Computer Vision Applied to Smart Markets: a Case Study for Empty Shelf Detection

Renzo P. Mesquita[1,2], Eduardo H. Teixeira[1],
Avner J. G. Ribeiro[1], Bernardo D. Gandolpho[1], Fábio Luiz F. Pontes[1]

[1] Instituto Nacional de Telecomunicações (INATEL)
Av. João de Camargo, 510, CEP: 37536-001 – Santa Rita do Sapucaí – MG – Brazil

[2]Instituto de Engenharia de Produção e Gestão (IEPG)
Universidade Federal de Itajubá (UNIFEI)
Av. BPS, 1303, CEP: 37500-903 – Itajubá – MG – Brazil

renzokuken31@gmail.com, eduardo.teixeira@inatel.br,
{avner.jose, bernardo.dalboni, fabio.pontes}@gec.inatel.br

Abstract. *Computer vision technique is attracting the attention of many industries once it has allowed the deployment of powerful applications capable of detecting patterns and changes in different kinds of environments by using popular hardware and software. Retailing is an example of an industry that has been leveraging the use of this concept in order to create resourceful environments such as smart markets. This paper focuses on demonstrating computer vision's applicability to detect empty spaces in supermarket shelves using a custom-trained YOLO model. This study provides insights into the practicality of using computer vision and a small single-board computer (SBC) in retail spaces and shows how the results found can be useful to be applied in real-life scenarios.*

1. Introduction

In recent decades, the Internet of Things (IoT) has gained significant prominence in the domains of technology and innovation. The exponential growth of interconnected devices, combined with the seamless integration of IoT technologies into everyday life, has ushered in a transformative era for the technological landscape [Ghosh et al. 2018]. This notable rise has been facilitated by a convergence of affordable hardware, advanced sensor technology, robust connectivity solutions, and powerful data analytics tools. These advancements enable the collection, transmission, and analysis of extensive data from a vast array of devices.

A domain in which IoT has been expanding is the retailing industry through the emergence of smart stores, often referred to as smart markets. The term *smart market* denotes a suite of technologies designed to offer customers a more efficient and intelligent shopping experience, while also enhancing management processes for retailers. These technologies can provide various features, including gondolas stock control, product tracking, automatic checkouts, and supply chain management. These advantages can be achieved by integrating a diverse set of technologies, featuring the use of IoT with cameras, sensors, and additionally by integrating artificial intelligence with concepts of computer vision and deep learning, allowing more efficient and automated decision-making in the retail environment [Hwangbo et al. 2017].

Additionally, inventory management is crucial for any store, as stock-out issues can have severe consequences. They can lead to customer frustration, loyalty decrease, and significant financial losses. In Europe, for instance, it's estimated that the industry incurs losses of up to 400 billion euros annually due to stock-out problems [Sanchez-Ruiz et al. 2018]. Therefore, it is important that retailers have efficient control over their product availability and disposal, so they can optimize their inventory management process, reduce money wastage, and enhance customer satisfaction by ensuring products are always available when needed.

Aiming to tackle this problem, computer vision, an artificial intelligence field that allows computers to interpret and identify digital images, has been a fundamental technology in the continuous expansion of IoT. One of the most famous real-life use cases is the *Amazon Go* [Wankhede et al. 2018], an Amazon's smart store initiative that relies on computer vision to track customers' actions, and by integration with other technologies, can provide a *"Just Walk Out"* experience, offering a completely new way of shopping. This initiative has started a race among big retailers and small tech startups worldwide [Times 2018], as they seek to innovate and establish a new global trend in smart store automation. However, despite the innovation proposed by companies like Amazon, their solutions are generally proprietary and little is known about how they could be used to help other companies and smaller establishments.

Therefore, in order to contribute to the literature in the area and provide a solution for popular use, this article seeks to implement a computer vision solution capable of supporting smart markets using modern machine-learning techniques and trained datasets to identify void spaces in supermarket gondolas. The efficient utilization of shelf space is crucial for retailers, as it can help them get a better view of their inventory levels and positively impact their customers' experience.

This article is organized as follows: in Section 2, important concepts about computer vision and related works are presented. Section 3 presents the material and methods employed to implement the proposed solution. Section 4 presents the experiments and results, and finally, in Section 5, the authors' conclusions are presented.

2. Background

2.1. Computer Vision

Computer vision is the field of artificial intelligence that enables computers to retrieve information from images, identify objects, detect movement, find patterns, track items, and more [Szeliski 2022]. This capability is essential for a wide range of applications, making computer vision a necessary part of many innovative solutions including autonomous vehicles, robotics, facial recognition systems, and smart markets.

Within computer vision, the concepts of machine learning and neural networks are fundamental components. Initially, a machine learning model must be created or selected from those available in the literature. The model is then trained with a dataset, aiming for it to learn to detect patterns among these data. In the realm of computer vision, these data consist of images pertinent to the context in which the model will be applied, generally labeled with information that the system aims to detect [Shanmugamani 2018]. The model processes the data, learning patterns and associations within it, until it becomes capable of recognizing and interpreting images that exhibit similar characteristics. Once trained, the model can be introduced to a new set of images, and upon using the knowledge acquired before, it can extract information from them and, therefore, be incorporated into other systems.

2.2. Computer Vision solutions applied in Smart Markets

To showcase the range of problems that computer vision can address in smart markets, a deeper examination of the Amazon Go case is presented. This particular implementation of a smart market has solved several challenges using computer vision to achieve the "Just Walk Out" shopping experience[Gross 2019]:

1. **Person identification:** Each person on the market is identified and tracked all the time by crossing information from different cameras and sensors;
2. **Product identification:** Similarly, all the products available on the shelves must be tracked and identified in a way that is possible to tell which product a specific customer removed from the shelf;
3. **Customer-product association:** Amazon Go also associates each customer to the products they removed from the shelf, for that it is needed a combination of solutions number 1 and 2. This solution also needs to track each customer's pose to determine their intention of picking up a specific item;
4. **Entry and exit detection:** Another scenario that Amazon Go smart markets cover is whenever a new customer enters the market or whenever a customer leaves. This makes it possible to create a shopping session for every active customer and charge each one of them for the items they removed from the shelves once they leave the store.

In the domain of smart market and inventory management systems, some significant works have contributed to the advancement of this field. These works, each offering unique insights and solutions, showcase the various approaches adopted to address challenges presented by modern retail environments. Some important works found in the literature are briefly presented in the following subtopics.

2.2.1. Computer Vision for Inventory Management

The work developed by [Kalahiki 2020] addresses the challenges of implementing deep learning-based computer vision systems within Internet of Things (IoT) devices. The authors explore techniques such as data collection, data preprocessing, and transfer learning to mitigate resource constraints. While their focus is primarily on technical aspects of computer vision, the relevance lies in the potential synergy with IoT technologies, aligning with the objective of enhancing inventory management in smart stores. This work provides a valuable reference for utilizing the power of computer vision in resource-constrained environments.

2.2.2. Revolutionizing Retail Stores with Computer Vision and Edge AI: A Novel Shelf Management System

The work proposed by [Savit and Damor 2023] presents an innovative approach to tackle disorganized features in retail stores. Leveraging computer vision and edge AI, the system aims to enhance the shopping experience for customers and optimize store operations. State-of-the-art object detection, image classification, and optical character recognition (OCR) models are implemented to identify shelf objects and voids, facilitating efficient restocking and detection of misplaced items. Task scheduling and model optimization for edge deployment are integrated to ensure seamless performance.

3. Materials and Methods

In the following subtopics, details about the hardware and software employed in the development of the proposed prototype are explained. This work seeks to implement a computer vision solution using machine-learning techniques and a SBC to identify void spaces in supermarket gondolas. Figure 1 provides a diagram illustrating the overall architecture of the proposed solution system.

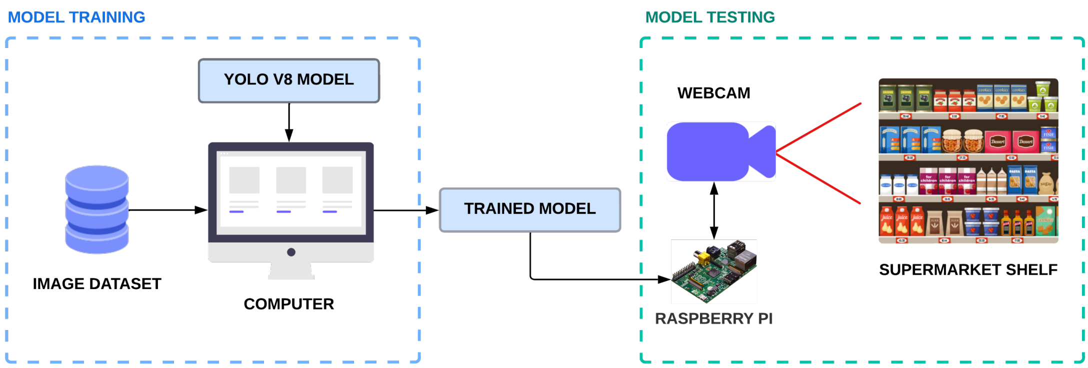

Figure 1. System Architecture: Visualizing the connections of components during model training and testing phases.

3.1. Hardware

For many machine-learning projects, the key to achieve high performance and robustness lies in training the model, and often necessitates substantial resources, making training on low-resource devices impractical. To address the challenge of training the model, it was used a high-performance PC setup fitted for machine-learning tasks, equipped with an AMD CPU Ryzen 9 5950X 16-Core and 128 GB RAM. Additionally, the machine was equipped with an NVIDIA GeForce RTX 3090 GPU, which included 24 GB of VRAM. The training process was exclusively performed by the GPU. For data storage, it was used a Kingston KC2500 2 TB SSD.

Prediction tasks are much less computationally demanding compared to training. Therefore, to test the model in real-time, it was used a SBC called Raspberry Pi 4 model B with 8 GB RAM connected to a webcam Logitech C920 PRO 15mp/1080p. The object photographed for the study was a customized MDF gondola with four shelves that replicates the racks located in supermarkets and grocery stores.

3.2. Software

On the software front, the PC was equipped with a Windows 11 Education operating system. The computer vision model was developed using Python programming language, version 3.10.11. The neural network was trained and tested using Ultralytics version 8.0.20 as the primary library, which is built on top of the PyTorch framework and incorporates the OpenCV library. The YOLOv8 model was directly imported from the Ultralytics library. The data used to train the model was gathered from a dataset named "Supermarket Empty Shelf Detector Computer Vision Project"[FYP 2023], which includes 497 images of supermarket shelves with empty spaces labeled accordingly, available online at Roboflow.

The training was conducted using the YOLOv8s model, which is a lighter version more appropriate for real-time tasks. Some important parameters that need to be configured to dictate the behavior of the model are batch size, number of epochs, and image size. Batch size refers to the number of samples that are processed at once during training. An epoch is one iteration of the training process, the dataset is split into batches following the batch size parameter, once all the batches pass through the model, it constitutes one epoch. Finally, the image size is the resolution for which the images are going to be resized during the training process.

This setup was managed by a Conda environment, to isolate dependencies from the PC environment configurations, along with a Jupyter Notebook to run the code locally. The Raspberry Pi used to embed the trained model was equipped with a Raspberry Pi OS, a Debian-based Linux distribution, and also Thonny, a Python lightweight IDE used to run the script responsible for executing the practical experiment.

4. Experiments and Results

4.1. Training Setup

Once all the required software and hardware were set up, the training was conducted. First, the dataset was split into training and testing subsets with a 70%-30% ratio, which are the default values for this dataset on Roboflow. An example of image found inside the dataset can be seen in Figure 2. Squares with yellow edges represent empty spaces.

To configure the YOLOv8 model, some values for the setup parameters were defined. First, the batch size used was 16 which is the default one for the model. Second, the image size was set to 640 which means that all the dataset images are going to be resized to a 640×640 pixels resolution. While this image size is the default value, it's worth noting that if the detected object has a large amount of details it is recommended to use a higher resolution, and if the model has fewer details and the training seems to be slow, it's recommended to use a lower resolution. Once the "empty space" object is not very detailed and the training process was swift, there was no need to change the image size for this case. Finally, it was chosen a value of 10,000 epochs to run the training process, this is a considerably large number of epochs, used in this case to make sure the optimal result would be found.

4.2. Training Results

As mentioned in section 4.1, the YOLO model was trained for 10,000 epochs. But after approximately 6,270 epochs, the precision started to drop, as illustrated by Figure 3, indicating overfitting, which occurs when a model becomes excessively tailored to the training data, losing its ability to generalize effectively to new, unseen data [Ying 2019].

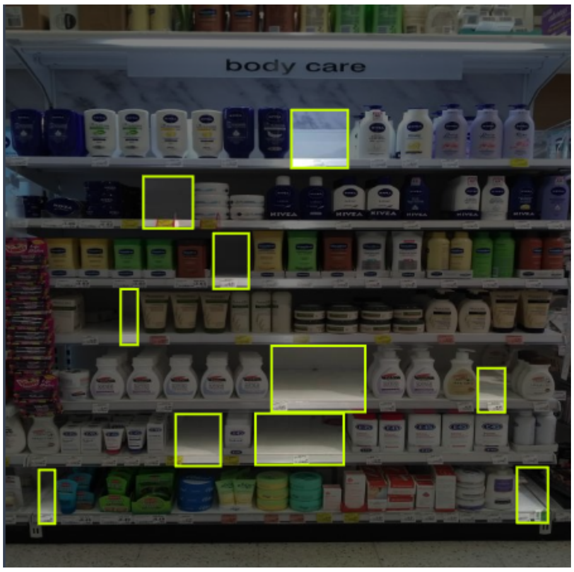

Figure 2. Dataset image example [FYP 2023].

Figure 3 shows the result of our training, showcasing loss and precision scores. The initial six images, arranged on the left, are related to loss metrics, which are expected to decrease during training. The top row illustrates the model's response to the training data, while the second row represents its performance on the validation data.

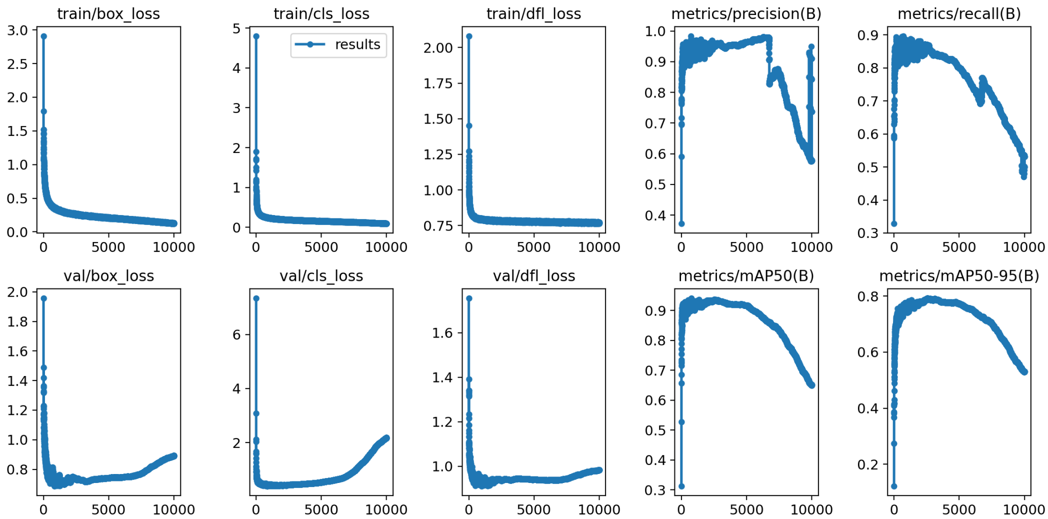

Figure 3. Training and validation performance metrics of 10,000 epochs, displaying losses, precision, recall and mAP scores.

Within the context of loss metrics, *box_loss* means the model's proficiency in identifying bounding boxes, *cls_loss* measures its accuracy in classifying objects correctly, and *dfl_loss* (Dual Focal Loss) addresses class imbalance, although it is less relevant in our case as our model deals with a single class.

Moving to the four graphs on the right side, these metrics are associated with the model's accuracy, employing various expressions for calculation. The precision graphic represents the model's capability to make correct positive predictions among all positive predictions, while the recall graphic illustrate the model effectiveness in capturing actual positive instances. Additionally, the mAP50 and mAP50-95 evaluate the model performance considering different intersections over union thresholds.

When training a model, YOLO stores two files, one called *last.pt* and another one called *best.pt*. The *best.pt* file stores the model weights with the highest validation precision. We used this file to conduct our tests. By limiting the training to 6,270 epochs, which is approximately where the *best.pt* model checkpoint was saved, we observed the convergence of the network and the evolution of its performance.

Figure 4 illustrates the training performance results as we reached the 6,270 epoch milestone. It clearly shows how the precision metric has stabilized during the training and now can be used to conduct real-life experiments.

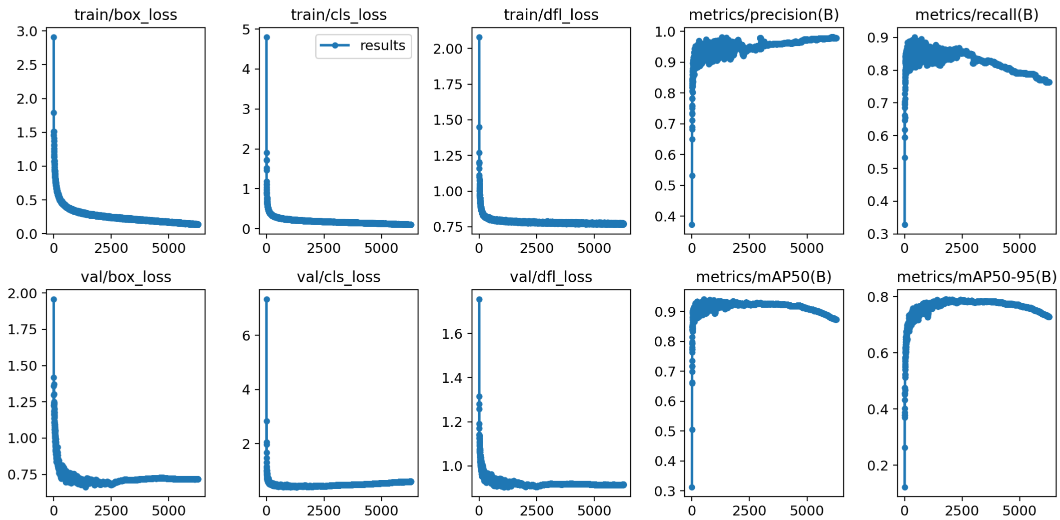

Figure 4. Training and validation performance metrics of 6,270 epochs, displaying losses, precision, recall and mAP scores.

4.3. Experiments on the Embedded System

To test our model, we set up a gondola with various products to simulate a real store situation. Then, we proceeded taking products off the shelf to see if the model can recognize the gaps created in different setups. For these tests, we used a custom-made MDF gondola.

The model runs on a Raspberry Pi connected to a USB webcam for real-time image of the gondola. Diverse product configurations were arranged to generate detection outcomes, enabling an evaluation of the model's performance across various scenarios.

For each experiment, there is an image with the detection result, which consists of a red rectangle with the word “empty” followed by a number from 0 to 1 which respectively represent the empty space detected and the confidence that the model has for that specific detection.

The first experiment conducted, depicted by Figure 5, consisted of all four shelves full of items. As expected, the model detected no empty spaces. The next experiments aim to explore the model detection accuracy in different scenarios.

Figure 5. Shelves full of items.

After analyzing the model's behavior for full shelves, the subsequent experiments were conducted using medium-sized empty spaces with different arrangements. Specifically, the first arrangement, illustrated by Figure 6, featured empty spaces on both middle shelves.

Figure 6. Medium-sized empty spaces on middle shelves.

The second arrangement, as shown by Figure 7, comprised empty spaces across all four shelves. The third arrangement, as presented by Figure 8, was executed with empty spaces on one of the middle shelves and the bottom one. The model was able to detect all the empty spaces but with less confidence in the shelf on the top.

Figure 7. Medium-sized empty spaces on all shelves.

Next, other experiment was conducted with large-sized empty spaces to further investigate the model's behavior. This experiment has empty spaces in the middle and bottom shelves, as shown by Figure 9. Last but not least, an experiment mixing small and medium-sized empty spaces was performed, as depicted by Figure 10.

Figure 8. Medium-sized empty spaces on bottom and middle shelves.

Figure 9. Large-sized empty spaces on middle and bottom shelves.

During these experiments, it became evident that the model encountered more challenges in accurately identifying large empty spaces, as revealed by the inconsistent detection observed during the real-time experiment and the lower confidence numbers in the images. It occurred because the dataset contained a limited number of training images for empty spaces with larger sizes.

In all the experiments performed, the empty spaces were consistently detected with a high level of confidence, emphasizing the model's stronger generalization ability for smaller spaces in contrast to larger ones. In future works, images of larger spaces will be added to the dataset in order to verify if the model presents more accuracy in detecting them.

Figure 10. Small-sized and Medium-sized empty spaces on middle and bottom shelves.

5. Conclusion

This study demonstrated the potential use of computer vision as a means of assisting in the control and management of a retail shelf inventory. The training process of the model in question can be customized for various types of datasets, enabling applications not only for this scenario but for any other involving object detection or their absence.

We present a practical experiment in computer vision, detecting empty spaces on gondola shelves, and showcasing how a model like YOLO can elevate retail inventory management toward higher levels of automation. Our research successfully demonstrated the effectiveness of detecting empty spaces on supermarket shelves using the YOLOv8 model.

Using a model trained at its best training checkpoint, we successfully validated our model in a real-world scenario, employing a customized MDF shelf with items across various test scenarios. This model can be useful for assisting in supermarket and grocery inventory management, directly influencing retailers' logistics, and indirectly impacting the experiences of customers and stockers.

It was observed that the model experiences a performance increase as the training progresses. However, upon training for 10,000 epochs, overfitting became noticeable. This indicates that such improvement has a limit, measurable in the number of epochs. In this specific experiment, this limit was approximately 6,270 epochs, considering the precision metric as the stopping point.

During the experiments, some challenges were encountered, such as the influence of lighting and the color of the background of the shelves on the detection. With the original MDF background, the model had a much lower efficiency, which was caused by the nature of the images of the dataset, in which most of the empty spaces were blacker due to the depth of the shelves. Therefore, it was necessary to darken the background of the gondola to simulate the shadow that is formed in shelves with a greater depth.

Another significant challenge encountered during the experiment was the impact of camera angles on detection accuracy. It became evident that centering the camera as closely as possible in relation to the gondola yielded superior results, whereas alternative angles noticeably affected the model's detection capabilities.

As a recommendation for future work, it may be possible to improve the model by increasing the number of images on the training dataset. Additionally, the model could be trained to identify and classify the products on the shelf, creating a system capable of inferring the number of items on the shelf. An alert system that notifies store employees when products are running low on shelves could also be developed. This system would be based on categorizing product types within different sections and identifying regions where product shortages are detected by the developed system.

Moreover, we aim to place the shelf in a real consumer environment to observe its behavior. Finally, there is a suggestion to advance the prototype into a Digital Twin, aiming not only to conduct detection processes but also to extract real-time shelf consumption data. It would facilitate real-time metrics collection and the evaluation of customer behavior in a cyber-physical context, thereby aiding employees and management in the decision-making.

References

FYP (2023). Supermarket empty shelf detector computer vision project. https://universe.roboflow.com/fyp-ormnr/supermarket-empty-shelf-detector. accessed: 2023-10-12.

Ghosh, A., Chakraborty, D., and Law, A. (2018). Artificial intelligence in internet of things. *CAAI Transactions on Intelligence Technology*, 3(4):208–218.

Gross, R. (2019). How the amazon go store's ai works. https://towardsdatascience.com/how-the-amazon-go-store-works-a-deep-dive-3fde9d9939e9. accessed: 2023-09-24.

Hwangbo, H., Kim, Y. S., and Cha, K. J. (2017). Use of the smart store for persuasive marketing and immersive customer experiences: A case study of korean apparel enterprise. *Mobile Information Systems*, 2017:4738340.

Kalahiki, C. B. (2020). Computer vision for inventory management. *Master's Thesis - College of Engineering and Science Louisiana Tech University, Ruston, LA*.

Sanchez-Ruiz, L., Blanco, B., and Kyguolienė, A. (2018). A theoretical overview of the stockout problem in retail: from causes to consequences. *Management of Organizations: Systematic Research*, 79(1):103–116.

Savit, A. and Damor, A. (2023). Revolutionizing retail stores with computer vision and edge ai: A novel shelf management system. In *2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, pages 69–74. IEEE.

Shanmugamani, R. (2018). *Deep Learning for Computer Vision*. Packt.

Szeliski, R. (2022). *Computer vision: algorithms and applications*. Springer Nature.

Times, T. N. Y. (2018). Retailers race against amazon to automate stores — the new york times.

Wankhede, K., Wukkadada, B., and Nadar, V. (2018). Just walk-out technology and its challenges: A case of amazon go. 1:254–257.

Ying, X. (2019). An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, 1168(2):022022.