the background FP describesbackground objects that are not a part of either class but are nonetheless recognized as one. Background FN indicates Trash or Non-Trash objects that the detector missedand are regarded as additional background objects.

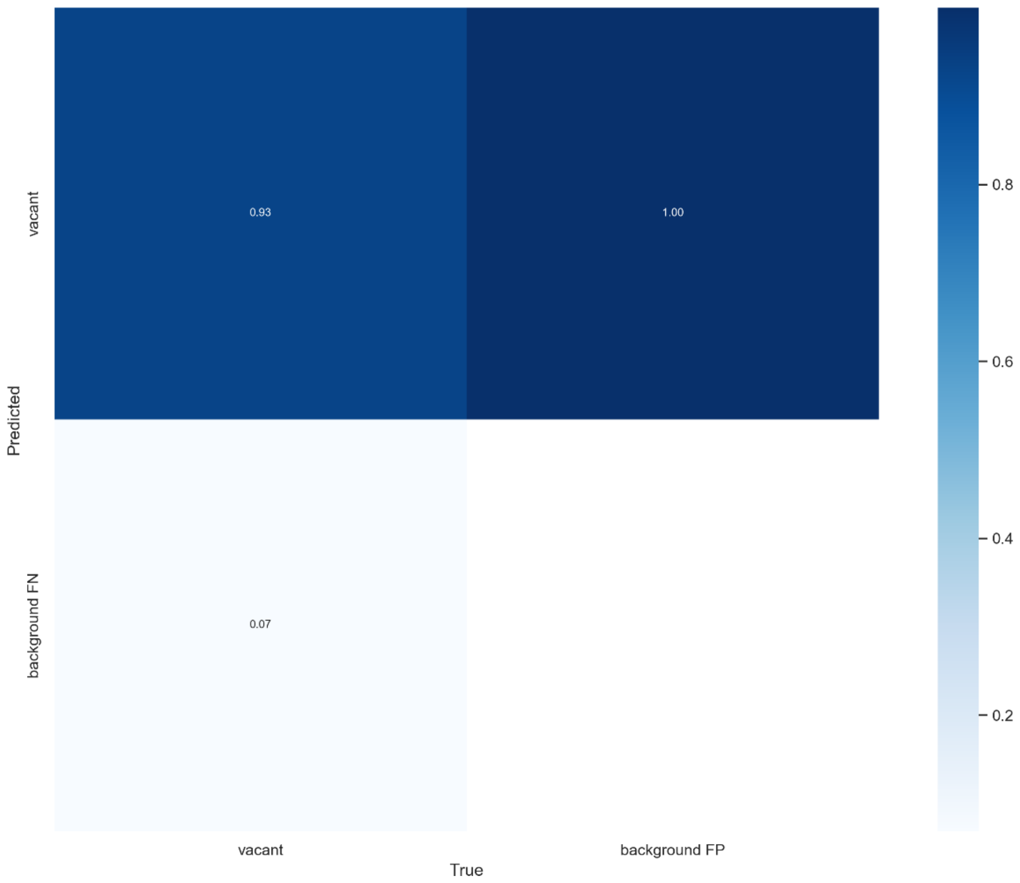

Figure 4: Confusion Matrix for best fit model trained with yolov7.pt.

After model training is finished, the good fit model can be used to detect vacant spacesin real time. This can process both image and video data, and it outputs predictions inthe form of a frame with a bounding box in real time. This model can detect empty spaces on the shelf in accordance with the quality, resolution, and variety of the dataset utilized during training. The results of the test inference for YOLOv7 good-fit model are clearly shown in Figure 5. The bounding box is predicted all around the vacant space in the image. The empty spaces may be seen to be correctly predicted on new test data using the good-fit model developed from YOLOv7 model training. The best fit model can clearly differentiate between the wall, shelf, and floor. The model also predicted comparatively smaller vacant space between the products. The results are accurate when there is a clear empty location on the shelf with no partial products.

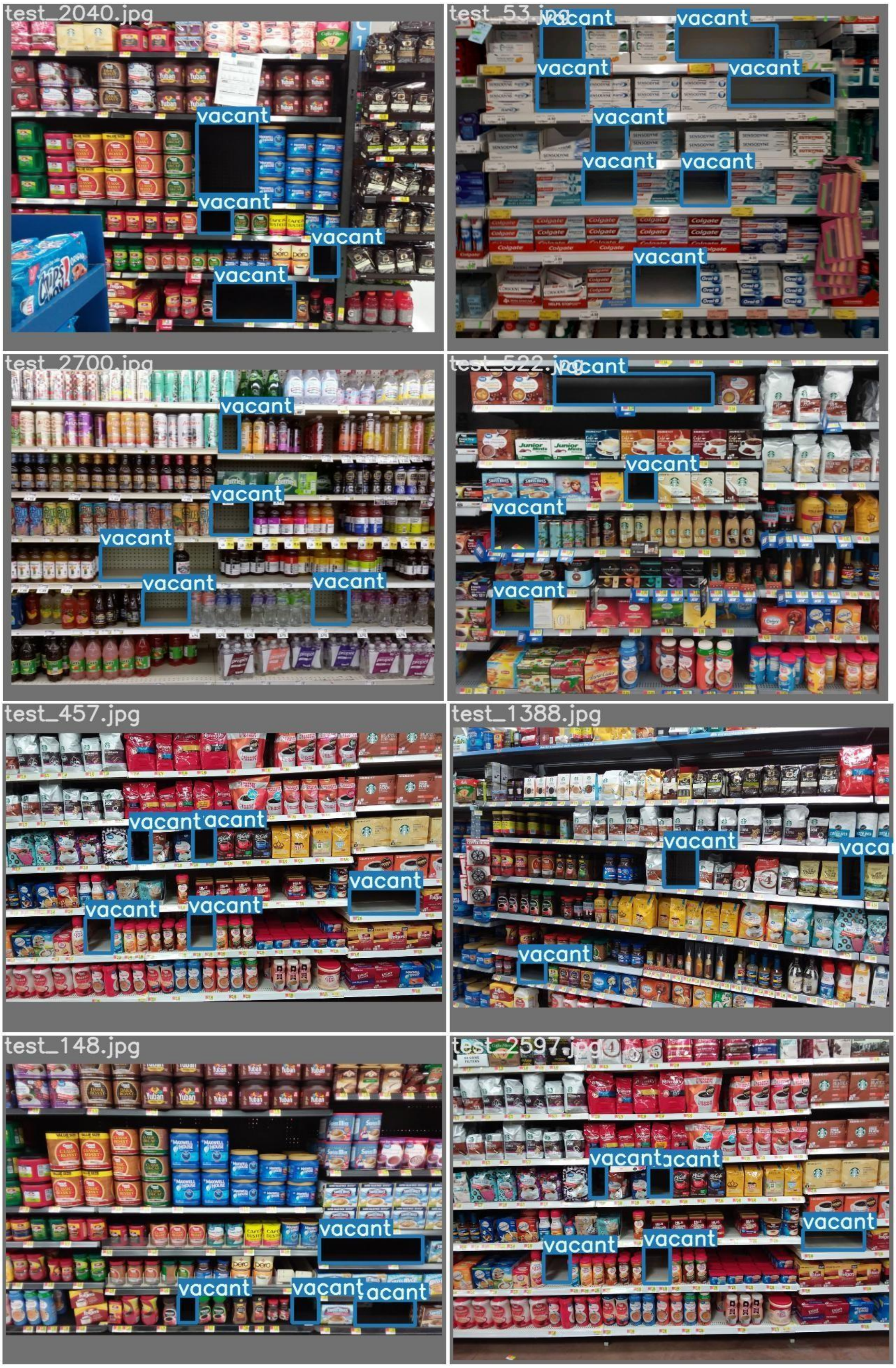

Figure 5: Empty shelf predictions using YOLOv7 good-fit model

6. Conclusion

In this research, we evaluated a dataset and developed a deployable, efficient deep learning model for real-time OSA detection in retail establishments. The experiment began with a dataset collected under the guidance of explicit instructions, annotated the data in line with clear rules, and then steadily increased the sample size until the learning curve achieved saturation. Various cutting-edge object recognition models were looked at while developing the models, along with thorough inference analyses and inference run-time optimizations. The study also provided a summary of its findings and in-depth exploration of the deployment-related concerns. The trained model predicted empty spaces with 93% mean average precision which is considerably good results based on quality of dataset.

In addition to resolving the issue of out-of-stock, this study, also lays out a clear machine learning development pipeline for creating beneficial applications for seamless system development. It largely focuses on the laborious but essentialdataset preparation process, which is a vital element of the supervised learning approach. According to the results of the experiments, the suggested strategy performs better than the already in use single shot detection methods. The research is the effort to employ YOLOv7 object identification strategy to monitor product availability on shelves. Considering how recently the YOLOv7 model was introduced, there is still a lot of possibility for innovation and development.

7. Significance and Future Works

From this work, it may be inferred that anyone, not just those in the retail industry intending to build their own end-to-end real-time deep learning systems from the ground up, would find it useful. This experiment is the first step in understanding how seamless technology works; in the future, it may be possible to deploy this solution inmore contexts, optimize inference time, and train it using more cutting-edge models.

The model employed in this study utilized supervised learning, which necessitates labelling the images manually for the object that is intended to be detected. In the future, this challenge can be investigated using semi-supervised or unsupervised learning techniques in place of supervised learning. There is currently very few research on OSA detection utilizing an unsupervised approach and semi-supervised learning. Since it can get rid of the limitations of the currently constructed model, this is most likely a possible area for improvement in this OOS situation.

The study serves as a foundation for understanding deep learning concepts and its ability to resolve challenging problems in the real world. This unquestionably makes it possible to consider more broadly about the complexity, challenges, and work involvedin building a complex system, such as seamless real-time checkout systems. With every day substantial progress in the disciplines of deep learning and AI, there are countless possibilities and opportunities to create better, more effective solutions.

References

[1]. E. C. R. Europe, ECR-optimal shelf availability: increasing shopper satisfaction at the moment of truth. ECR Europe, 2003

[2]. D. Papakiriakopoulos, K. Pramatari, and G. Doukidis, “A decision support system for detecting products missing from the shelf based on heuristic rules,”Decision Support Systems, vol. 46, no. 3, pp. 685–694, Feb. 2009.

[3]. Nguyen, Kien & Le, Minh & Martin, Brett & Cil, Ibrahim & Fookes, Clinton. (2022). When AI meets store layout design: a review. Artificial Intelligence Review.

[4]. Rosado, L. and Gonc¸alves, J. and Costa, J. and Ribeiro, D. and Soares, F. Supervised learning for Out-of-Stock detection in panoramas of retail shelves. In IEEE International Conference on Imaging Systems and Techniques, pages406–411, 10 2016.

[5]. H.A.M Priyanwada, K.A.D Dilanka Madhushan, Chethana Liyanapathirana, and Lakmal Rupasinghe. Vision based intelligent shelf-management system. In 20216th International Conference on Information Technology Research (ICITR), pages 1–6, 2021

[6]. Jun Chen, Shu-Lin Wang, and Hong-Li Lin. Out-of-stock detection based on deep learning. In International Conference on Intelligent Computing, pages 228–237.Springer, 2019

[7]. Rahul Moorthy, Swikriti Behera and Saurav Verma. On-Shelf Availability in Retailing. International Journal of Computer Applications (0975 – 8887) Volume 116 – No. 23, April 2015, PP. 47-51.

[8]. Tianze Rong, Yanjia Zhu, Hongxiang Cai, and Yichao Xiong. A solution to product detection in densely packed scenes. arXiv preprint arXiv:2007.11946, 2020

[9]. Ramiz Yilmazer and Derya Birant. Shelf auditing based on image classification using semi-supervised deep learning to increase on-shelf availability in grocery stores. Sensors, 21(2):327, 2021

[10]. Annalisa Milella, Roberto Marani, Antonio Petitti, Grazia Cicirelli, and Tiziana D'Orazio. 3d vision-based shelf monitoring system for intelligent retail. In International Conference on Pattern Recognition, pages 447–459. Springer, 2021.

[11]. Kaggle - https://www.kaggle.com/manikchitralwar/webmarket-dataset, 2022

[12]. Redmon, Joseph and Divvala, Santosh and Girshick, Ross and Farhadi, Ali. YOLO: Real-Time Object Detection. 2016.