Human Object Ownership Tracking in Autonomous Retail

Submitted in partial fulfillment for the requirements for
the degreee of
Doctor of Philosophy
in
Electrical & Computer Engineering

João Diogo Lisboa De Menezes Falcão
B.S., Electrical & Computer Engineering, Instituto Superior Técnico
M.Eng., Electrical & Computer Engineering, Cornell University
M.S., Electrical & Computer Engineering, Carnegie Mellon University

Carnegie Mellon University
Pittsburgh, PA

December 2021

Abstract

In retail 85% of sales occur in physical stores. In the U.S. alone, people spend roughly 37 billion hours each year waiting in line in physical stores. This leads to 4.4 billion potential work productive hours lost or comparatively 4.4 billion hours of leisure, rest or time with your loved ones lost. Autonomous stores can remove customer waiting time by providing a receipt without the need for scanning the items. Autonomous stores in the grocery sector can further serve locations, so called 'food deserts', that would otherwise not have access to grocery stores. This is done by reducing the physical size of these stores while still maintaining the commercial opportunity through automation.

Understanding physical object ownership transfer is a key element of physical commerce, and is central to the understanding of when and how people grab products off of a store. For a machine to understand this it not only needs to sense and identify individual objects in a constraint physical space but also how their ownership changes over time. Humans are often at the center of such transfers and detecting and characterizing human object ownership over time opens the possibility for multiple applications to improve through automation. Applications such as inventory counting, surveillance, supply chain management, inventory management, and checkout-free retail all benefit from the ability to understand human object ownership over time by allowing automatic decisions to be made. In this thesis I will use the autonomous retail as a guiding application to demonstrate the applicability of this approach.

Approaches such as using manual intervention (e.g. cashiers at a supermarket), on-object sensing (e.g. RFID tags), contextual modeling through vision only or combining computer vision with other sensors can be used in applications that require the understanding of object ownership transfer, these however require directly, or indirectly, a large amount of human labor making it impractical to scale in real-world applications. Furthermore, the general low accuracy and throughput of these approaches further hinders their applicability in a broader real-world context.

This work explores the automatic tracking of human object ownership over time. In this thesis I propose a framework for detecting and characterizing human object ownership and introduce a method where physical context of the application is combined with the available sensing modalities. By modeling the static physical context (e.g. location, appearance, weight, volume), dynamic physical context (e.g. human motion, temporal-spatial proximity) and the physical relations (e.g. historical ownership, relative motion) between objects and people, sensing data can be combined with such context to enhance the detection and tracking of object ownership changes.

Resulting from the combination of physical context modeling with the multiple sensing modalities, this approach significantly reduces the computation requirements while maintaining high accuracies which in turn enables the scale requirements of a real-world deployment. This method is validated across several retail applications, which due to its trading nature include a rich set of annotated ownership transactions. In the context of inventory monitoring our tracking approach achieved 92.6% item identification accuracy, a 2x reduction in error compared to the 86% accuracy reported for self-checkout stations. For autonomous retail stores, we maintained an average of up to 96.4% receipt accuracy over 1 year of operation, across over 65,000 transactions with a total of 1653 total different products being sold.

Acknowledgments

The journey that culminated in writing this thesis was filled with twists and turns and I am very thankful that at everyone of those turns there were interesting people and ideas that I've got to meet an explore. I could have not asked for a more enriching experience.

First of all, I would like to thank my advisor, Pei Zhang, who taught me many things, but most importantly that science goes beyond universities and institutions names it is a way of thinking, acting and sharing knowledge. I feel honored to have met you and have been a part of the group of people of whom you share ideas through presentations and conversations, making my Fridays much more enriching. You have helped me understand the difference between engineering and research and how to leverage both in my career. Furthermore, I appreciate you giving me the chance to get back into the program and continue my academic pursuits. Thanks for making these final years possible.

Thank you to my defense committee. Thank you Hae Young Noh, for showing through example that kindness and a sharp mind can co-exist, and for your availability to help whenever it was needed, you bring a light to this world that I found unique. Thank you John Shen and Orathai Sangpetch for all the feedback and discussions to make this work more clear and impactful.

And of course, thank you Carlos Ruiz, your friendship and contagious happy smile have made it possible to survive the hard times, and to put into perspective that even thought it might not look like it, in the moment, things do get better.

Thank you Adeola Bannis, Amelie Bonde, Jonathon Fagert I couldn't have asked for better labmates and support group. I can't thank you enough for all the research discussions, papers feedback and crazy drawings on Friday.

Thank you Martin Griss, Hakan Erdogmus, Bob Iannucci, João Paulo, for being my advisors throughout this journey, providing me academic and industry advice and technical help both inside and outside the Silicon Valley classrooms.

Thank you Steve Gu for allowing me to pursue this effort while going through the challenging and uncertain ride that is the life cycle of a startup, always incentivizing more collaboration and contribution to the academic world. Thank you Kees Vissers and Kristof Denolf for providing me the experience of working with you and giving me great advice and thank you for your friendship.

Thank you Mostafa Mirshekari, Shijia Pan, Ankit Shrisvastava for your helpful feedback despite the distance and for all the good times we've had remotely and in-person. Thank you Ervin Teng for the fun on campus and writing in those early days building boards and preparing classes together.

Last but not least, I would like to thank my wife, who supported me through the crazy hours juggling writing papers and working in a startup. For all that you've abdicated by proximity for me to accomplish this.

Thank you to my family and friends in Portugal that have always been proud of me and loved and supported despite the distance. I am very fortunate to have you all in my life.

To everyone who I had the opportunity to meet and was fortunate to interact with during this path, I thank you for being a part of this journey.

All the work presented in this thesis has been partially supported by AiFi Inc. and ThaiBev. The views and conclusions contained here are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed implied, of CMU, or the U.S. Government or any of its agencies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Autonomous shopping systems, or more generally Autonomous Retail, refers to technology where parts of the shopping experience are automated. These systems are designed to make selling and buying consumer goods more efficiently. Brick-and-mortar stores still represent more than 85% of shopping transactions and since the pandemic–COVID-19– autonomous retail in physical stores has seen a bigger rate of adoption (whether on payment, queuing, or delivery). Automation at this level allows stores to operate in locations where it wasn't previously commercially viable, such as food deserts, or smaller locations closer to consumers, increasing consumers access to food and fundamental goods. Other benefits stem from being able to save people's time from waiting lines. Americans spend 37 billion hours waiting in line each year.

However, in order to achieve these benefits or level of automation a retail system needs to understand the shoppers behavior, similar to how a cashier does. More importantly the moments where ownership of the products is transferred between the store and their shoppers, or more simply the system needs to understand when a shopper grabs a product and leaves with it. Given how normally people behave when interacting with objects **Human Object Ownership Tracking** is generally challenging for single modal sensing systems to achieve. Single modal sensing systems are subject to occlusions and ambiguity, sensing noise present, or introduced, in the system, or are simply not accurate enough to understand everything in a real-world application. On the other hand multi-modal sensing systems perform better by having devices collaborate to gain a better understanding of the surroundings, these however still need to understand the physical context of the environment in order to accurately operate in the real-world.

This raises the **Human Object Ownership Tracking through Physical Context Modeling using Multi-Modality Sensing** problem: in order to recognize and track, in space and time, that a object has been transferred between *owners*, multiple sensing devices need to share a common physical understanding and context of the world to associate each sensed event to the correct object and correct person.

This common physical context is typically done through data driven methods that implicitly declare the relationship between two or more sensing features (e.g. an image of a red basket close to a person with a blue jacket is defined as a employee restocking products in a store, or an image with a person extending their hand is linked to a pressure change sensed on a shelf to define that an object is being picked up). While this can be highly efficient at defining very specific actions and scenarios, it remains a challenge to solve a more general problem such as understanding a transfer of ownership of a product, which can be done under a variety of conditions, in under almost an uncountable number of ways. Defining every possible pair or connection between different sensing modalities across large enough datasets is highly labor intensive and has shown impractical for real-world applications. Therefore, modeling the physical context and leveraging the physical properties and knowledge of the environment, and objects, to define explicitly the relationship between people and objects across different sensing modalities is extremely important as it enables applications to operate in the real-world.

Different approaches have been proposed to solve the human object ownership transfer. They can be broadly categorized into 2 independent axis: *invasive vs. noninvasive* and *supervised vs. unsupervised*. Invasive approaches require direct interference with the person interacting with the objects. For instance solutions that require a person to carry a device and register every object picked (i.e. Scan & Go), or that instead require that a final validation and recount of all objects picked at the end (i.e. Self Checkout) are considered invasive, because they require the person to change their behavior for the purpose of tracking their otherwise normal interaction with objects. While these are generally easy to deploy they tend to be inaccurate given the inconsistency of the person registering the objects' interactions. On the other hand, noninvasive approaches trade-in deployment ease and operation labor for the lack of explicit behavioral interference, making them more accurate and consequently more broadly adopted.

Supervised approaches require a human observer that does not participate in the event to label the object transfer, this can be done either in-situ or remotely (i.e. Amazon Go). Unsupervised solutions while less labor intensive are perceived less accurate. Therefore finding better ways to improve accuracy under noninvasive and unsupervised conditions is crucial to enable wide adoption of this technology.

1.1 Thesis Statement

The statement presented in this thesis is as follows: **tracking of objects' ownership through multi-modal sensing devices is enabled by modeling the physical context of those objects' and surrounding people**. This work focuses in the retail environment, where sensors can be placed in the infrastructure, such as cameras in the ceiling or sensors on the surfaces (e.g. floor, shelves, entrance, etc.). Furthermore I explore the automatic tracking of human object ownership over time and space. For example, when a shopper enters a grocery stores and picks a drink from a shelf, this can be sensed by weight sensors on the shelf as well as seen by cameras nearby. In this case, continuous time-series event detection equations can be combined with motion estimation through both the sensor and camera to understand this behavior. Due to the vast number of objects (products) and potential people present in the store, these systems require a common physical context to be able to localize this event in order to attribute this ownership to the right person. Similarly knowing the physical characteristics of the object (weight, volume, appearance) allows for the correct object to be accurately detected across its close-by neighbours. Other context such as historical ownership transfer events or relative human motion and proximity allow for further disambiguation of who, or what is being interacted with by eliminating the non physically possible scenarios.

In order to enable the wide adoption of autonomous retail through physical context modeling using multi-modality sensing, the approach needs to be noninvasive, this means that the people are not required to change their behavior to register their interactions, as this has shown impractical and inaccurate. In addition the identification and tracking of the ownership transfer must be done without human intervention, as this leads to increased human labor impeding wider adoption. Furthermore the system is required to provide an accurate result, such that retail operations become successful.

1.2 Research Questions

Achieving accurate tracking of the transfer of objects' ownership through information collected by multiple sensing devices across an unconstrained set of human behavior comes with several challenges. These challenges can be grouped into 3 main topics: shared multi-modal physical context definition (common physical environment definition), human object ownership interaction modeling (ownership transition state definition), human object spatial-temporal relationship modeling (human to object ownership transfer matching). The following subsections detail each of these challenges and describe each of research questions addressed in this thesis.

1.2.1 Multi-modal individual object ownership transfer

Multi-modal systems' imply that sensing devices observe different physical representations of the same event. In order to combine information obtained by these sensing devices it is necessary to model the physical context in which these devices operate. Furthermore, an object ownership transfer is defined by a combination of multiple pieces of information (who, what, where and when), it becomes important to correctly combine the partial information obtained by each modality. However, given the unconstrained human behavior many times occlusion of individual modalities can occur, and the object ownership information has to be inferred from the available sensing information. The main research questions that consider these issues are the following:

- How to model the physical shared environment between different sensing modalities?
- How to fuse partial information from each sensing device into a complete object ownership transfer detection?
- How to leverage different modalities information under occlusion or ambiguous estimates when combining multi-modal object ownership transfer predictions?

1.2.2 Object ownership transfers between humans

In order to correctly track object ownership transfers a system needs to understand when and where such transitions occur. When the available set of sensing devices is exposed to the human interaction leading to a object ownership transfer, it is possible to model the trigger that defines such transition. However, many times there are occlusions, or simply there is a lack of one sensing modality in a particular ownership transfer and the event has to be inferred from combining contextual information. Correctly predicting the object ownership transfer under these conditions involves answering the following research questions:

- How to model the human object relationships that define an object ownership transfer?
- How to account for partial lack of line of sigh when detecting a object ownership transfer?
- How does a object ownership transfer detection impact historical object ownership transfer information?

1.2.3 Simultaneous object ownership transfers with multiple humans

In most real-world shopping applications multiple humans are part of the equation. Accurately tracking the history of object ownership changes requires to disambiguate between several people and several objects that interact simultaneously withing the same system. It becomes important to model the interactions between different objects and different people. Therefore, the research questions related to simultaneous interactions with multiple humans are the following:

- How to model the physical interactions–e.g. relative motion, historical ownership–between different objects and different people?
- How to disambiguate between multiple humans and multiple objects when concurrent object ownership transfers are detected?

1.3 Contributions

This thesis presents a methodology where physical context of the environment is combined with multiple sensing modalities in order to enhance the detection and tracking of objects' ownership changes over time. In this section I provide a summary of the three main contributions of this work.

1.3.1 Using objects physical context to enable Multi-modal individual object ownership transfer

The physical characteristics of the objects in the scene are great individual identifiers of the objects, allowing, in conjunction with the sensors, to make statistical inferences about which objects are interacted with. The location, position and size of the objects along with their sensed appearance –visual in case of cameras and weight in case of weight sensors– enable an adaptive sensor fusion approach, where similarities in the sensing domain (occlusions or similar weights) can be atoned by the use of the model. i.e. a coca-cola and a coca-cola diet are being interacted with, their location model quickly resolve any confusion between the objects.

Leveraging the explicit physical context (object and owner: location, appearance, weight, volume) to detect individual object ownership transfers in our experiments has achieved a 92.6% item identification accuracy in a noninvasive and unsupervised manner. These experiments were done across 400 shopping events, including 85 items disposed to mimic a fixture of a 7-Eleven convenient store, instrumented with weight sensors on the shelves and 4 cameras in the ceiling. The results were close to $2x$ reduction in error when compared to the 86% accuracy reported for self-checkout stations.

1.3.2 Modeling human-object physical relationship to enable ownership transition detection

Even with an explicit physical context of the objects present in the environment defined, to correctly detect and identify the transfer of ownership of objects it is important to understand when and where such transitions occur. Typically these transitions, in stores, occur at the shelf (selling space) area, allowing for proper instrumentation of these locations to be done. However given the unconstrained behavior of people, ownership transition can occur between people at any time and place making partial sensor information (e.g. force sensor in shelf) irrelevant.

I tackle the detection of the transition moment between owners by focusing on the human behavior through motion, which can be estimated through vision. Utilizing prior knowledge of what a human looks like, traditional object detection and tracking techniques can find the position of each human from vision. Furthermore, by modeling the distances and relative motion of all people with each other and objects of interest we can reduce the search space. Estimating joint velocity and proximity with zones of interest allows for triggering such interactions even in the absence, or occlusion, of direct sensing modalities.

Finally when faced with dynamic objects, objects that are in motion (carried by someone) before the transition occurs, we are faced with an added challenge of understanding when and where this transition has occurred. This can be simplified by modeling the physical relationship between humans and objects. Understanding how these two move together allows for indirect inference, this means, that we can estimate the location of an object if we observe the motion of its carrier. i.e. We can estimate the location of a bag of chips that sits on a cart, even if there are occlusions by leveraging the knowledge that after the bag is placed in the cart, it will move in the same trajectory as the cart (which is not occluded). This type of modeling permits the transfer of object ownership and allows indirect sensing improving the human object ownership interaction detection.

1.3.3 Matching multiple human object ownership transfers by modeling spatial-temporal interactions between humans and objects

Similar to detecting the moment when a transition of ownership occurs, matching the correct person to the transition in the presence of simultaneous interactions with multiple people can benefit from the modeling of human motion. For instance, inside a store, when a group of friends are deciding which product to buy, they may interact closely to each other and with very close-by products. Exploiting the physical dynamics of a human interacting with and object, and creating a model of how humans interact with objects (i.e. A person picking a mug tends to use their hands to hold the wing of the mug.) can allow the correct matching between different people and products' ownership. In this case the typical location in the object as well as the motion of the human when interacting with the object define this model. Such model improves the spatial-temporal confusion that might occur when the system is faced with multi-humans interacting simultaneously with multiple objects, each person will interact slightly differently providing information about which object they are interacting with.

This approach was evaluated in an live operating convenience store covering 800 square feet with 1653 distinct products, over 65,000 ownership transfer events and more than 20,000 available items. Over the course of 13 months of operation achieving a receipt daily accuracy of up to 96.4%. The multiple people matching approach yielded over 2x error reduction in cases where more than 3 people were interacting close-by (within a radius of 2 meters).

1.4 Thesis Organization

This thesis is organized as follows: Chapter 1 introduces the topic of 'Human Object Ownership Tracking' along with the challenges and research questions therein. Next, Chapter 2 presents some background around the autonomous retail as well as highlights limitations/gaps in existing approaches. Then, Chapter 3 describes the proposed methodology and overall system architecture to tackle the human object ownership transfers in autonomous retail. Chapter 5, concentrates on the modeling of the physical context of objects. Chapter 4, addresses the model of physical relationships between humans and objects to enable the detection of ownership transitions, while Chapter 6 focuses on the modeling of physical interactions between humans and objects to improve the matching of simultaneous ownership transfers between multiple humans with multiple objects. Finally, Chapter 7 summarizes the conclusions of this work.

Chapter 2

Background

Physical retail requires understanding of what the customer is intending to purchase in order for a correct sale to occur. Traditionally this happens through a manual cashier system. In other words, an employed person reviews every item for purchase and registers each one manually into a system in order to subsequently charge the customer the correct amount. While this approach works and is broadly adopted, it creates a bottleneck in the physical sales process. This process is inherently slow and generates queues. Furthermore, in order to sustain the economical viability of the physical retail space, the profit of a store has to be sufficient to justify the labor cost at the cashier. In grocery and convenience stores profit proportionally grows with the number of products available for sale, which implies that larger physical spaces unlock more cashiers. Leaving a gap either in smaller retail spaces, or in spaces where the cost of operations surpasses the benefit of operating there. There exists a wide range of prior works that have tackled this issue, and they can be grouped into manual, on object-based sensing and infrastructure-based sensing.

Manual methods of registration of the products to be sold to a customer do not rely on systems' detecting individual object ownership transfers as they occur, but instead require the person shopping to process each individual product through a checkout stage before leaving, such as: giving the products to a cashier [74, 112] or registering themselves the products with a retailer system (self-checkouts) [4]. While these are the most used approaches, they are slow and labor intensive [71]. There are other manual methods such as single queue management [112], which tackles this issue by predicting the shoppers traffic patterns and allocating more cashiers when needed. However this approach is limited by the availability and the on-boarding and off-boarding speed of the cashiers– it is impractical to find a workforce available to work unpredictably for an arbitrary amount of time and then go home. In order to address this, others have used remote cashiers [82, 105]. However, this requires cashiers with remote visibility access to the object ownership transfer occurring in the store. This approach displaces the labor force into more cost effective locations, but it creates a delay in the delivery of the receipt which shoppers who are more budget conscious and that come with the mindset of how much they will spend for their shopping reject. This is driven by the fact that in-store discounts and the "smart-shopper feelings" towards pricing act as a major component of the emotional response affecting shoppers' behavior to favor in-store price confirmation [13, 62, 77]. We therefore need automated ways of tracking object ownership transfers.

On object-based sensing approaches, on the other hand, do not require the customer to go through a labor intensive checkout process. These approaches identify the objects to be purchase by ensuring that each object has a unique method of identification, and that its ownership transfer can be tracked out of the sale space. Such as using RFID tags [60, 84, 88, 125], or individual product slot location (i.e. vending machine, hotel minibar) [64]. While these are highly accurate, they have not shown to be practical in terms of cost, deployment (tagging every single item for sale) or operations (having to position every product for sale in a single constrained slot).

Finally, infrastructure-based systems instrument the physical space of a store in order to keep track of the object ownership transfers in spite of the type and number products in the store. This infrastructure instrumentation can vary from vision only [25, 41, 50, 116, 119] which focuses mainly on the object recognition and misses the objects' motion or its owner, to vision with additional sensors. These combined approaches use devices such as weight sensors [56, 60], vibration sensors for people detection and tracking [67, 68] and inertial wearable sensors [92]. Their main limitation is that they lack a unified view of the physical context and dynamic of the objects and people in the space. With regards to this limitation, this thesis directly addresses this gap by presenting a framework to track object ownership transfers through physical modeling of the context and dynamic of objects and people, providing higher tracking accuracies.

Chapter 3

Methodology

In this Chapter I describe the methodologies applied in this Thesis. The overall methodology can be described as **contextually enhanced multi-modal sensing**, which consists in explicitly describing the physical contextual characteristics of the people, objects and structures of the underlying environment in order to enhance the processing of the information gathered from the multi-modal sensing devices. To demonstrate this methodology I focus on a representative application: Autonomous retail. More specifically tracking objects ownership transfers within a physical store, which benefits from contextually enhanced multi-modal sensing. This methodology is applicable to all the elements in the physical environment of systems that track automatically object ownership transfers. Elements such as, the static physical context (e.g. location, appearance, weight, volume), the dynamic physical context (e.g. human motion, temporal-spatial proximity) and the physical relations (e.g. historical ownership, relative motion) between objects, structures and people. While this work may specifically target autonomous retail, the insights and approaches obtained throughout this thesis can be applied for a broader set of problems related to tracking objects being transferred between people and other objects, in a instrumentable physical space (e.g. Warehouse inventory monitoring).

3.1 Terminology

In this section the terms used throughout this thesis are defined. Human-Object Ownership refers to the relationship of ownership from objects to humans. The types of objects referred by this term are best demonstrated by Figures 3.1 and 3.2. The transfer of ownership occurs the moment the objects starts sharing higher bounds (similar motion, velocity and location) with another owner. In this thesis ownership has no relationship to the financial or legal ownership of objects, but exclusively with its physical characteristics.

In Fig. 3.1 we can see an example of a human driving a forklift and delivering boxes into a warehouse. This figure demonstrates the multiple elements that constitute a Human-Object Ownership transfer. In this case we can see a *Single Human* carrying a *Single Object*– multiple items of the same object (boxes).

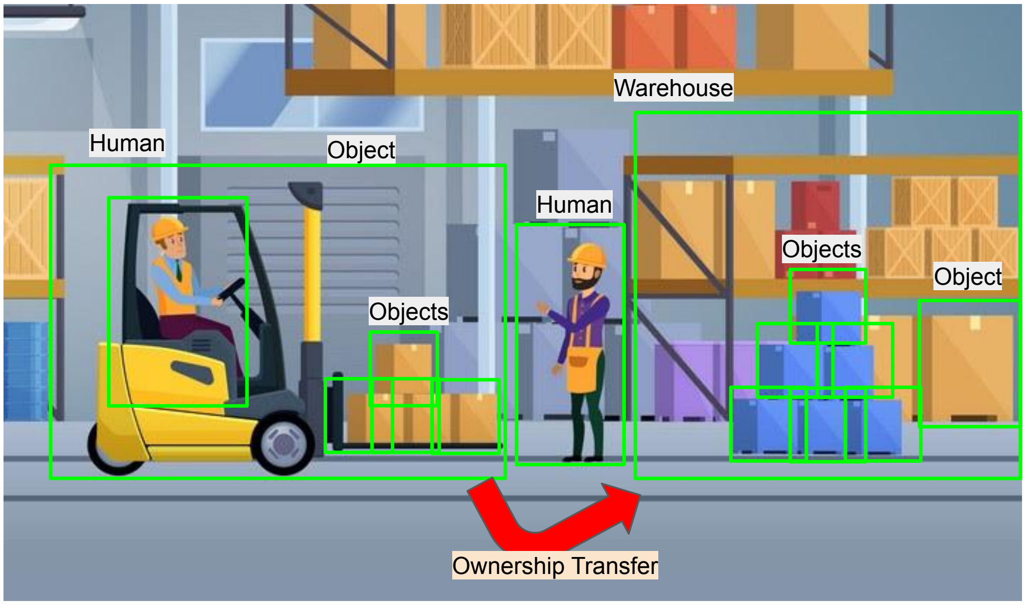


Figure 3.1: Representation of a warehouse with a single type of object (boxes) being carried by a single person. Concepts of interest are highlighted in green: Human, Object and Warehouse. The red arrow represents the Ownership Transfer that occurs when the boxes are unloaded from the forklift and left inside the warehouse.

Figure 3.2 shows a scenario with multiple examples of Ownership Transfers inside a store. Here we can see highlighted ownership transfers where a *single person* picks up *multiple different objects* (in purple) and a second example where *multiple people* pick up *multiple different objects* (in blue).

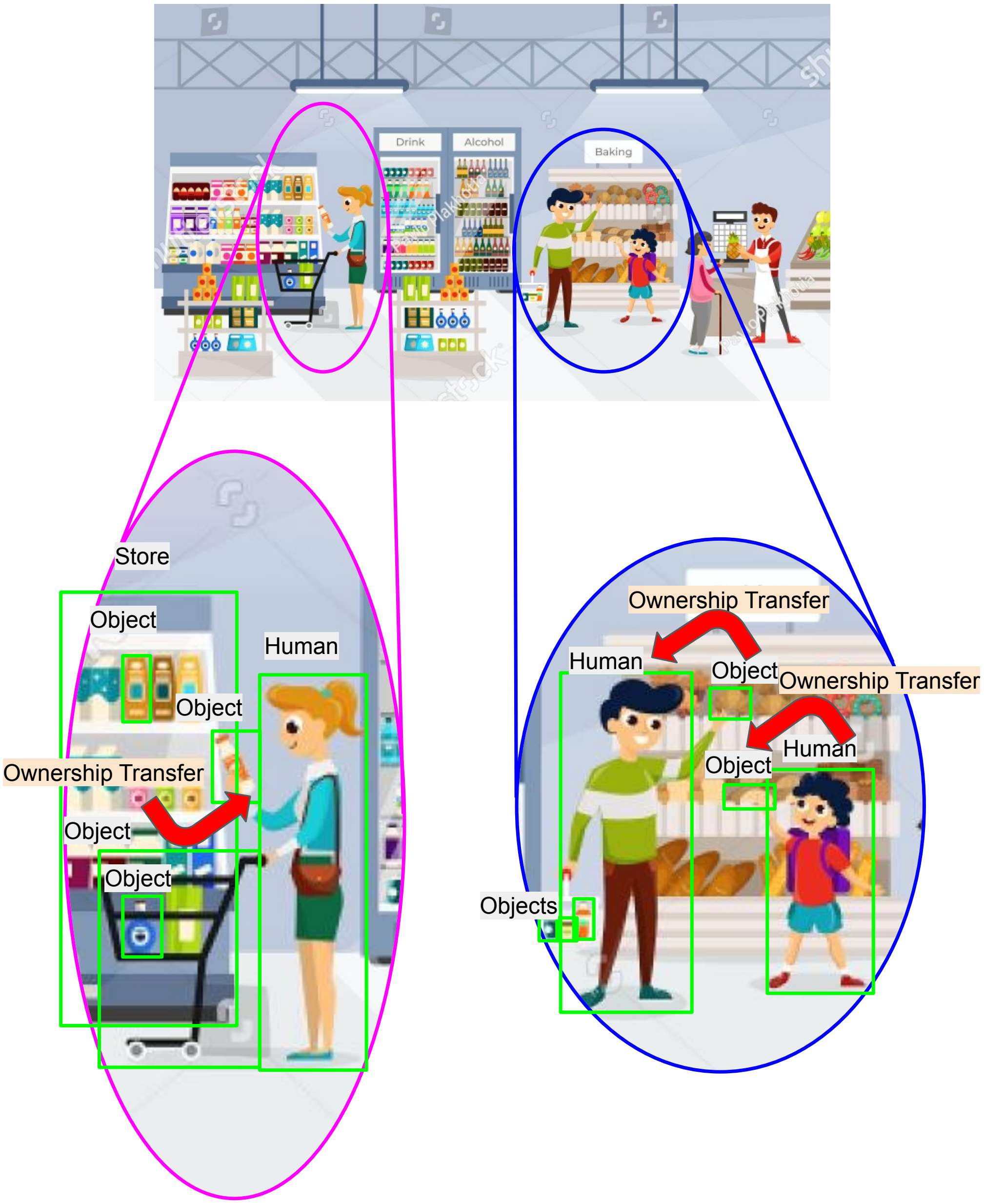


Figure 3.2: Representation of a store scenario with multiple examples of ownership transfers. Concepts of interest are highlighted in green: Human, Object and Store. **Purple** highlights an Ownership Transfer where a single person picks up multiple different objects. **Blue** highlights Ownership Transfers where multiple humans pick up simultaneously multiple different objects. The red arrow represents the Ownership Transfers that occurs in theses two scenarios.

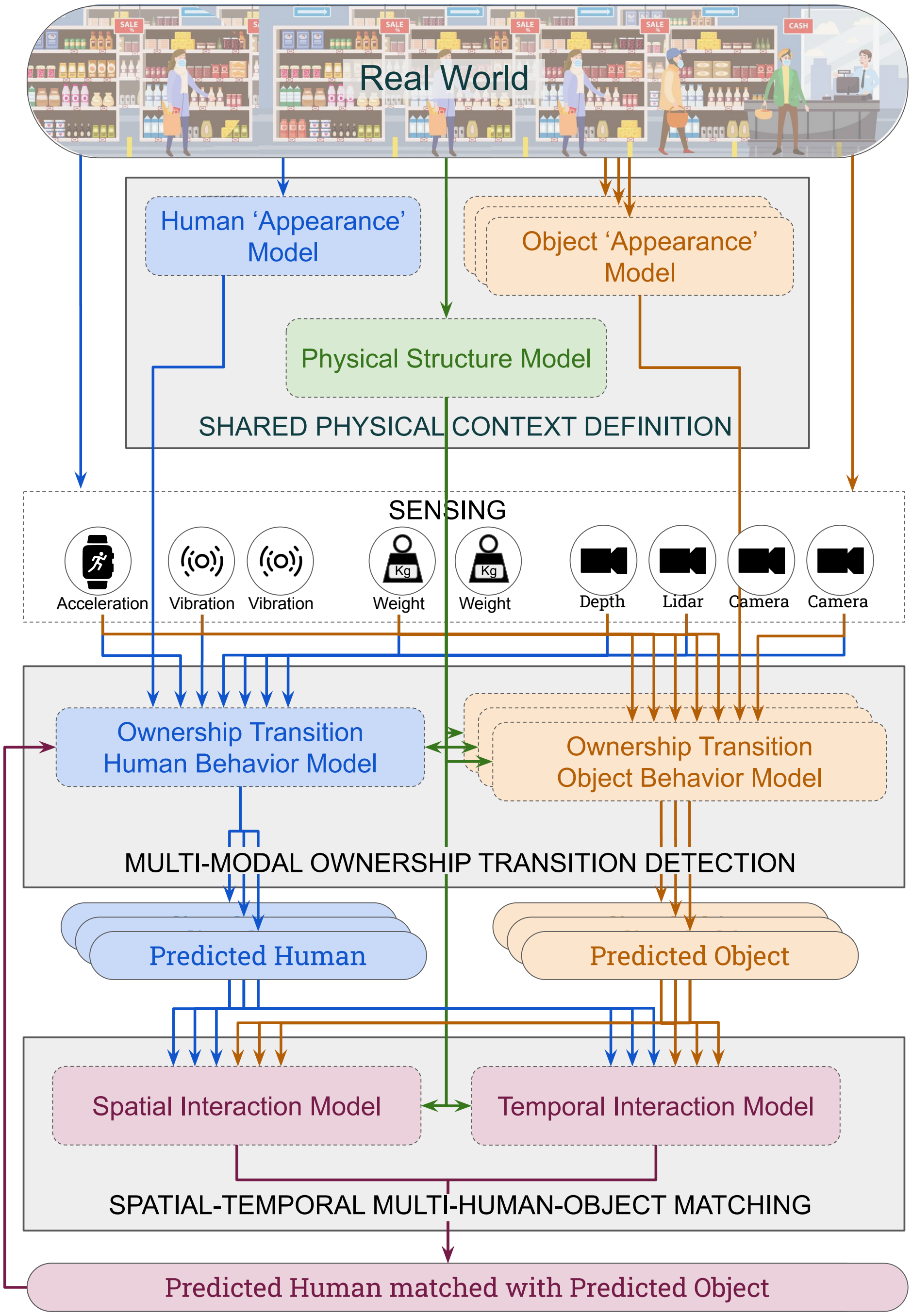
3.2 'Human-Object Ownership Tracking' Framework

Tracking objects that are being transferred between people requires an understanding of objects' location relative to the people. This is traditionally done implicitly and within a particular sensing modality. For example, when a camera is observing a scene and detects a person passing his friend a bottle of water it does so by observing the pixel distance between the identified bottle and the detected people present.

This however could be enhanced by the explicit knowledge of how big a bottle of water is, that it needs to be supported by something (i.e hand) or even where the camera in physically located. Without this information, perspective mistakes and ambiguous results can be obtained. Figure 3.3 shows the 'Human Object Ownership Transfer' framework used in this thesis to address these types of issues. The following sections provide an overview of the framework's components and provide insight into what contextual information is used to define the models at each component.

3.3 Shared physical context definition

In attempting to track objects a key step to achieve accurate results is defining a model of the objects in question. Typically that means either creating or generating a dataset of those objects where they are exposed to the particular sensing modality and labelled in that feature space. For example, in a visual dataset created for tracking apples you would see several images of several types of apples, taken from various different angles with several different backgrounds. This is done in the attempt to generalize and extract the meaningful features of what an apple is. However, the apple's physical context is mostly missed. In other words, this model would benefit from understanding a typical apple volume in relationship to other objects, or that apples cannot fly (without being thrown) and require a holding element (person, floor, surface). Or even more specifically, that in a particular application such as a grocery store, most of the time apple's will either be in the fruit section, someone's hand or in their basket (object contextual location). With this information, better more effective and accurate systems can be designed.



Real World
Human 'Appearance' Model
Object 'Appearance' Model
Physical Structure Model
SHARED PHYSICAL CONTEXT DEFINITION
SENSING
Acceleration
Vibration
Vibration
Weight
Weight
Depth
Lidar
Camera
Camera
Ownership Transition Human Behavior Model
Ownership Transition Object Behavior Model
MULTI-MODAL OWNERSHIP TRANSITION DETECTION
Predicted Human
Predicted Object
Spatial Interaction Model
Temporal Interaction Model
SPATIAL-TEMPORAL MULTI-HUMAN-OBJECT MATCHING
Predicted Human matched with Predicted Object

Figure 3.3: 'Human Object Ownership Tracking' Framework. Physical context of the structure, humans and objects is combined with the available sensing information to define the behavior models of humans and objects, and finally their respective interaction models.

In this framework the first stage focuses in defining an explicit shared physical context across all sensing modality. The insight is that regardless of the modalities in question there is a common physical reality that binds all sensors together, and that modeling that bound allows for impossible hypotheses to be pruned out at the moment of detection and tracking. In particular I propose to focus on the people, physical structures (floor, columns, walls, etc.), and objects' of interest contextual location and 'appearance' in the available sensing modalities (e.g. shape, color, weight, vibration response, volume). This allows multi-modal systems' to combine sensing information an context for accurate tracking of objects [28, 89], while reducing the required computation by focusing the attention into the areas that make physical sense [91]. Chapter 5 explains in further detail this technique and the process of designing each model and explicitly sharing it across sensing modalities.

3.4 Multi-modal ownership transition detection

After defining an explicit shared physical context, sensing modalities can then have a common context to collaborate when detecting objects and people. However a successful object transfer relies not only on the ability to sense the objects and people, but also on triggering when a *tranfer* occurs. Human ownership of an object is defined as the moment when the motion features of an object directly, or indirectly, reflect the motion features of a human. This means, that and object sharing the same trajectory, relative motion and location with a person is owned by them. Consequently a transfer of ownership occurs when these motion features similarities switch between owners. This change is sensed differently between different modalities (weight change event, hand acceleration towards object, RF signal strength change). But similarly to the physical 'appearance´, this transfer has an inherent contextual behavior that can be modeled and shared.

By designing a system that can model the behavioral patterns of humans and objects being transferred between one another, we enable the possibility for sensing modalities to trigger one another to focus on a particular event (e.g. a load sensor providing a higher confidence to a camera that an event is occurring in a particular location) [28, 29]. Furthermore, when individual modalities are partially, or completely, occluded this behavioral model is key in ensuring a correct ownership transfer detection, by providing visibility into which sensing devices are providing accurate information [89]. Details on how the methodology is applied and behavioral models leveraged for multi-modal ownership transition detection are provided in Chapter 4.

3.5 Spatial-Temporal Multi-Human-Object Matching

As with any sensing system that operates in the real world, its performance is affected by the breadth of possible interactions occurring. More simply, the system accuracy and general performance is dependant on the complexity of the behavior of its actors. For example, in a convenience store multiple people can simultaneously perform several types of interactions, with typically several thousands of products available. These interactions can increase the difficulty of a sensing system to detect ownership, through occlusions or ambiguity in nearby object interactions.

Similarly to detecting when a transition occurs, to disambiguate who is interacting with an object a system benefits from the modeling of human motion. However in this case the defining aspects of motion are the characteristics that link the object to the person rather than the general human behavior [29]. These characteristics closely match a particular human to a type of object, such as the spatial proximity of specific joints of a human, or the relative motion required to carry such object. For example, a heavy large box might require two hands to carry while a single cup does not, therefore when simultaneous people are interacting with these objects and ambiguity arises this information allows to make a further distinction. These interaction characteristics are further described and evaluated in Chapter 6.

Chapter 4

Individual Object Ownership Transfer by Modeling Objects' Physical Context

In any sensing system tasked to identify the transfer of objects' ownership, it is necessary to identify the objects of interest, and to do so the first step is to extract the features that define the object, in the sensing space of interest. This means mapping raw sensing data, such as vibration signals or video footage, and map higher level characteristics of this signal to the object in question. While these identity features, or ownership transfer features, can be learned with enough training data, it is often impractical if not impossible to collect and label enough instances when faced with real-world deployments.

In this Chapter I focus on how to model the physical context of both the objects (color, size, relative position, motion) and the scene (zones of ownership interest, physical realism of location of objects) in order to provide higher accuracies and reduce the number of labels required to deploy such a system.

The publication of large-scale object recognition datasets has enabled an unprecedented use of camera-based systems using deep learning. While supervised learning approaches have shown success in classification, detection, and segmentation, these, however, fail to generalize well in the absence of large-scale labeled datasets [97]. Particularly in inventory monitoring applications where a large number of specific objects – e.g. products in retail store, products flowing in and out of

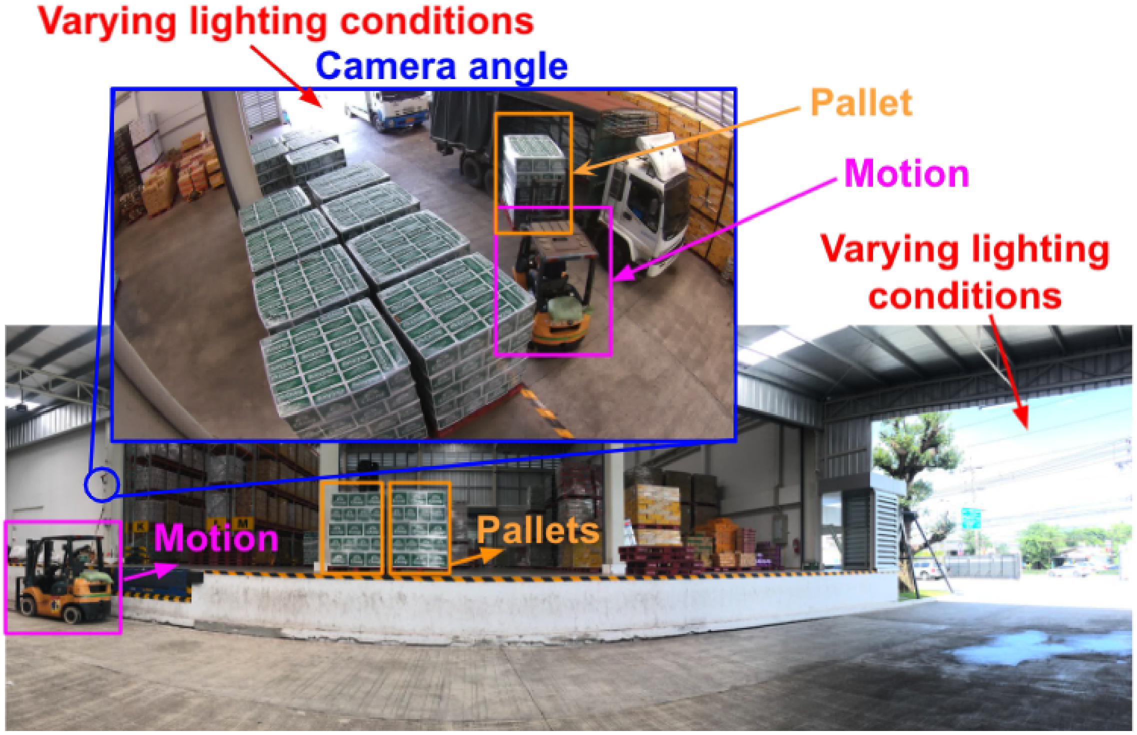


Figure 4.1: Typical warehouse setup with forklifts moving the pallets in and out. Diverse set of environmental factors are observed: motion specific to forklifts, camera angle limitation, varying lighting conditions impacted by the weather.

a warehouse; and a vast diversity of task-specific environmental factors are present (see Figure 4.1). Such applications require immense human labor to label data and parameterize the model, making it impractical for real-world adoption.

In order to address this issue, recent techniques, such as domain adaptation techniques [118], data augmentation [100], incremental learning [124], few-shot learning [81] and automatically labeling techniques [91], have appeared to reduce the amount of human labor required in order to adapt the learned models to the domain and data distribution of the specific application. Using the physical context in order to generate synthetic data allows for the creation of large datasets out of a very small amount of labels by varying characteristics of the original labeled sample and the creation of a model of the object and the deployment. While this allows the trained model to become robust to such variances, it will further over-fit to the patterns in the augmentation itself.

In this Chapter, we present PIWIMS which explores different ways of generating

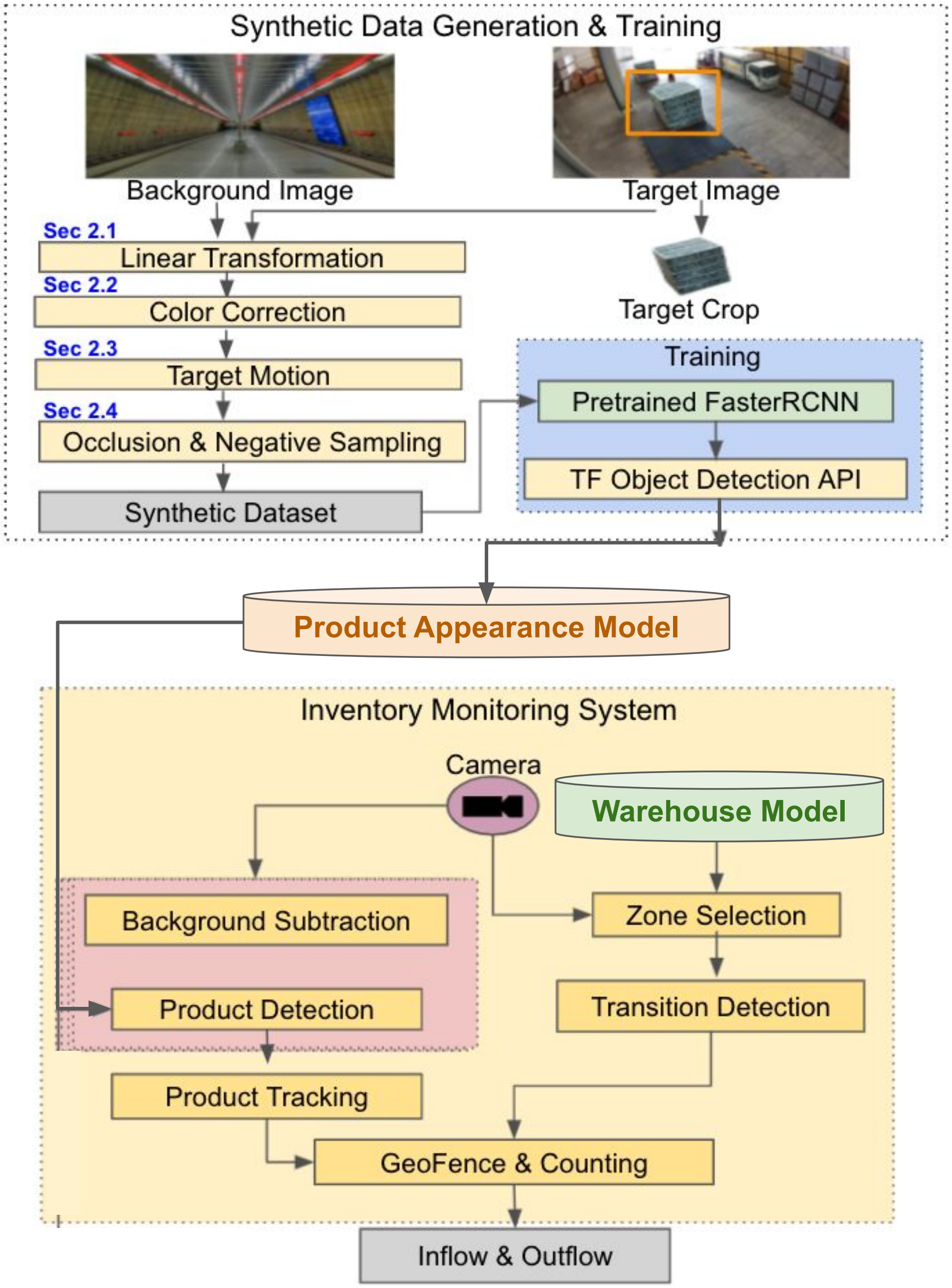


Figure 4.2: PIWIMS consists of two steps: randomly sampling a background image and a target crop, selecting the PIWIMS approach, and generating the dataset according to the desired physical characteristics of the scene and the object. The generated dataset is used to train the product appearance model that is fed to the Inventory Monitoring System.

synthetic warehouse datasets by leveraging the physical context of the deployment setup and objects through compositing augmentation to bind the generative parameters closer to the physical reality with minimal human annotation. The contributions are as follows:

1. We propose a new generative data-augmentation technique PIWIMS, which leverages physical context –scene lighting, camera position and angle, and object motion, size, and shape– to model the objects, and scene, and generate datasets that more closely resemble the deployment data distributions with minimum human annotation labor.
2. We present an analysis of a 4 month deployment of PIWIMS used in a real world warehouse inventory monitoring system.
3. We demonstrate the ability of PIWIMS to generalize across camera deployments with varying position and angle.

The remaining chapter is as follows: Section 4.1 describes in detail the approach use in PIWIMS. Section 4.2 presents the experimental results and analysis of PIWIMS in real-world warehouse setting. Section 4.3 shows previous works on data augmentation techniques and learning with limited labeled data. Finally, we conclude this chapter in Section 4.4.

4.1 PIWIMS Overview

PIWIMS takes inspiration from the ability humans have to look at an image and project different placements for objects of interest. To project an object to a different place in an image, humans have to have a rough understanding of the physical layout of the scene (where the floor is, natural orientation of the object, sources of light, natural motion). This rough understanding of the physical context of a scene is designed into our approach.

Like other compositing techniques [26, 110], PIWIMS requires as input a background image, b, and a target image crop, c. Our approach requires as well a linear transformation function, T, and a color correction function, M, and finally applies an alpha blending technique [54] to compose the positive training case. The approach focuses on constraining the parameters of T and M closer to the bounds of physical possibility, while still exploring a larger data distribution. In the simplest case, the final training image is then computed by: $i = b \oplus M(T(c))$ (where "$\oplus$" corresponds to the alpha blending). However, we can consider multiple c's and end up with the following formulation:

$$i = b \oplus M_{x_1}(T_{y_1}(c_1)) \oplus M_{x_2}(T_{y_2}(c_2)) \\ ... \oplus M_{x_n}(T_{y_n}(c_n)), \quad \forall x \in \mathcal{M}, \forall y \in \mathcal{T} \quad (4.1)$$

Note that each M_x and T_y are different color correction and linear transformations respectively.

4.1.1 Physically informed linear transformation

The linear transformation defined by T in Eq. 4.1 is further defined by the input parameters (X, θ, s). X represents the final location of the center of mass of the cropped image (c) in $\mathcal{R}^2$. X can vary randomly to any point in the image. Θ however is constrained by the maximum rotation that c can see given the application-specific setup. This is computed manually by placing c in its original background b at the highest/lowest rotation possible. By manually placing the crop c, PIWIMS is bounding the space of possible rotations for c. We define the possible set of rotations as $\mathcal{Y}$.

Similarly, the size S of the crop c is equally computed by manually placing c in the original background b. In this case, the annotator will resize the crop to the largest and smallest possible perceived size given the application-specific setup. Once again, bounding the possible sizes of c to the physically reasonable ones. The size of c is defined by the ratio of the area (number of pixels) of the crop c and the area of b, and its possible space is defined as $\mathcal{S}$. Finally $T \in \mathcal{T}$ where $\mathcal{T} = (\mathcal{Y}, \mathcal{S})$.

4.1.2 Physically informed color correction

When bounding the possible synthetic images in the color space, we focused on lighting. This is formulated through the function M present in Eq. 4.1. M represents the color space of the c image in HSL (Hue, Saturation and Lightness) space. In order to encode the lighting characteristics (origin, color, and intensity) easily intuited by humans, PIWIMS requires the annotator to understand the lighting in the application-specific setup and place the cropped image c in the original background b, and, similarly to subsection 4.1.1, search for the maximum/minimum hue and lightness value possible. The range computed between the maximum and minimum values define the space $\mathcal{M}$. Our approach ignores saturation based on the fact that c is extracted from the same source that the model will operate in, which already includes the data characteristics of the light sensor. This assumption can further be alleviated to improve the model.

While this is a homogeneous color mapping transformation, we believe that it is possible to enhance this model with a directed lighting source.

4.1.3 Target Motion

Even though the motion is associated with a continuous set of images (video), it can still be perceived in a static image. Motion blur can be detected in a single image, indicating a natural motion that creates a loss of detail due to the capture rate of the camera's sensor. The same principles applied in Section 4.1.1 and Section 4.1.2 can be applied here. Motion blur can be applied by a convolution between a kernel k, defining the direction and intention of the motion, and the image c.

In cases where the ordered series of images is available, motion can further be used through a pre-processing step of background subtraction [101]. PIWIMS transforms the background image b using a synthetic background subtraction removal function f. f randomly samples the coordinate space of b and converts that pixel into a black pixel with probability p. In case one of its 8 neighboring pixels has been converted to black its "turning black" probability doubles becoming $(1+p)/2$. This is defined as such:

$$P(b_{x,y} = 0|p) = \begin{cases} p & , \forall x, y \text{ where } b_{|x-1|,|y-1|} \neq 0 \\ \frac{(1+p)}{2} & , \forall x, y \text{ where } b_{|x-1|,|y-1|} = 0 \end{cases} \tag{4.2}$$

We have identified empirically that $p = 0.53$ has the best performance when reproducing the effects of background subtraction. See 4.2c). for an example of a resulting background images with $f(b, 0.53)$.

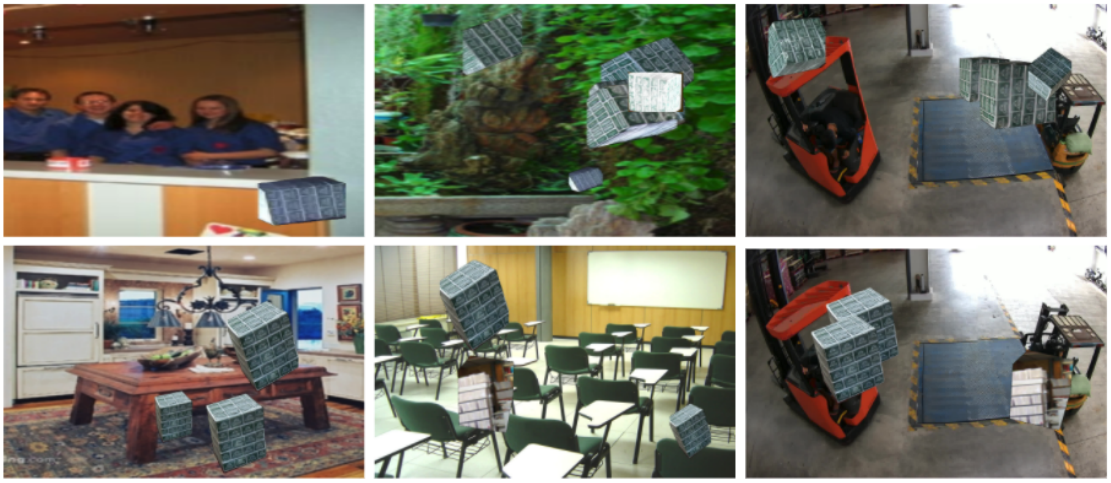


Figure 4.3: The top three images show the effect of random dataset generation that lies outside the physical reality of the application. The bottom three images demonstrate the generation process based on PIWIMS linear transformation, color correction and negative sampling.

4.1.4 Occlusion and Negative Sampling

When blending the background, b, with the multiple target cropped images, c's, [110] has demonstrated that the artifacts created by composing the final image can be identified by the training model, which leads to it expecting such artifacts in the application-specific images. Therefore PIWIMS further adds similarly shaped, to c, extra cropped images denoted as v with content from the different random background, in order to create negative samples with the compositing artifacts. (See Fig. 4.3). The last step of PIWIMS is to remove samples generated with extreme occlusions by applying NMS (non-maximum suppression) with a threshold of 0.8 on the generated c images.

4.2 Real-World Evaluation at an operating warehouse

In this section, we present a component-wise analysis of the different stages of PIWIMS and the results of the deployment of PIWIMS in a real-world operating warehouse monitoring system. In partnership with ThaiBev, we have deployed a warehouse inventory monitoring system based on the data generated by PIWIMS.

We analyze in this chapter 4-month of operation of this warehouse. And further evaluate each component of PIWIMS in Sections 4.1.1, 4.1.2 and 4.1.3 (See Fig. 4.2).

4.2.1 Warehouse camera deployment and dataset generation

We deployed two cameras at a ThaiBev's warehouse for varying camera positions and angles. The cameras' placement is chosen in a way to capture the maximum activity occurring at each door. The cameras operated at 15fps and recorded all activities during the four months of the deployment. The deployment was coordinated to record the 4 busiest months of the year. We have further obtained the inventory count from a separate warehouse monitoring system (WMS) for comparison.

Ground truth was generated by manually annotating 35 hours of randomly selected video frames from different days and cameras. We further evaluate the effectiveness of our approach by comparing our visual inventory monitoring results with ThaiBev's WMS results.

Dataset generation was achieved by extracting 32 cropped product images of the highest selling item in the warehouse from each deployed camera, c_n. We have further leveraged the Indoor Scene Recognition Dataset [80] to generate our backgrounds b for an increased variety of backgrounds which facilitates the generalization process. By following the approach describe in Sec. 4.1, we create fully synthetic datasets of 30,000 images out of the 32 cropped sampled images (with a 20:20:60::validation:testing:training dataset composition), for each component provided in Sec. 4.1 (See Fig. 4.2). We then train a FasterRCNN [85] model with the synthetically generated datasets for the different steps in PIWIMS.

4.2.2 Metrics for Product detection and Product flow counting

Inventory Monitoring requires the vision systems to detect the products of interest and track their flow over time. For this purpose, we have divided our analysis of the results into *Product detection accuracy* and *Product flow counting*.

Product Detection is a localization and classification task, which we evaluate using the F1-score @ 0.8 IoU, that is $\frac{|G \bigcap P|}{|G \bigcup P|} >= 0.8$, where G is the ground truth bounding box and P is the predicted bounding box.

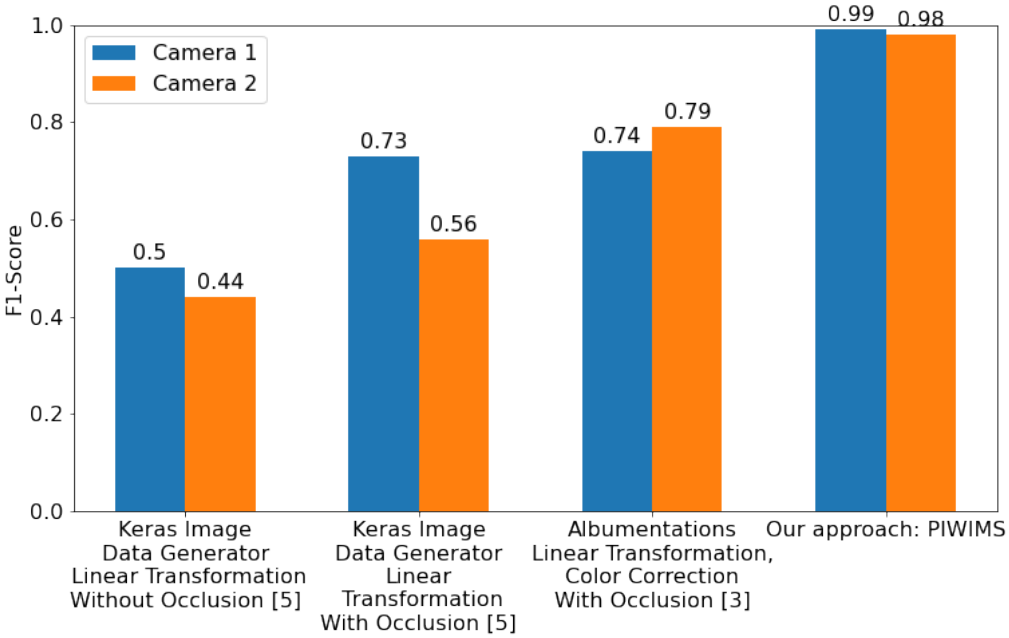


Figure 4.4: Detection Accuracy results with IoU = 0.8, using fully synthetic datasets generated using various data augmentation techniques. We show that PIWIMS with minimum human labor (only 32 manually labeled images) outperforms the other techniques for both the deployments.

Product flow counting aims to estimate the locations of products in a video sequence and yield their individual inflow/outflow count based on the entrance and exit zone in the warehouse, which we evaluate using the F1 score. If the ground truth states a product inflow/outflow at time t_1, and the predicted time states a product inflow/outflow at time t_2, it is considered a true positive if $|t_1 - t_2| <= 50$, where we consider 50 frames (equivalent to 3.5 seconds) to accommodate for human annotation errors.

4.2.3 PIWIMS data augmentation performance

In order to evaluate the performance of PIWIMS we have leveraged several data augmentation techniques and used them for the warehouse inventory monitoring deployment. We compare our model with Albumentations [9], and various techniques from Keras Image Data Generator [16] on product detection and product flow counting subtasks. The results show that all the components of PIWIMS perform better or on-par with the baseline on respective tasks (See Fig. 4.4 and 4.5).

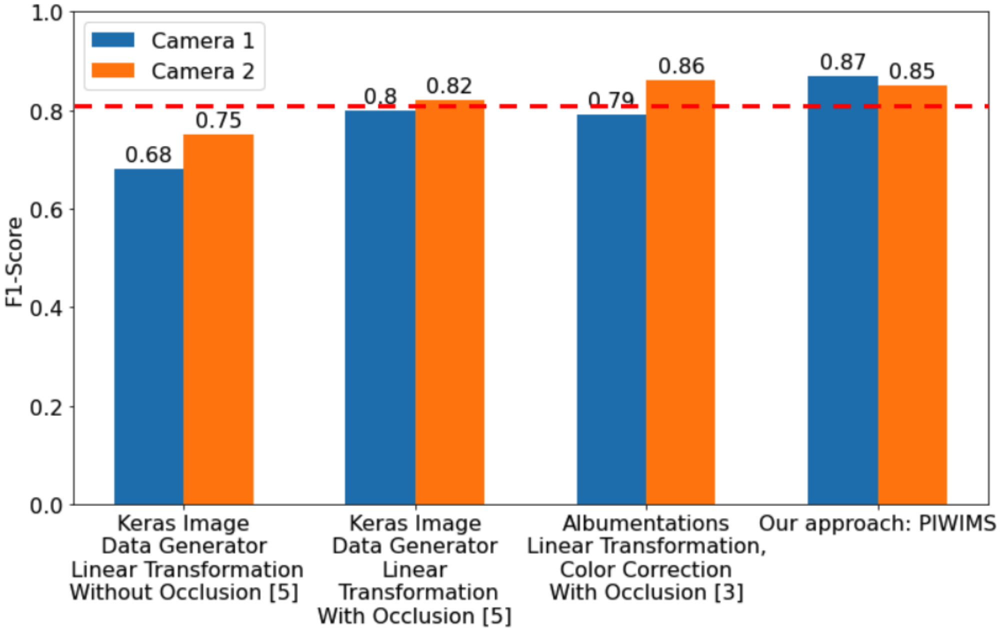


Figure 4.5: Inventory Count Evaluation Results for different models, using synthetic datasets generated with only 32 manually labeled images. Our approach, PIWIMS has an 8% performance increase compared to Albumentations [9] 31% error reduction as compared to WMS (dotted line).

Product Detection: Following the apporach in Sec.4.1 we create detection models from the synthetically generated datasets for PIWIMS, Albumentations [9] and Keras Image Data Generator [16] methods. We demonstrate in figure 4.4 that with a minimum amount of human labor (only 32 cropped samples), a combination of all the components of PIWIMS achieves 99% F1-score @ 0.8 IoU.

Product Flow Counting: We detect the inventory by running our trained models and then use SORT [123] to track the movement in 2D of the products across time. We further define an entrance and exit zone in the image, determining whether a moving product is considered inflow or outflow. Figure 4.5 shows that we see an 8% improvement in F1-score for the first camera and on-par performance for the second camera as compared to other data augmentation techniques.

Furthermore, we compare our model with WMS data (See Figure 4.5), the 3rd party warehouse industry average. The results demonstrated that our models resulted in a 31% reduced error as compared to WMS.

4.2.4 Generalization across different camera angle and position

In order to demonstrate the ability of PIWIMS to generalize across different camera installations, we deployed a second camera at the same warehouse (see Figure 4.1). The second camera is deployed at the same height from the warehouse platform as the first camera. As both cameras are deployed in the same warehouse, similar physical information is extracted when generating the synthetic datasets.

In Table 4.1 we list the factors that impact the physical parameters used in PIWIMS. As the second camera is deployed at the same height as the first camera, the linear transformations $\mathcal{S}$ remain similar across cameras. However, the linear transformations $\mathcal{Y}$ is adjusted according to the angle of the second camera. The color space $\mathcal{M}$ for images collected from the second camera differs from the first camera. This occurs since the first camera is installed close to the door with direct sunlight access, but this is not the case for the second camera installation. Motion blur and occlusion parameters are similar across camera installations as they are specific to the warehouse, and the product of interest moves at similar speeds from both perspectives.

Evaluation was done by running the same trained model produced for the first camera on the second camera. We randomly selected 10 hours of video from the second camera, from different days and times of the day of the four months of the deployment. This data was annotated by manually looking at the video and registering how many inflows and outflows of the desired product occurred during this 10 hours period. Figure 4.4 and 4.5 demonstrate that PIWIMS generalizes well across different camera installations without added human labor for annotation. PIWIMS exhibited a 19% improvement on product detection subtask compared to the other data augmentation techniques and on-par performance on product counting subtask.

4.3 Related Work

There is a significant amount of work in synthetic data generation with the intent of overcoming the burden of labeling large datasets for training neural networks. However, most of these works lack the optimization for the target application and focus

Physically informed parameters	Impacted by
Linear Transformations ($\mathcal{Y}$)	Camera angle and position
Color Correction ($\mathcal{M}$)	Lighting angle and intensity
Motion Blur	Warehouse product motion
Occlusion Parameters	Camera position

Table 4.1: Analysis of physical realities that impact the paramters used for data generation in PIWIMS. Notice that, in general, linear transformations and color correction are derived from camera placement motion blur and occlusion parameters are specific to the warehouse characteristics.

on just slightly perturbing the available data distribution. These can be grouped into learning-based image synthesis, photo-realistic mapping and contextual image composition.

Learning-based image synthesis: In order to synthesize positive examples for training object detection models prior works [6, 109, 127] leverage pre-existing labeled datasets and vary its characteristics (i.e. saturation, brightness, orientation, etc.) to augment the data distribution with the attempt at maintaining the patterns that make such an example a positive one. These parameters can be learned by a model to generate these synthetic images [14, 19] or generated through adversarial examples [5, 78, 110]. ST-GAN [54] uses spatial composition to generate new images, however, without the contextual understanding of the target application. These, however, still require a large available labeled dataset to learn meaningful patterns and generate image samples that improve the training performance.

Contextual image composition: Image composition refers to the technique of cropping separate images and composing them in such a way that new images are generated with, ideally, similar data distribution to the target context. While these techniques [26, 27, 117] have shown an increase in performance through the random placement of images crops on random backgrounds, they tend to generate unrealistic datasets, limiting its ability to adapt to new domains [26]. To improve this, data augmentation techniques have adopted a parameterized method to limit the distribution of the composition [6]. However these parameters require a large amount of human labor to fine-tune its performance for each composition. To address this, prior works use heuristical [27] techniques or contextual cues [70] to improve the realism of the composition.

Photo-realistic mapping: Photo-realistic 3D simulation engines, such as game engines, have recently gained popularity for data augmentation purposes [42, 86, 113]. These approaches provide a large degree of control over the possible transformations done to available data and into the realism of the resulting dataset. However, the significant downsides of this approach are the domain gap between simulated images and the real application and the hidden need for human labor to generate and ingest 3D data into the simulator.

4.4 Chapter Summary

We introduce PIWIMS, a physics-informed synthetic data generation technique for real-world deployments. PIWIMS utilizes the physical constraints of the environment and the objects to generate realistic datasets. Our approach requires minimal human annotation, and we demonstrate that with an empirical study of inventory monitoring in a warehouse, where we achieve 87% accuracy in inventory tracking with only 32 manually labeled images.

While these results show great promise in deploying a vision system with virtually no available data and with minimal human labor, they are still preliminary. Future work will involve evaluating this approach against an increasing variety of products and an increase variety of deployments. In addition, further exploration is required to understand the minimum required number of manually annotated samples are required to perform well.

Chapter 5

Ownership Transition Detection by Modeling Physical Relationships between Humans and Objects

In Chapter 4 I showed how to model the physical characteristics of the objects and scene (structures) present in a deployment. While this showed significant improvements in reducing the amount of required labels to track certain objects it focuses on a limited subset of ownership transitions where the original, and final, location of the object define its owner (i.e. moving a crate from outside an warehouse to inside). However there are other scenarios where the location of the objects may change but it does not define its ownership (e.g. moving a chocolate inside a store from a shelf to another). In these cases we have to model the physical relationship between the person and the object, to understand when ownership changes. Furthermore, an understanding of the physical relationship between the people and objects, such as the insight that people tend to grab objects with their hands, reduces the search space required to track the objects in the first place. In this Chapter I will focus on inventory tracking in a retail store setting, which is subject to several objects and people behaviours that require the modeling of their relationships in order to accurately track the objects' ownership transfers.

Traditional retail stores face significant labor costs to monitor shelf inventory regularly, often postponing this operation until off-peak hours. A delay in inventory monitoring causes high sales losses when a particular item is gone from the shelf

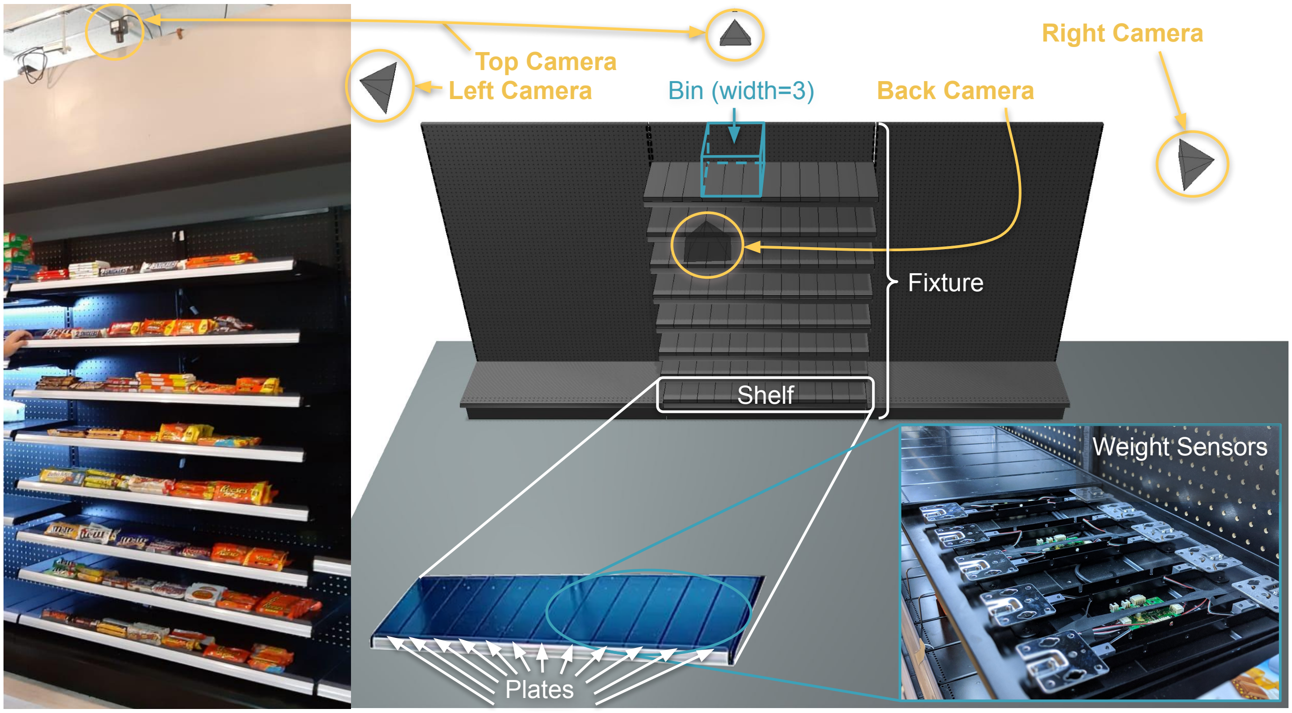


Figure 5.1: FAIM scenario and real implementation. Through weight sensors and cameras, the goal is to autonomously detect and identify what item(s) customers take.

though additional stock existed in the warehouse. An ordinary convenience store faces out-of-shelf stockout rates of 5-10%, which results in a loss of up to 4% of sales [39]. In North America alone, this accounts for approximately $93 billion annual losses [52].

In order to address this issue, current approaches focus on three ways to monitor shelf stock: manual, on-item tags, and vision-based sensing. Manual approaches are the norm and mainly rely on visual inspection of the shelves to reorganize and restock when needed. Employees bandwidth typically only allows for up to a few checks a day, leading to high cost and minimal effectiveness especially in high traffic stores.

Other approaches use sensors on every item (e.g. RFID tags) to monitor remaining stock of each product [8, 20]. However, the added cost of the tags together with the labor cost of labeling every item make this approach impractical other than for high-end goods, such as electronic consumer goods or apparel [20, 69]. More recently, cashier-less stores using a variety of sensors are being explored. Most approaches still require human operators for proper functioning and have highly constrained stocking requirements. While some stores are already operational, such as Amazon Go [1], their automated accuracy has not been revealed and several reports point out that they heavily rely on employees watching the cameras to avoid low receipt accuracy [82, 106].

In this chapter, we present an Autonomous Inventory Monitoring system, FAIM, which tracks shelf-level stock in real-time as the customers pick up or return items. Using weight sensors on each shelf, our system identifies the item being taken based on the location and absolute weight change of an event, which is fused with visual object identification once the item is in the customer's hand. FAIM leverages physical knowledge about the customer–shelf interaction to focus the attention of the visual classifier only on the item being interacted with. We fully implemented FAIM on a 5 shelf setup with 4 cameras and 60 weight plates. To evaluate the system in a real-world setting, we used 85 items from 33 unique products and mimicked the item layout of a local 7-Eleven. Therefore, our contribution is three-fold:

- FAIM, the first fully autonomous shelf inventory monitoring system without human-in-the-loop.
- An adaptive sensor aggregation algorithm to combine information from different sensing modalities, in particular shelf weight differential, visual in-hand item recognition and prior knowledge of item layout (i.e. product location).
- A visual item recognition model training methodology that leverages traditional visual descriptors along with an implementation and evaluation in a real-world market setup with 33 products replicating the layout of a 7-Eleven store.

The rest of the chapter is organized as follows. First, Section 5.4 discusses related works and background. Section 5.1 describes the design of the FAIM system. In Section 5.2 we present the fusion algorithm that combines location, weight and vision information. Next, Section 5.3 provides results and analysis of the real-world evaluation in our store setup. Finally, we conclude in Section 6.7.

5.1 System Design

To the best of our knowledge, FAIM is the first fully autonomous shelf inventory monitoring system without human-in-the-loop. This section provides our system design choices and assumptions.

5.1.1 System Overview

Figure 5.2 shows FAIM's system framework. It utilizes multi-modal sensing to improve item identification accuracy. In particular, I focus on three sources of information: item layout, weight and appearance. FAIM's pipeline is triggered when a change in the total weight of a shelf is detected. From that it extracts two features: the absolute weight difference and the spatial distribution of the weight. The *weight change*-based prediction computes the probability of each product class by comparing the absolute weight difference to each product's average weight. The *location*-based prediction computes the probability of each product class by combining the spatial distribution of the weight change on the shelf with prior knowledge of item layout. The *vision*-based prediction leverages human pose estimation and background subtraction to focus the visual object classifier's attention to identify the object(s) in the customer's hand. Finally, FAIM fuses all three predictions by applying an adaptive weighted linear combination.

5.1.2 Assumptions

Handling all the intricacies and corner cases of a fully autonomous system for inventory monitoring is a very challenging task. I limit the scope of this chapter in order to fully address the problem defined: handling pick up and put back events under *normal shopping behavior*. While I am aware that cultural, age, and other factors impact what's considered normal behavior, I observe some general trends on customers shopping in convenience stores. I make the following assumptions in order to scope this chapter:

1. **At any given time, at most one customer is interacting with a particular shelf**. Unlike big supermarkets, convenience stores observe a much lower customer density albeit a higher foot traffic. In addition, most customers shop

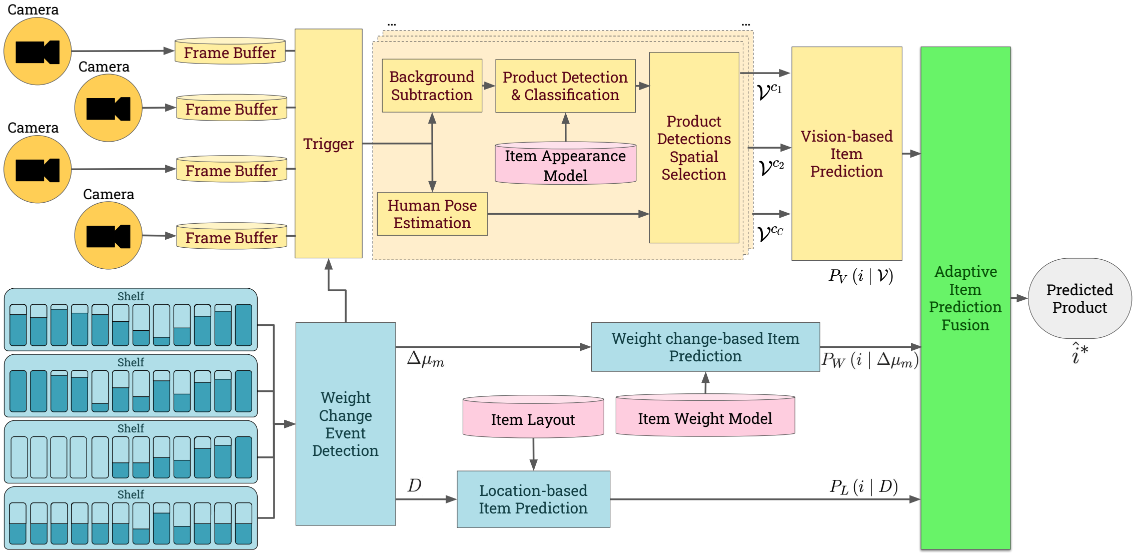


Figure 5.2: FAIM system overview. Top: Vision-based pipeline (yellow). Bottom: Location- and Weight change-based prediction pipeline (blue). Right: Fusion algorithm (green). Prior knowledge of item layout, weight and appearance (pink).

individually and respect other customers' personal space –i.e., if someone is picking an item from the same shelf they want an item from, they wait for the other customer to get their items first. This assumption is particularly true recently given the necessity of social distancing –due to COVID-19, people remain 6 feet apart–. This physical separation and the typical size of a shelf (3 to 4 feet) make this a reasonable assumption.

2. **Customers don't place outside objects on shelves**. It is common for customers to enter the store carrying certain objects, such as a purse or a drink, but unless they are purposefully trying to fool the autonomous system, they rarely leave anything on a shelf that wasn't picked inside the store. Therefore, while users are free to return items they do not want anymore, FAIM can safely assume any put back event corresponds to items from the inventory.

3. **Customers don't alter items' properties (weight or appearance) before putting something back**. For instance, our system assumes that customers won't pick a bag of chips, eat half of them and put them back on a shelf.

4. **Customers pick one item a time**. When customers want multiple items from the same shelf, it is uncommon to pick different items with each hand *at the same time* –i.e., even if they use both hands, they usually pick a product with one first, then the other, so FAIM would correctly flag them as two separate events. Note that this assumption could be relaxed by considering all combinations of up to N items being picked up, though N should be kept small to limit computational complexity.

5.1.3 Hardware design decisions

There are many design choices involved in the instrumentation of smart retail stores for inventory monitoring. In this section, I discuss some insights I gained by implementing FAIM and working with actual retailers, as well as the impact and tradeoffs of different hardware approaches.

Weight sensing

An interesting tradeoff to consider when instrumenting retail store fixtures with weight load cells is the size of each weight-sensing plate, that is, the size of each independent platform suspended over one or more weight sensors. On the one hand, larger sensing areas –e.g., one per shelf or even one per fixture– means lower hardware cost and processing, but also lower signal resolution (the load cells need to support a larger maximum capacity) as well as lower spatial resolution (more items per plate, which increases the chances of having multiple items with very similar weights). In Section 5.3 I explore the impact of different plate sizes on FAIM's accuracy.

Furthermore, weight plate size and design can have a big repercussion on item layout flexibility, an often desired demand by retailers. Product dimensions vary in a wide range, and so does the stock offered at convenience stores and their item layout. Therefore, limiting each product to a single sensing plate, while helping weight sensing by isolating each product, would lead to a hard constraint on the possible products on display, limiting its practical use.

Retailers' profit margins are very low and maximizing item density is of utter importance (see Figure 5.8 for an example of a typical fixture layout in a convenience store). I therefore adopted a flexible hardware design that can be easily mass-produced to bring costs down, as shown in Figure 5.1. Our single-size design consists of narrow weight plates (4" width, divisible by the standard 48" fixture length) laid contiguous to each other. This design has the added advantage that such small weight plates won't have as much weight on top, allowing for lower maximum capacity load cells thus higher weight resolution (sub-gram) without requiring expensive ADCs (Analog-to-Digital Converters). Moreover, Section 5.1.4 details how FAIM handles cases where items span across multiple weight plates (the weight difference in each individual plate does not correspond to the total weight of the item, hence the product prediction has to cluster neighboring weight cells into a single event for a correct item identification).

Vision sensing

There are also many design considerations related to installing cameras in retail stores, from camera specifications, to camera placement and even number of cameras to deploy. As vision processing improves, camera specification constraints can be relaxed. From our initial experiments, camera resolution doesn't play a huge role (in fact most deep learning networks downsize the input image to about 300-720 pixels wide for training and computation efficiency purposes). As for frame rate, I have empirically observed 25-30fps to be enough to get at least one good frame of the item being picked. In addition, optimal lighting might help get sharper and more consistent views of the products, but that is out of the scope of this chapter.

Camera quantity and placement pose trade-offs worth exploring more in depth in future work. Overall, the intuition is that by having multiple cameras spread across different viewing angles, the system can minimize the likelihood of visual occlusions. While this is true, the added hardware, setup, power and computational cost can dramatically impact the benefits of autonomous inventory monitoring. For instance, the first Amazon Go [1] store in Seattle features hundreds of cameras –hanging from the ceiling, on top of each fixture and even below each shelf– and still relies on a human-in-the-loop approach to resolve uncertainties [82, 106]. From our initial experiments I empirically noticed weight sensors to be a much more robust –and cheaper– predictor of what item was picked up or put back on a shelf. Therefore, I do not consider shelf-mounted cameras in this chapter. Section 5.3.2 however analyzes

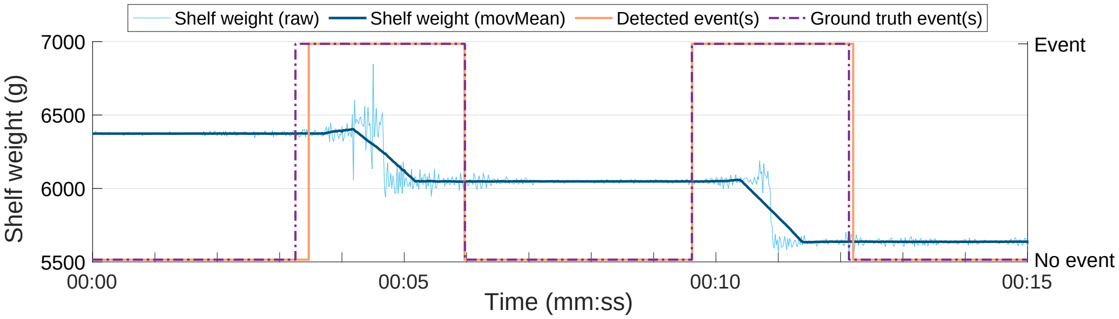


Figure 5.3: Aggregated weight sensed by a shelf during the first 15 seconds of one of our experiments (raw in light blue, filtered in dark blue), as well as the events detected by FAIM (orange) and the annotated ground-truth (purple). Two products were picked up in this section of the experiment.

the impact of any combination among four different camera placements –top-down, sides and in front of each fixture– on FAIM's accuracy.

5.1.4 Customer-shelf interaction detection

The first step in FAIM's pipeline is to detect when an event took place (i.e., a customer picked up or put back an item on a shelf). In our proposed system architecture, displayed in Figure 5.2, the processing of every change in the inventory starts with a weight change trigger. After carefully analyzing some initial experiment data, I came to the conclusion that, even during normal shopping behavior –i.e., customers not trying to fool the system–, visual occlusions from hands or the body are highly likely (especially for smaller items), which makes vision much less reliable than weight for triggering events. Unless someone purposefully drops an object of similar weight as they pick an item from a shelf –which would break Assumption 2–, the weight difference on the load sensors is generally enough to detect an event.

There are numerous prior works to detect events based on weight change on a load sensed surface [73, 98]. In essence these approaches compute the mean and variance of the weight values over a sliding window, and classify the state as either *stable* –no interaction– or *active* –an interaction is taking place– by comparing the moving variance to a threshold. Once the state is back at *stable*, the mean weight before and after the *active* state is extracted and reported as the weight difference of the event, where the sign indicates whether it was a pick up or a put back.

Furthermore, inevitably –even with physical separation between the weight-sensing plates– there might be cases where items lay on more than one plate. In those scenarios, looking at the individual plate scale would yield erroneous weight difference values. Instead, I aggregate all weight plates in each shelf and detect events at the shelf level. This also makes the event detection more robust to light items laying on more than one plate –which might go undetected at each individual plate, but would still provide a big enough change on the aggregate moving variance and mean. See Figure 5.3, which shows the shelf aggregated weight data along with the weight moving variance, mean and the events detected.

Mathematically, let $w_{s,p}^{n}$ define the weight on the p^{th} weight plate on shelf s at discrete time n. I compute the shelf's aggregated weight as:

$$w_s^n = \sum\nolimits_p w_{s,p}^n \tag{5.1}$$

Then, the shelf's aggregated moving mean and variance are, respectively, μ_s^n and ν_s^n:

$$\mu_s^n = \frac{1}{2\,\mathrm{N}_w + 1} \sum_{t=n-\mathrm{N}_w}^{n+\mathrm{N}_w} w_s^t \tag{5.2}$$

$$\nu_s^n = \frac{1}{2\,\mathrm{N}_w + 1} \sum_{t=n-\mathrm{N}_w}^{n+\mathrm{N}_w} |w_s^t - \mu_s^t|^2 \tag{5.3}$$

where N_w is the sliding window half-length in samples, which corresponds to 0.5s in our implementation ($2\,\mathrm{N}_w + 1 = 61$).

An event is detected according to Equations 5.4:

$$\begin{aligned} \text{Event begins on shelf } s\text{:} \quad & \nu_s^t > \varepsilon_\nu, \quad \forall t \in [n_b,\, n_b{+}\mathrm{N_h}) && \text{(5.4a)} \\ \text{Event ends on shelf } s\text{:} \quad & \nu_s^t \leq \varepsilon_\nu, \quad \forall t \in (n_e{-}\mathrm{N_l},\, n_e] && \text{(5.4b)} \\ \text{Temporal consistency:} \quad & n_e > n_b && \text{(5.4c)} \end{aligned}$$

where $\mathrm{N_h}$ and $\mathrm{N_l}$ correspond to the minimum length the weight variance has to exceed or fall short of the threshold ε_ν in order to detect the beginning and end of an event. Based on some initial experiments I empirically set the values to

$\mathrm{N_h} = \mathrm{N_l} = 30$ (0.5s) and $\varepsilon_\nu = 0.01\, kg^2$. Once an event has been detected, the Weight Change Event Detection module determines the event weight difference $\Delta\mu$ and the location –set of weight plates $\{p^s\}$– according to Eq. 5.5:

$$\text{Event weight difference:} \quad \Delta\mu = \mu_s^{n_e} - \mu_s^{n_b} \tag{5.5a}$$

$$\text{Event location:} \quad L = \{\, p^s \;:\; |\, \mu_{s,p}^{n_e} - \mu_{s,p}^{n_b}| \;\geq\; \varepsilon_\mu \} \tag{5.5b}$$

where ε_μ indicates the minimum weight contribution of a single plate in order to be included in the event, which I set to $\varepsilon_\mu = 5\, g$. Once the event weight difference $\Delta\mu$ is determined, this module further computes the event weight distribution –set of weight contributions $\{\Delta\mu_{s,p}^{\%}\}$– according to Eq. 5.6

$$\text{Event weight distribution:} \quad D = \left\{ \Delta\mu_{s,p}^{\%} \;:\; \frac{|\, \mu_{s,p}^{n_e} - \mu_{s,p}^{n_b}|}{\sum_{p'} |\, \mu_{s,p'}^{n_e} - \mu_{s,p'}^{n_b}|} \right\} \tag{5.6}$$

I define Equations 5.5a, 5.5b and 5.6 as the output of the *Weight Change Event Detection* block (as seen in Figure 5.2). I leverage these definitions in *Weight change-based item Prediction* (Sec. 5.2.2) and *Location-based Item Prediction* (Sec. 5.2.1).

5.1.5 Vision event extraction

Understanding customer–item interactions and identifying the products picked from or returned to a shelf from video streams is very computationally expensive (especially for higher camera densities). For this reason, FAIM only saves a small buffer of recent history and uses the Weight Change Event Detection trigger from Section 5.1.4 to start analyzing the images. For *put back* events I directly analyze the buffer as soon as an event is detected, whereas for *pick up* events I delay the vision analysis until the oldest frame in the buffer coincides with the event trigger, n_e from Equation 5.4b.

The Vision Event Extraction pipeline is divided in two sequential tasks: Vision Event Preprocessing and Product Detections Spatial Selection. The former gathers different sources of visual evidence and is composed of the Human Pose Estimation,

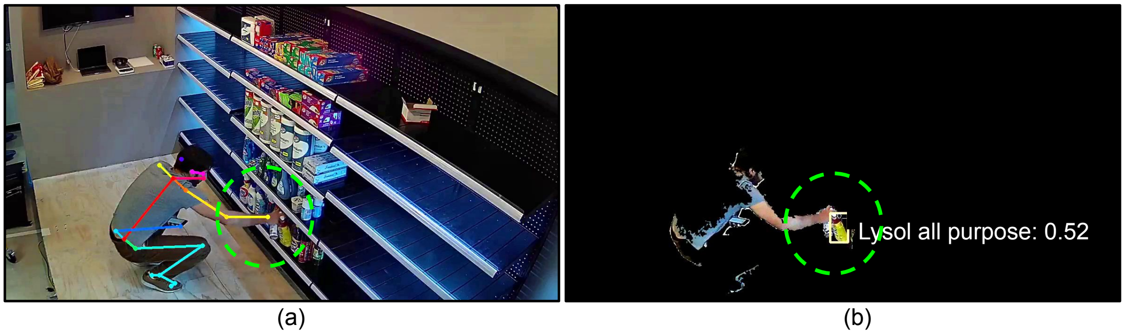


Figure 5.4: Human pose estimation provides us an estimate of the location of the customer's hand, where the visual classifier will look for items **(a)**. I also apply background subtraction to the original image to remove detected items present around the hand **(b)**, such as the objects remaining on the shelf.

Background Subtraction and Product Detection & Classification modules. The latter then aggregates all the information and determines which object detections to keep or reject based on the customer's hand location (See Figure 5.4). As the output of the Vision Event Extraction pipeline, those detections together with their associated product probabilities, are fed into the Vision-based Item Identification module (Sec. 5.2.3) which tries to determine what product was picked.

Vision Event Preprocessing

Ideally, this step should only be comprised of Product Detection & Classification. However, visual object classifiers, such as [41, 85], provide a set of (location, object class) for anything found on an image (i.e., they would also detect all products on the shelves). In order to focus just on the item that was picked, the Vision Event Preprocessing takes additional steps, as shown in Figure 5.2. On the one hand, it performs Human Pose Estimation, a popular research topic in the Computer Vision literature which tries to localize the joints of each person. There are many works in this domain such as [10, 12, 75, 120] which, through different approaches, are all quite mature and robust to varying lighting conditions, clothing and even substantial occlusion. On the other hand, leveraging the fact that cameras are stationary, Background Subtraction techniques such as [40] can be used to "hide" all the products that remain on the shelf and therefore focus the attention of the

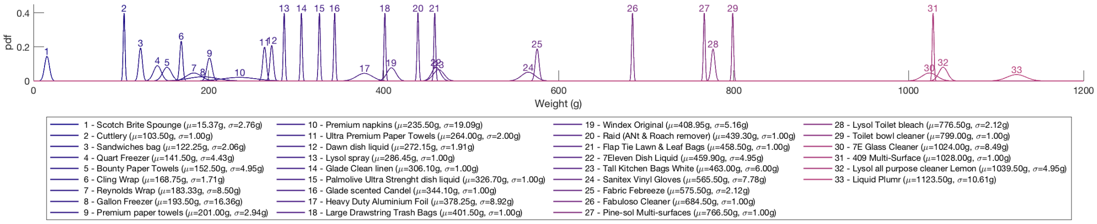


Figure 5.5: Weight distribution of each product used in our experiments.

visual classifier only on the moving *foreground*, where it can find the item being taken or returned.

Product Detections Spatial Selection

From the skeleton of the customers, the location of their wrists represents a simple-yet-effective attention mechanism: by ignoring any detections with centroids further away from the hand than a given threshold, FAIM eliminates most detections of the object classifier that do not correspond to the item the customer is picking or putting back. I call this threshold $\mathrm{R}_{\mathrm{h}}^{c}$, for each camera c, and pick its value empirically based on the camera–shelf distance. This spatial selection can be very useful to complement Background Subtraction when there is more than one customer moving in the scene or when the customer interacting with the shelf has products on the other hand which they had previously picked.

5.1.6 Inventory prior knowledge

FAIM relies on three sources of information to produce an accurate estimate of what item was returned or taken from a shelf by a customer. In order to do so, the system needs to be informed about certain properties of each product. These models can be categorized based on the source of information they provide: item layout (product location), weight and appearance model.

Item layout model

Item layout is a mapping between each product and their –initial– location in the store. The granularity or resolution of this layout could dramatically vary due to different factors, such as store size, complexity of the layout or even time and cost associated with manual annotation. Let $\mathcal{I}$ represent the set of products in the inventory and $i \in \mathcal{I}$ be any particular product (e.g., Scotch Brite sponge, Fabric Febreeze, etc.). In a generic way, item layout can be defined as a function $l(\cdot) \rightarrow \{i \in \mathcal{I}\}$ that returns the set of products expected to be found at any query location. The *resolution* of the item layout can then be defined as the smallest change in location which yields a different value of l. Given the narrow width of our weight plates, the highest resolution considered in this chapter is the plate level: I constrain the query location to a given shelf s and plate p such that $l(\cdot)$ can be rewritten as:

$$l_{s,p} = \{i \in \mathcal{I} \mid \text{product } i \text{ is stocked at plate } p \text{ on shelf } s\} \tag{5.7}$$

Where $|l_{s,p}|$ is the total number of items at plate p on shelf s. This way, I can simulate lower spatial resolutions by recording the item layout at virtual plates p' that aggregate multiple real plates, e.g., $a_{s,p'} = a_{s,p_1} \cup a_{s,p_2} \cup a_{s,p_3}$ (I call p' a bin of width 3). I evaluate the impact of three different levels of item layout granularity –plate, half-shelf and shelf– in Section 5.3.2. Throughout the chapter, I will refer to the item layout model as:

$$\mathcal{L} = \{l_{s,p}\}, \quad \forall s, \forall p \tag{5.8}$$

Item weight model

In order to predict which product was picked, FAIM first needs to have some knowledge about the weight distribution of each product in the inventory, $W(i)$. This one-time calibration step consists in weighing every item and then parametrizing the distribution, which can be approximated by a Gaussian distribution (characterized by its mean μ and standard deviation σ) as displayed in Figure 5.5. The *item weight model* $\mathcal{W}$ is therefore just a list of μ and σ pairs:

$$\mathcal{W} = \left\{(\mu_i, \sigma_i) \, : \, W(i) \sim \mathcal{N}(\mu_i, \sigma_i^2)\right\}, \quad \forall i \in \mathcal{I} \tag{5.9}$$

Note that in general, most products have a fairly consistent weight distribution, thus once estimated, new items would not need to be weighed on restocking.

Item appearance model

There are many ways to encode a visual representation of each item, as discussed in Section 5.4.2. Since the focus of this chapter is on how to combine the different sources of information to produce an accurate product prediction and not in how to improve each source in particular, I follow state-of-the-art approaches for modelling the visual appearance. That way, FAIM is not constrained to a particular object detector and can directly benefit from any improvements in the state-of-the-art. Currently, the best visual identification results are obtained through two data-driven approaches: 1) using *visual descriptors* as defined by the MPEG-7 standards [111]. This approach extracts contextual content associated with images, such as Color, Texture, Shape, Motion and Location. It is most often applied in the domain of search engine for images; 2) Convolutional Neural Network (CNN) based models, such as the ones mentioned in Section 5.4.2 [35, 41, 83, 85]. Either approach requires a large data collection process. I provide more details of our data generation, for both approaches, and training implementation of deep learning models to recognize different object classes –also known as instance segmentation– in Section 5.3.1.

In this chapter I applied both approaches. I trained a CNN and generated *visual descriptors* for all 33 products. I then used the visual descriptors to validate the training accuracy for similar looking products. To achieve this I've collected 20 second videos from multiple angles and distances of a single item of each product placed in a turntable. Then, I cropped the product using background subtraction. For each generated image I've extracted color [63], texture [93] and shape descriptors [7]. Through observation of the distribution of these descriptors I've noticed that each product produces, maximally, four clusters for a full 360^o revolution of the product. Therefore I've extracted the centroid of these clusters and used them to measure the mahalanobis distance between the centroids of each class (see Figure 5.6). I see that the products in our database are visually distinguishable from each other. This is natural as different brands continuously attempt to create their unique visual identity in order for consumers to easily pick out their products from all the similar competitors' products. Although this technique allows us to distinguish the object based on visual information, it is highly sensitive to occlusions. I therefore use this technique as an indicator of which objects are *more* similar to each other in order to then test and validate the performance of our trained CNN with those

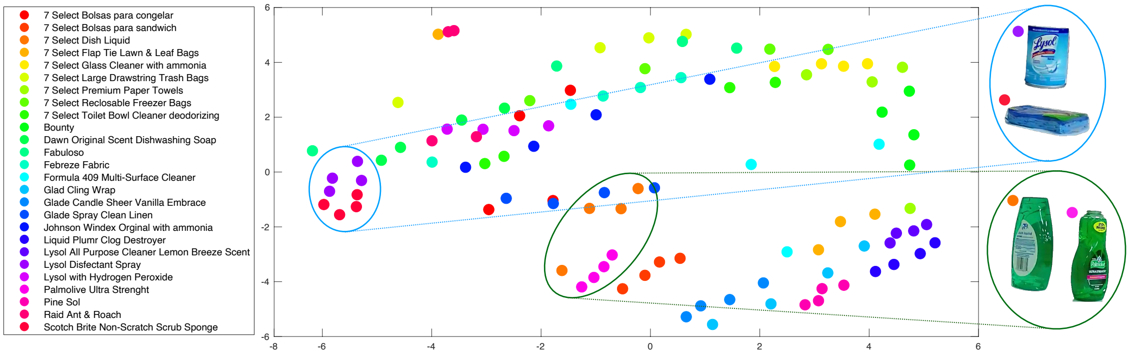


Figure 5.6: Dominant Color Descriptor distribution of each product trained in our model. Each *dot* represents a centroid of a cluster of dominant color descriptors obtained for different angles of each product. Statistical Neighboring Embedding was used to visualize the similarity between different products. *Dots* that are closer to each other represent products that are similar in dominant color. E.g. "Palmolive Ultra strength" and "7 Select Dish Liquid" seen on the right.

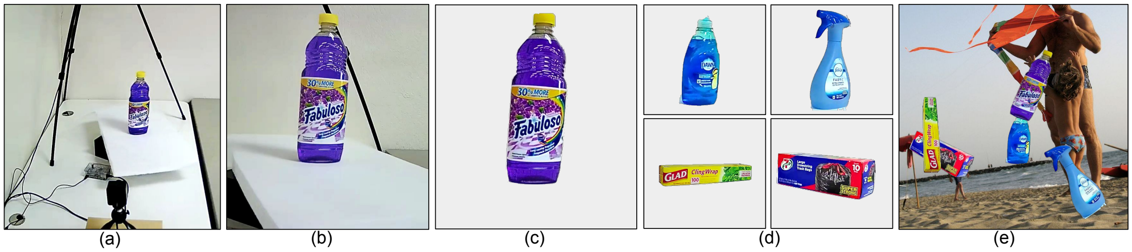


Figure 5.7: An example of the data generation for training. I collected videos of the items from three different angles shown in (a). (b) illustrates the view from one of camera angles. From (b) I removed the background (c) and overlaid the multiple cropped objects (d) onto images from the COCO Dataset [55]. Finally, a Faster R-CNN [85] model was trained with 30k images like (e).

products.

I've further used common data augmentation techniques and generated 30k multi-item images (like the ones on Figure 5.7d) for the network to learn a model of the visual appearance of each product.

5.2 Combining location, weight and vision

The previous section described the item layout, weight and appearance models that FAIM relies on. Here, we detail how each sensing modality estimates the likelihood of the item in the event belonging to each product class, $i \in \mathcal{I}$ by leveraging this inventory prior knowledge (Sections 5.2.1-5.2.3), and how these sources of information are then all combined to emit the final product prediction (Section 5.2.4).

In this chapter $P(I = i)$ is considered uniformly distributed, given that we have generated the receipts by picking randomly from each class of products, using a uniform distribution. However this assumption can change if more information about which products are picked up more regularly is available. A better model of $P(I = i)$ could provide better estimates of which product was picked.

5.2.1 Location-based Item Identification

From the *item layout model* ($\mathcal{L}$, Eq. 5.7-5.8) and the event location information ($L = \{p^s\}$, Eq. 5.5b), the Location-based Item Identification module estimates the likelihood of the item in the event belonging to each product class, $i \in \mathcal{I}$. A simple approach would predict $P_L\left(I = i \mid L\right)$ according to:

$$\text{All items arranged at location L:} \quad \mathcal{L}_L = \bigcup\nolimits_{p^s \in L} l_{s,p} \tag{5.10a}$$

$$P_L\left(I = i \mid L\right) = \begin{cases} 1/\left|\mathcal{L}_L\right| & i \in \mathcal{L}_L \\ 0 & \text{otherwise} \end{cases} \tag{5.10b}$$

where $|\mathcal{L}_L|$ indicates the number of different products stocked at the weight plates $L = \{p^s\}$. Note that since a location L might be composed of multiple plates, we first need to take the union (Eq. 5.10a). Then, any item in the resulting set would have equal probability of being picked, and any item outside of the event's location would be ignored.

However, such approach would be disregarding some useful information: some plates may have observed a much larger portion of the total weight change than others, and it is therefore more likely that the item comes from those particular plates. For this reason, FAIM uses a weight change-guided location-based item identification:

$$P_L\left(I = i \mid D\right) = \sum_{p \,:\, i \in l_{s,p}} \frac{|\Delta\mu^{\%}_{s,p}|}{|l_{s,p}|} \tag{5.11}$$

Where $P_L(I = i \mid D)$ is computed by summing each plate's *weight change*, weighted by the amount each plate contributed to the total *weight change* of the shelf. We then use P_L in our fusion computation, in Sec. 5.2.4 to calculate the final product prediction.

5.2.2 Weight change-based Item Identification

Product prediction based on weight change is fairly straight-forward. The main idea is to estimate how close the event's weight change $\Delta\mu$ is to the distribution of each product, given by the *item weight model.* It is important to also account for the noise affecting the weight sensor readings, which is generally approximated by a normal distribution with zero mean and some standard deviation estimated empirically ($\sigma_w = 5$g in our experiments). We define then ΔM and ΔM_m as random variables of the *true weight displaced* and the *measured weight displaced* in the event respectively. And $\Delta\mu$ and $\Delta\mu_m$ as the values selected from these variables. Furthermore the probability of $\Delta\mu_m$ given $\Delta\mu$ can be defined as a normal distribution, such that:

$$P(\Delta M_m \mid \Delta M = \Delta\mu) \sim \mathcal{N}(\Delta\mu, \sigma_w) \tag{5.12}$$

Then, using the Bayes's rule the probability of the item belonging to each product class is determined by:

$$P_W\left(I = i \mid \Delta M_m = \Delta\mu_m\right) = \frac{P\left(\Delta M_m = \Delta\mu_m \mid I = i\right) P(I = i)}{P\left(\Delta M_m = \Delta\mu_m\right)} \tag{5.13}$$

where, in our experiments, $P(I = i)$ is uniformly distributed and $P(\Delta M_m = \Delta\mu_m)$ is constant since we have the measurement. We obtain:

$$P_W\left(I = i \mid \Delta M_m = \Delta\mu_m\right) \propto P\left(\Delta M_m = \Delta\mu_m \mid I = i\right) \tag{5.14}$$

We include ΔM in the calculation of P_W by marginalizing the joint conditional probability $P(\Delta M_m = \Delta\mu_m, \Delta M = \Delta\mu \mid I = i)$, obtaining:

$$P_W\left(I = i \mid \Delta M_m = \Delta\mu_m\right) \propto \int\limits_{-\infty}^{+\infty} f_{\Delta M_m, \Delta M \mid I=i}(\Delta\mu_m, \Delta\mu)\, \mathrm{d}\Delta\mu \tag{5.15}$$

Given the chain rule:

$$f_{\Delta M_m,\,\Delta M|I=i}(\Delta\mu_m,\Delta\mu) = f_{\Delta M_m|\Delta M=\Delta\mu,I=i}(\Delta\mu_m)\cdot f_{\Delta M|I=i}(\Delta\mu) \tag{5.16a}$$

therefore: $$P_W\left(I=i \mid \Delta M_m=\Delta\mu_m\right) \propto \int\limits_{-\infty}^{+\infty} f_{\Delta M_m|\Delta M=\Delta\mu,I=i}(\Delta\mu_m)\cdot f_{\Delta M|I=i}(\Delta\mu)\;\mathrm{d}\Delta\mu \tag{5.16b}$$

We assume conditional independence of ΔM_m and I given $\Delta M = \Delta\mu$, obtaining:

$$P_W\left(I=i \mid \Delta M_m=\Delta\mu_m\right) \propto \int\limits_{-\infty}^{+\infty} f_{\Delta M_m|\Delta M=\Delta\mu}(\Delta\mu_m)\cdot f_{\Delta M|I=i}(\Delta\mu)\;\mathrm{d}\Delta\mu \tag{5.17}$$

In Equation 5.17 the right side of the integral ($f_{\Delta M|I=i}(\Delta\mu)$) follows Equation 5.9 and is therefore normally distributed given a particular i. In Sec. 5.2.4 we combine *Weight change-based Item Identification* with *Location-based Item identification* by leveraging Eqs. 5.11-5.17.

5.2.3 Vision-based Item Identification

While the Product Detections Spatial Selection module (Sec. 5.1.5) already determines the set of detected objects with high probability of being on the customer's hands, these predictions need to be combined together –for all frames in the buffer– to output a single probability value for each product class. Unlike weight and location, which are very hard to *occlude*, visual classifiers often suffer from temporary occlusions –especially for smaller items or when customers carry multiple items in their hand. As a consequence, simply concatenating (i.e. multiplying) the *logits* (classification score) of all objects would lead to undesired results, since an item not detected in a frame would end with a probability of 0 regardless of how confident all other frames were. We instead propose using a *noisy OR* model, which in essence computes the probability $P_V(I=i)$ that each product was seen by taking the complement of the probability that the product was never seen. Mathematically, let $\mathcal{V}$ represent the set of valid detections for the current event –output by the Product

Detections Spatial Selection module– and $v_i \in [0,1]$ the classification score for each product class $i \in \mathcal{I}$, then:

$$P_V\left(I = i \mid \mathcal{V}\right) = 1 - P(i \text{ not seen in } \mathcal{V}) = 1 - \prod_{v \in \mathcal{V}} 1 - v_i \tag{5.18}$$

This approach is also easy to extend to multi-camera deployments: given the detections $\mathcal{V}^c$ of each camera $c \in \mathcal{C}$, the overall probability $P_V(I = i \mid \mathcal{V}^{c_1}, \ldots, \mathcal{V}^{c_C}) = P_V(I = i \mid \bigcup_{c \in \mathcal{C}} \mathcal{V}^c)$

5.2.4 Item identification combining all sensing modalities

FAIM's last stage of the pipeline fuses all sources of information to emit a final product prediction, and the one with the highest probability score will be selected. Following Bayesian inference, this fusion would be mathematically described as:

$$P(I = i \mid \Delta\mu, \mathcal{V}^{\mathcal{C}}) = \frac{P(\Delta M = \Delta\mu, \bigcup_{c \in \mathcal{C}} \mathcal{V}^c \mid i) \cdot P(I = i)}{\sum_{i \in \mathcal{I}} P(\Delta M = \Delta\mu, \bigcup_{c \in \mathcal{C}} \mathcal{V}^c \mid i) \cdot P(I = i)} \tag{5.19a}$$

$$\hat{i}_{\text{MLE}}(\Delta\mu, \mathcal{V}^{\mathcal{C}}) = \arg\max_i P(\Delta M = \Delta\mu, \mathcal{V}^{\mathcal{C}} \mid i) \tag{5.19b}$$

The main challenge in this sensor fusion arises from the difficulty of estimating the joint conditional probability $P(\Delta M = \Delta\mu, \mathcal{V}^{\mathcal{C}} \mid i)$, since the visual features and the weight change may not be conditionally independent on i. In this chapter, we approximate this likelihood as a weighted linear combination of each individual sensor modality –weight and vision– prediction. We compute the probability P^i_{weight} that item i was picked up/put down, from the weight sensing modality, using:

$$P^i_{\text{weight}} = P\left(I = i \mid \Delta M_m = \Delta\mu_m, D\right) \tag{5.20a}$$

$$= \frac{P\left(\Delta M_m = \Delta\mu_m, D \mid I = i\right) P(I = i)}{P\left(\Delta M_m = \Delta\mu_m, D\right)} \tag{5.20b}$$

Although ΔM_m and D are not independent, they are however conditionally independent given $I = i$. This is true because once i is set, a product is selected. Thus, the location from where the product was taken is independent of its measured weight and vice-versa. We therefore obtain:

$$P_{\text{weight}}^{i} = \frac{P\left(\Delta M_m = \Delta\mu_m \mid I = i\right) P(D \mid I = i) P(I = i)}{P\left(\Delta M_m = \Delta\mu_m, D\right)} \tag{5.21}$$

Using Bayes's Theorem for the conditional probabilities in Equation 5.21 we get:

$$P_{\text{weight}}^{i} = \frac{P\left(\Delta M_m = \Delta\mu_m\right) P(D)}{P\left(\Delta M_m = \Delta\mu_m, D\right) P(I = i)} P\left(I = i \mid \Delta M_m = \Delta\mu_m\right) P(I = i \mid D) \tag{5.22}$$

As in our experiments $P(I = i)$ is uniformly distributed and all terms in the first fraction are constant –given that we have the measurement taken $\Delta\mu_m$ and its weight distribution D. Equation 5.22 can therefore be combined with Eqs. 5.11-5.17, obtaining the following:

$$P_{\text{weight}}^{i} \propto P_W\left(I = i \mid \Delta M_m = \Delta\mu_m\right) \cdot P_L(I = i \mid D) \tag{5.23}$$

In general, information from weight modality (P_{weight}^{i} i.e., Location- and Weight Change-based Item Identification) is a more robust product predictor –partially because it is less affected by occlusions–, thus we assign it a higher relevance when alpha-blending ($\alpha = 0.7$ gave us the best results). It is also worth noting that, as discussed in Section 5.2.3, cameras can be occluded, lighting conditions may change, etc. therefore an object not being seen should not result in a final probability of 0. For these reasons, FAIM *sums* both modalities predictions instead of multiplying to fuse them. Furthermore, while our vision pipeline tries to ensure that only the item being picked or put back is seen by the Product Detection & Classification module, it may happen that several object detections get selected by the Product Detections Spatial Selection (e.g., when the customer has other items in their hand). In those cases, it doesn't make sense that probabilities add up to 1 (e.g. 25% confident that it saw 4 objects), but rather that each product was seen with probability 1. Consequently, FAIM *does not normalize the vision product predictions* before alpha blending. Therefore Eq. 5.19b becomes:

$$
\begin{aligned}
\mathrm{P}^{i}_{\text{weight}} &\propto P_W\left(I=i \mid \Delta M_m=\Delta \mu_m\right) \cdot P_L\left(I=i \mid D\right) && \text{(5.24a)}\\
\mathrm{P}^{i}_{\text{vision}} &= P_V\left(I=i \mid \mathcal{V}\right) && \text{(5.24b)}\\
\mathrm{P}^{i}_{\text{fusion}} &= \alpha\, P^{i}_{\text{weight}} \;+\; (1-\alpha)\, P^{i}_{\text{vision}} && \text{(5.24c)}\\
\hat{i}^{*} &= \arg\max_{i} \mathrm{P}^{i}_{\text{fusion}} && \text{(5.24d)}
\end{aligned}
$$

where $\hat{i}^{*}$ is the product predicted for the event.

5.3 Evaluation

This Section presents our implementation of FAIM, the experimentation setup, the metrics used to evaluate the performance of different approaches, and the actual experiment results.

5.3.1 System Implementation

Our system utilizes a large array of weight and vision sensors. Below we describe the details about our hardware implementation and training procedure, followed by how the experiments were carried out and the evaluation metrics.

In order to understand the effect of having weight sensors at different spatial resolutions, we designed narrow (4") weight plates which fit nicely on standard 48" shelves used by many retailers. This allows us to simulate plates of different widths (which we call bins) and evaluate the bin size parameter.

For vision sensors, we utilized 720p IP cameras and wrote scripts to record all video and weight data to disk. Although we process the results offline, our algorithms run in real-time on a cluster of 3 Nvidia GeForce GTX 1080 GPUs for our vision pipeline –the computation required for the weight change- and location-based predictions is negligible.

As Figure 5.8 shows, in order to fully evaluate a real-world setup, we went to a local 7-Eleven convenience store, purchased all items on a fixture (Fig. 5.8(b)) and arranged them in the same manner on our shelves (Fig. 5.8(a)). In the process, we initialized our system by:

1. Weighed every item and fitted a Normal distribution to generate the *item weight model* (displayed on Fig. 5.5).
2. Marked what items lie on what plates (e.g., 409 is stocked on plates 1 and 2 on shelf 1) to generate the *item layout model.*
3. Extracted general information descriptors from the frames collected for training of each product. Specifically Dominant Color Descriptor (DCD) [103] (displayed in Fig. 5.6), Homogeneous Texture Descriptor [93] and Region-based Shape Descriptor (RSD) [47]. While these descriptors are not suited to distinguish the object during a pick-up/put-down, due to occlusions, they provide insights into the visual similarity of the products.
4. We replicated [121]'s approach for training the visual product recognition pipeline. While rotating 360° on a white turntable (facilitates background removal), three cameras at different angles and distances simultaneously recorded an item of each product class, placed on its back as well as front side. From those videos, the items could be segmented out and overlaid on top of random images in random positions, sizes, rotations, etc. (see Figure 5.7). We pre-trained a Faster R-CNN [85] model (using ResNet-50 and FPN) on [121]'s dataset, trimmed the last (classification) layer, changed the number of output classes to 33+1 (background) and trained on our dataset of 30k images generated from such cropped products.
5. Evaluated the *item appearance model* by collecting data on the most similar products guided by the product visual similarity obtained in Step 3.

Experiment Settings

We designed our experiments to try to simulate a real shopping experience where customers have a notion of what they want to purchase but may not know where items are located. Our 8 participants had never seen this item layout and, at the beginning of each trial (5 repetitions per person) they were given a randomly generated *shopping list* with 3 to 6 items (repetitions allowed). To incentivize the presence of some *put back* events, with 20% probability subjects were asked to return one of their items (without specifying where they should put it). Participants were

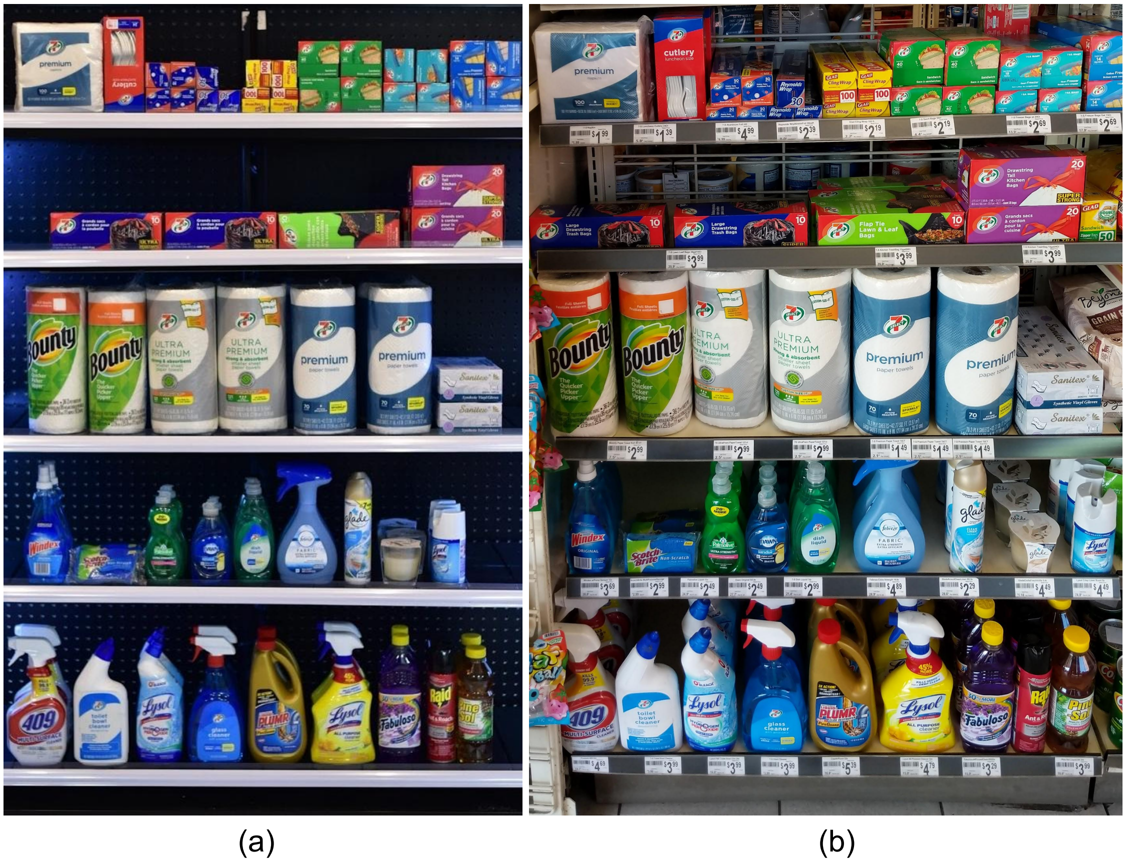


Figure 5.8: In order to follow a realistic item layout and understand potential failure cases in real deployments, we replicated the layout of a 7-Eleven store.

instructed to leave the experiment area once they thought they had collected all items in their list. We set up the 4 cameras to cover the fixture (set of shelves) from all the main angles: top-down, in front and both sides, as shown in Figure 5.1.

Metrics

The main area this chapter tackles is item identification. We define the average item identification accuracy, *Avg. ID accuracy* for short, as:

$$\text{Avg. ID accuracy} = \frac{\#\ \text{correct items predicted}}{\#\ \text{events}}\ (\%) \tag{5.25}$$

Its complement, the average identification error, can then be easily defined as:

$$\text{Avg. ID error} = \frac{\#\text{ incorrect predictions}}{\#\text{ events}} \ (\%) \tag{5.26}$$

It is worth noting that the whole framework relies on a *successful event detection*, that is, missing an event or detecting two consecutive events as a single event would have a direct negative impact in the identification accuracy. However, since event detection accuracy tends to be high and it is out of scope for this work, we focus on the evaluation of item identification (measured by the Avg. ID accuracy). In the next subsection, we analyze our experiment results and the dependency of FAIM's Avg. ID accuracy on different system parameters.

5.3.2 Real-world experiments

In order to fully evaluate the system in a real-world setting, we conducted the "7-Eleven fixture" experiments described above, and present the results here. We want to understand how the resolution or amount of information on a given sensing modality affects the identification performance.

For weight sensing, one of the system parameters that impacts the spatial resolution is the bin width. As a reminder from Figure 5.1, we refer to a bin as a virtual plate that aggregates (sums) multiple real plates, where *bin width* indicates how many plates make up a bin. In Figure 5.9, we can see how location-based (P_L, in light brown) item identification suffers the most as bins get wider –and therefore more items lie inside– dropping from 76.1% at bin width 1 to 11.9% when we only have one plate per shelf (bin width 12). On the other hand, weight change-based (P_W, in orange) identification is independent on the bin width, since it only takes as input the total weight change on the whole shelf. Of course, some items are very close in weight to others (see Fig. 5.5) so only relying on the absolute weight change yields an accuracy of 68.2%. By combining these two weight sensing-based sources of information ($\mathrm{P}_{\text{weight}} = P_L \cdot P_W$, in brown), the accuracy rises to 91.5% and the impact of larger bin widths is reduced, only dropping to 79.0% for bin width 12. Finally, as we investigate deeper below, fusing the weight with visual information ($\mathrm{P}_{\text{fusion}}$, in blue) can bump the accuracy all the way to 93.2% for bin width 3 (slightly

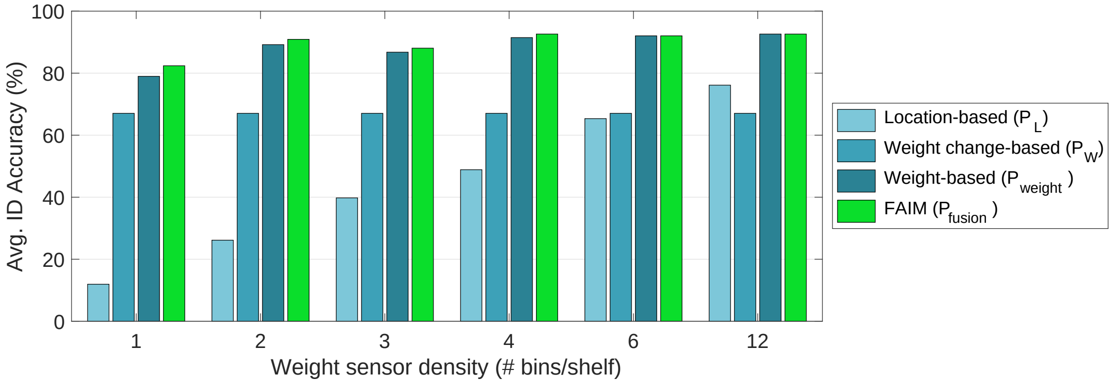


Figure 5.9: Item identification accuracy for weight sensor densities. As the weight sensor density increases (higher spatial resolution) performance –slowly– increases.

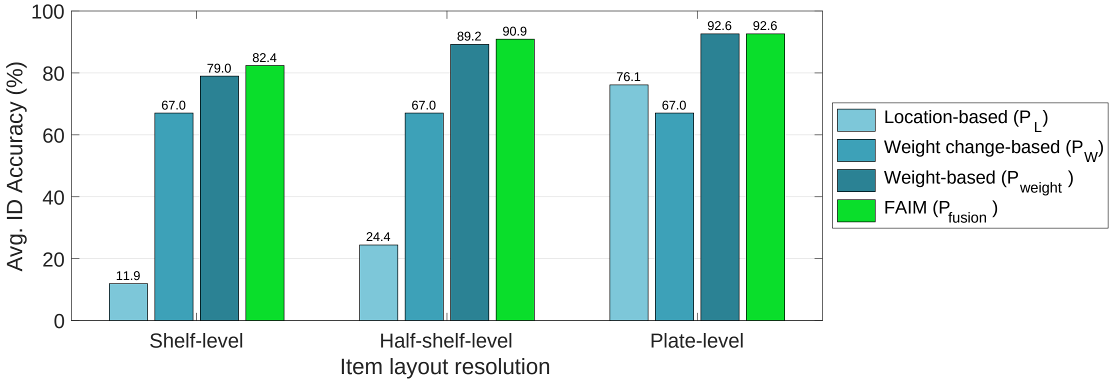


Figure 5.10: Item identification accuracy for different resolutions of item layout (i.e., how knowing what products are in each plate, in every half-shelf or only in every shelf affects performance). Half-shelf offers a great compromise between accuracy and hardware cost.

higher than bin width 1, and corresponds to the width of most of the items in the inventory).

However, it isn't only bin width that affects weight-based predictions. The resolution of the item layout model, $\mathcal{L}$, can also significantly impact performance. As shown in Figure 5.10, we consider three different resolutions: plate-level ($\mathcal{L}$ contains which items could be located at each individual plate), half-shelf-level ($\mathcal{L}$ only records which items lie on the left 6 or right 6 plates of each shelf) and shelf-level ($\mathcal{L}$ only logs the product–shelf mappings). What we observe is that a half-shelf layout model is almost as good as the plate-level layout (90.9% vs 92.0% for FAIM's fusion or 90.3% vs 91.5% for weight-based), while requiring a lot less effort to generate and maintain. On the other hand, when the item layout resolution is at the shelf level, the accuracy drops to 80.1% and 79.0% respectively, a likely unacceptable level for real deployments.

Figure 5.11 explores the benefits of using different amounts and combinations of cameras. For short, we refer to them as **L**eft, **R**ight, **T**op and **B**ack (their exact location with respect to the shelves can be seen in Figure 5.1). For this comparison, we use two different baselines for vision-based item identification:

a) Simply taking the $\arg\max$ on P_V (dark red), which often contains multiple items with probability 1 –and would report a lower accuracy.

b) Thresholding on P_V, e.g., considering an event was correctly identified as long as the ground-truth product had a 0.9 or higher visual score –which captures better whether vision would help the fusion or not.

From the results it is interesting to observe that just using Left and Right cameras already leads to the fusion scores up to 92.6%. It is also worth noting that the low vision-based accuracies reported here are a combination of multiple factors: the domain adaptation gap between the lighting and environmental conditions where the item appearance model was collected and the experiments were conducted, the imperfections on the background subtraction to crop the items and the difficulty of focusing the attention of the visual classifier on the item in the customer's hands, to name a few. But even with this room for improvement, FAIM's fusion approach still can extract useful visual information and achieve up to 3.4% higher accuracy than without cameras, which reduces error from 21 to 17.6% (Figure 5.12), a 19% reduction in error. Cameras contribute the most to the system when less sensors are used, or the knowledge of the item layout is reduced.

It is relevant to note that vision plays an important role as the *weight sensor density* and *item layout resolution* decrease. In other words, as the number of sensors, and the granularity of the knowledge of where the products are in the shelf, reduce, cameras compensate for the reduction in available information (See Figures 5.9-5.10). 92.6% is also an improvement over the reported accuracy of current self-checkout systems, which can give results of only 86% [4] and still is widely used in retail stores.

In the real world there are other factors that contribute to the accuracy of FAIM, such as the number of people interacting simultaneously with the shelves or the density of item arrangement. Sensing signals obtained from pick ups or put downs

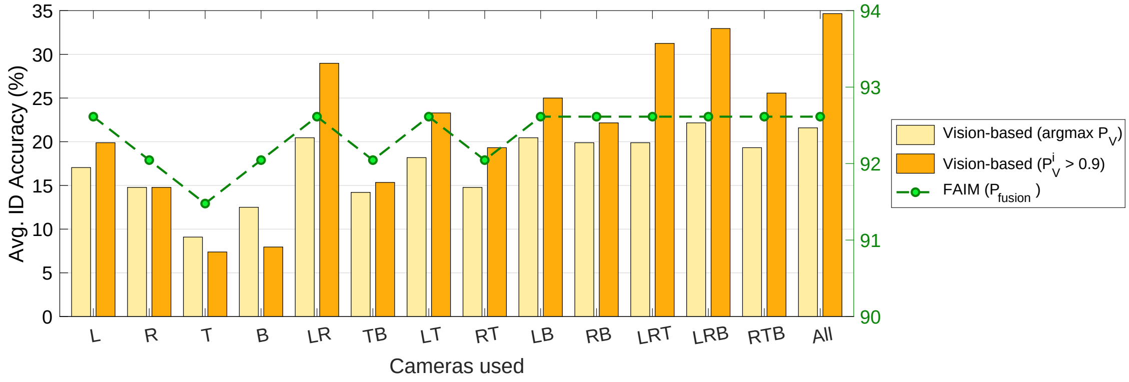


Figure 5.11: Item identification accuracy for different camera combinations (**L**eft, **R**ight, **T**op, **B**ack [see Fig 5.1]). With only Left+Right FAIM already achieves highest accuracy (92.6%).

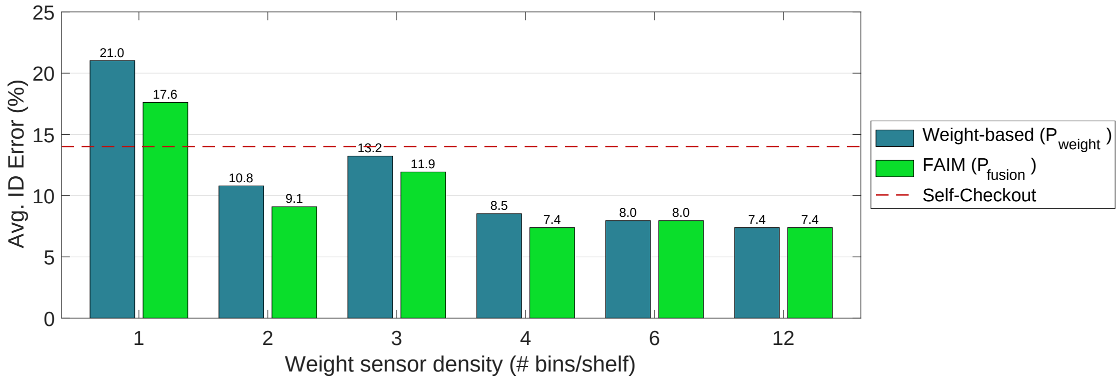


Figure 5.12: Item identification error for different bin widths. Note the reduction in error when vision assists weight (FAIM, beige, is consistently better than no fusion, green). Starting at 2 bins/shelf the Avg. ID Error is already smaller than Self-Checkout [4].

of items in the shelves are affected by multiple people interacting simultaneously, this presents further challenges in the weight change detection module. To address this it is necessary to identify who is interacting with the shelf. This can be done with infrastructure sensing [68], vision [15] or both. However this stayed outside the scope of this chapter, unfortunately due to COVID-19 it became impossible to conduct more experiments and collecting multiple people interacting simultaneously with a shelf. In our future work we intend to study the system performance in crowded scenarios and higher density stores to understand the reliability and deployability of such a system.

5.4 Related Work

There exists a significant amount of work on object identification from weight as well as appearance features, though most focus on only one sensing modality. While there are solutions that can successfully identify objects solely by their weight, they fall short in tackling the inventory monitoring domain, where many sets of products weigh similarly (e.g., soft drinks, energy bars). Although vision-based object identification would be able to tackle the above cases when their packaging is different, the convenience retail market is filled with similar-looking items that have distinct content (e.g. any yogurt vs. its fat-free version). Futhermore there are works that attempt to identify the person interaction using visual and inertial fusion approaches [92], however these are outside the scope of the Chapter given that we are only addressing at most one customer interacting at a time (see Section 5.1.2).

None of these solutions, alone, is capable of fully addressing the autonomous inventory monitoring problem due to the nature of the sensing modalities and the complexity of the environment. Sections 5.4.1 and 5.4.2 cover the state-of-the-art in weight- and vision-based object identification, respectively, and their individual limitations.

5.4.1 Weight-based object identification

The location of an event is a key piece of information to understand what object was picked –or placed– and is used by most prior works to complement the magnitude of the total change in weight. Different hardware approaches have been proposed to measure the weight on a surface and identify objects placed on it. A common solution consists in instrumenting a platform, such as a table or a shelf, with multiple load cells to measure the total weight of the objects on the platform [72, 73, 99]. While this makes it possible to identify objects by tracking changes in the total weight, it cannot handle cases where certain object weights are multiples of each other. In addition, location information can only be retrieved during an event and for that particular object.

To overcome these limitations, some stock-level monitoring approaches rely on capacitive weight-sensing mats [36, 38, 58, 65, 66, 87]. By measuring the change in capacitance between two parallel plates, these works analyze the presence, absence or shape of individual items to identify and count them. However, such approaches become hard to utilize when the sensing material is facing the harsh environment of an operating convenience store: liquids spilling, continuous impact, friction, permanent deformations and cost dramatically constrain their practicality in a store setting. Furthermore any weight-based *only* object identification method suffers from low accuracy when dealing with categories of items that contain a high variance of weight (e.g. fruits, vegetables or packaged salads). For such classes of objects other sensing modalities are required.

Finally, a small subset of prior works have attempted to complement weight sensing with vision, for applications such as grocery re-identification in a fridge [46, 99]. Despite the small scale and highly controlled environment, these works indicate the benefits of multi-modal sensing for item identification.

5.4.2 Vision-based object identification

There are two major approaches for visual object identification:

Traditional **feature-based** and **descriptor-based** techniques, such as Bag of Features (BoF) [76], shape descriptors for object retrieval [61], Local Binary Pattern (LBP) [33] and Speeded-Up Robust Features (SURF) [102], rely on different sets of features designed to extract meaningful characteristics of regions in an image, which allows the detection and classification of objects. In practice, these approaches involve a high degree of parameter tuning, limiting their scalability and adaptability to new scenarios –such as different lighting conditions or camera angles [62].

On the other hand, Deep Learning techniques try to overcome this issue with a **data-driven approach**. Mask R-CNN [41], Faster R-CNN [85], Yolo [83], CenterNet [25], NormalNet [116], FoveaBox [50], all leverage a big amount of data, both in 3D and 2D, to generalize the characteristics of the objects to be identified. While in theory they should not require human input, the training procedure still involves some hyper-parameter tuning. In addition, in order to achieve high accuracy levels, these models require thousands of images of each object, under different conditions, with different backgrounds and from different angles. These images require labeling of the ground-truth for training, this can either be done manually, creating a labor intensive problem, in a semi-supervised [107] or fully automated fashion [91].

In the retail domain, several prior works have tackled the item identification problem [48, 79, 108, 121] but current state-of-the-art is still limited in accuracy due to the great similarity in appearance across products (e.g., chocolate bars of the same brand with different sizes, flavors or textures). On top of that, none of these works take videos as input, meaning an added layer of scene understanding is required in order to figure out which of the detected products is the one the customer took (vs. products which remain on the shelf).

Chapter 6

Multi-Human Multi-Object Matching by Modeling Interactions between Humans and Objects

The previous two chapters' experiments face a common limitation that is present in most sensing system, the fact that real-world deployments are subject to multiple people interacting with multiple objects simultaneously. In the previous example of a retail store, I focused on a situation where multiple objects were present but a single person would be interacting with them. I further exemplified it by designing a system that leverages the framework present in this thesis by focusing on the models of the objects and of the relationships between people and objects. In this chapter I am taking that example further and expanded the system to a entire convenience store that operates normally, that is, with several people and several thousands of objects available for purchase.

In 2020, 90% of retail sales are still happening in physical stores [21, 22, 114]. In these physical stores customers suffer from having to wait in long lines to checkout (get their receipts and pay). These long lines are the most common reason for shopping trip abandonment. Over half of shoppers are willing to spend less money in a store, or even walk away entirely, to avoid a slow checkout [32]. Having to wait in long lines affects customers' satisfaction and loyalty. The checkout experience and length of the checkout line influences the shopping decisions of close to 40% of shoppers [32].

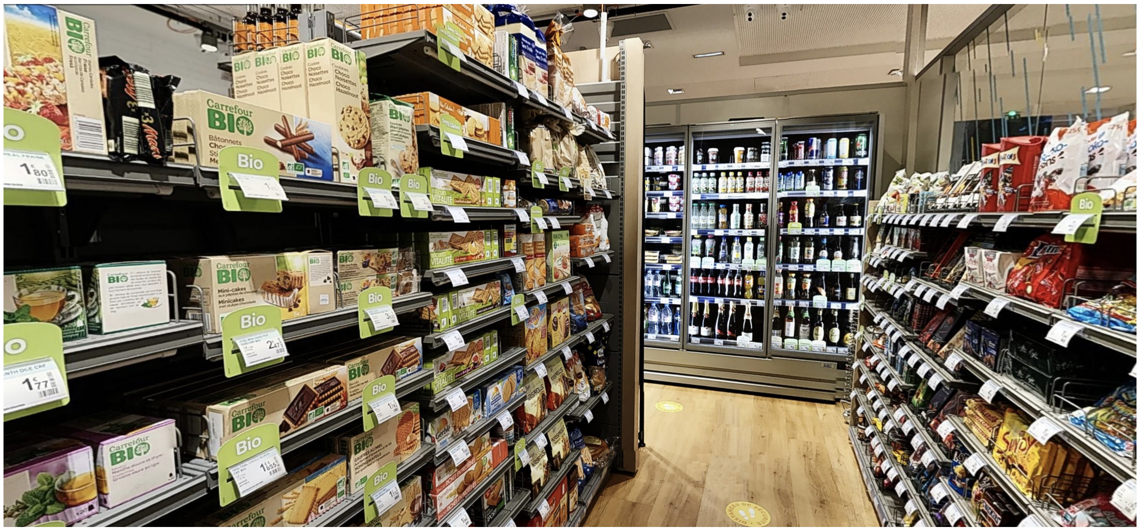


Figure 6.1: ISACS deployed autonomous checkout store. This store is deployed by Carrefour SA. inline with their convenience store format.

Autonomous stores have started to emerge in order to address this issue. Companies (such as Amazon, Grabango, Zippin) have started to experiment with checkoutless stores. In these stores the wait time is removed by providing a receipt without the need for a cashier to scan the products [1, 37, 126].

These autonomous stores can be grouped into three main approaches: vision-only, a combination of computer vision with additional sensors, or a combination of computer vision with sensors and human product recognition. **Vision-only** approaches for autonomous stores are most common. However, these approaches require a large number of cameras in multiple places to minimize blind spots. This approach also require high computation requirements to scale to a store-level deployment. In addition, there is also a very large human labor required to train vision systems to identify products. Tens of thousands of labelled images are required from different angles with different illuminations. Furthermore these prior approaches look at images without obstructions from humans shopping. The best approaches have only been able to reach 85% accuracy in unobstructed retail datasets [94]. This 15% loss is unacceptable for retail businesses, where stores need to operate at a maximum loss of approximately 10% [44]. In order to improve the accuracy, companies such as Amazon Go have added a human-in-the-loop to generate the receipt in the cloud. As the customer leaves a "cloud cashier" will manually review the transaction and images of the items, then the customer will be charged accordingly [1, 82, 106].

This process can generate more accurate receipts, but can cause delays that are up to hours [24]. Amazon has been leading the autonomous checkout space with several Amazon Go stores deployed, however these stores are being developed and deployed behind closed doors, and many of the learning's and details are lost to the broader community.

Other more traditional retailers have started to experiment with automation in their stores by installing *self checkout* and *scan and go* checkout systems [31, 51, 115]. In order to remove the servicing time *scan and go* technology requires the customers to carry a device from the store and scan every item they intend to purchase as they pick them up, avoiding then the time required for the employee to scan the items at the checkout station [31]. However, retailers that used *self checkout* stations did not see a reduction in the customers' wait time due to unfamiliarity of customers to the checkout process. Furthermore, those who deployed *scan and go* technology in their stores found an added product loss of as much as 1% for every 10% of *scan and go* sales [2].

While Amazon is currently operating their Amazon Go stores it is quite hard to estimate their system accuracy, as these numbers are not public information. However prior published approaches which also combine **vision and weight** sensors can achieve up to 94% accuracy in identification by including weights of the item in addition to visual data [89]. However, this approach fails when multiple people are present at the store at the same time due to confusion from *multiple people to multiple objects matching.* Furthermore, the accuracy is limited in this approach due to the assumptions made in a traditional operating stores' setup, while a dedicated *autonomous store setup* can have high impacts on the accuracy.

In this chapter, we present an In-Store Autonomous Checkout System for retail, *ISACS*, which tracks multiple shoppers throughout their shopping journey, combining a physical model of the store and customers, with live inventory monitoring and multi-human 3D pose estimation in order to produce a receipt and present it to the customer in less than 2 seconds. *ISACS* greatly reduces the computational requirements by only using weight sensors on each shelf to track items coming in and out of the shelves while identifying them. Our approach combines the 3-D physical model of the customers and their item interactions with item tracking on the shelves to match multiple people with multiple products in close proximity.

In order to fully evaluate the autonomous setup, operation, and accuracy, we fully implemented *ISACS* on a 800 sqft store with 52 cameras and 580 weight sensing shelves for 13 months in a store owned and operated by Carrefour SA, in Massy France. The store contained on average 1653 unique items, with a real stock of close to 20000 products (See Fig. 6.1). This store had an average traffic volume of around 140 transactions daily before COVID-19 and close to 110 daily transactions during the pandemic. We present our results and lessons-learned throughout the pre- and during COVID'19 pandemic period.

The main contributions of this chapter are:

1. An open and transparent analysis of a deployment for a in-store autonomous checkout system, where a receipt is automatically produced before the customer leaves the store, without human intervention.
2. An algorithm that combines 3D body pose estimation, physical modelling of people and their interaction with the store, and inventory knowledge to enable multiple people and multiple products matching.
3. Analysis and experiences of the deployment and operations of our system in an 800 square feet operational convenience store with 1653 distinct products, over 13 months in a store owned and operated by Carrefour SA, in Massy France.

The rest of the chapter is organized as follows. First, we discuss the background works in people tracking, inventory management and inventory to people association in Section 6.6. Next, we introduce our system in Section 6.1. Then in Section 6.2 we present the association algorithm that combines 3D pose estimation with inventory management. We then describe our evaluation of *ISACS* in the real world in Section 6.3. In Section 6.4, we discuss our experiences and lessons from deploying *ISACS* in the real world, and how we improved the operations of our system over time. Finally, we conclude in Section 6.7.

6.1 ISACS Overview

ISACS is the first deployed, fully operational autonomous checkout system with in-store receipts. ISACS 1) accurately produce receipts in a real-world convenience store setting, while the shopper is in the store (a key and legal requirement for

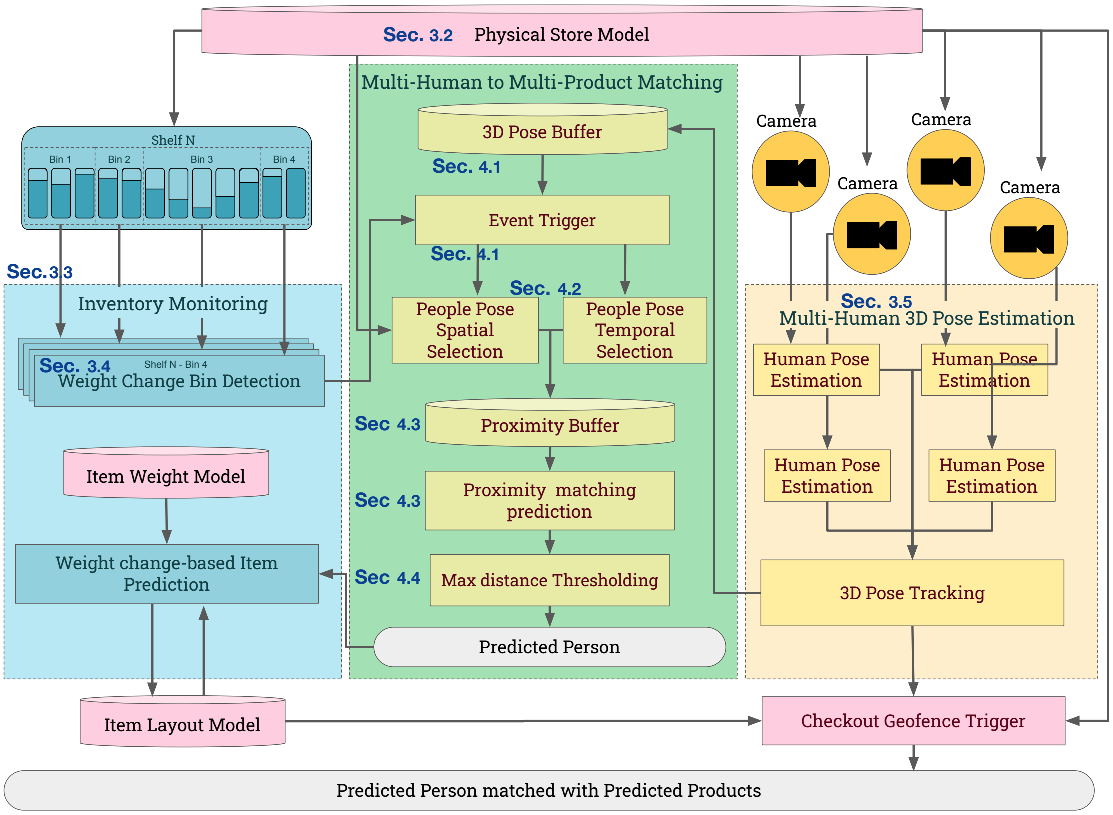


Figure 6.2: ISACS System overview. Left: Weight sensing pipeline for inventory monitoring (blue). Middle: Multi-human to Multi-product matching pipeline (green). Right: Computer vision pipeline for Multi-human 3D Pose Estimation (yellow). Models of the store, Items and Item location (pink). The corresponding sections of each component are noted as Sec. X.Y

operational stores) and 2) does not require human-in-the-loop during checkout. In this section, we provide an overview of our system.

Figure 6.2 shows ISACS system framework. *ISACS* combines a physical model of the store, multi-human 3D pose estimation, and live inventory monitoring to improve the accuracy of the match between the selected products and the customers. *ISACS* is constantly doing cross-view multi-human 3D pose estimation for every camera in the store. As a customer walks in the field of view of more than one camera he/she gets an anonymous ID assigned to him/her. When the customer picks up a product from a shelf, the *inventory monitoring* and *multi-human to multi-product matching* pipelines are triggered. With the knowledge of the physical model of the store, the system is able to understand the 3D location of the product being picked up –*where*–.

From there the ISACS's multi-human to multi-product matching pipeline determines the probability of each person picking up the object –*who*– by looking at the physical distance of each hand over a small window of time during the interaction –*when*–. The inventory monitoring section will compute the probability of the product(s) being picked up –*what*–, as well as the quantity, based on an updated knowledge of the inventory on the shelves and people. A similar process occurs when a customer puts back the objects. Finally as the customer enters a virtually geo-fenced defined area (checkout area), a receipt is then computed based on each event and delivered to the terminal for payment.

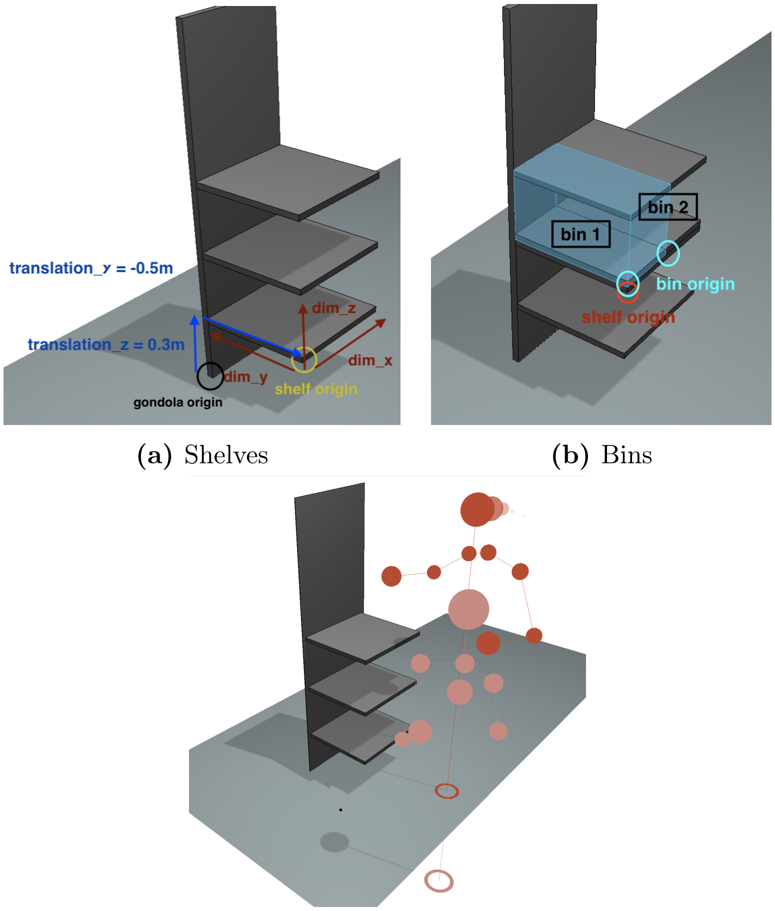
6.1.1 Assumption

Our intention was to operate a normal convenience store with minimal impact to the way customers behave in a normal store, in order to validate ISACS's ability to operate with real world constraints. However we made the assumption that **the last customer who removes a product from a store's shelf and leaves with it is the customer charged for that product.** While this naturally occurs in most transactions, it does not happen for all transactions. In our deployment we educated the customers about this limitation by adding visual diagrams/instructions explaining this.

6.1.2 Physical Store Model

ISACS relies on a centimeter accurate physical model of the store to produce an accurate matching between the movement of products and people. In order to do so, the system needs to know where each shelf and camera are located, in 3D, and their orientation.

We define 4 types of spatial entities: a gondola, a shelf, a bin and geo-fenced areas. We can define all of these by their origin location, dimension and rotation (X_o, X_d, θ) (See Fig. 6.3). We assume that gondolas can only be rotated over the z (height) axis; shelves can only rotate on the y (horizontal, for tilt) axis; while a bin and a geo-fence cannot rotate. We mathematically define G^i as gondola, S^i shelf, and B^i bin with index i. And G^i_c, S^i_c, B^i_c as the center of these volumes. We formalize the location of the gondolas/shelves/bins, such that (note that G can be



(a) Shelves

(b) Bins

(c) Multi-human 3D Pose Tracking

Figure 6.3: Physical store model showing the origin and dimensions of a gondola (a), shelf (a) and bin (b) and 2 people being tracked (c). ISACS's uses this this information to accurately map the inventory changes to the shopper.

swapped with S and B to denote a shelf or a bin):

$$G^i = \left\{ X : \begin{array}{c} |X_x - G^i_{x_o} + G^i_{x_d}/2| < G^i_{x_d}/2 \\ |X_y - G^i_{y_o} + G^i_{y_d}/2| < G^i_{y_d}/2, \quad \forall X \in \mathcal{R}^3 \\ |X_z - G^i_{z_o} + G^i_{z_d}/2| < G^i_{z_d}/2 \end{array} \right\} \tag{6.1a}$$

$$G(X) = \begin{cases} i & X \in G^i \\ 0 & \text{otherwise} \end{cases} \tag{6.1b}$$

$$G^i_c = \left(G^i_{x_o} + G^i_{x_d}/2, G^i_{y_o} + G^i_{y_d}/2, G^i_{y_o} + G^i_{y_d}/2\right) \tag{6.1c}$$

A gondola represents a physical fixture that contains multiple shelves stacked vertically. A shelf contains multiple sensing platforms and multiple bins. A bin is a virtual concept that is defined by the space that an integral number of plates a product occupies on a shelf. Each bin contains only 1 product assigned to be placed inside it. This means that changing the placement of the products implies a simple substitution of the old bin location –containing the products being changed–, with the new bin location with respect to the shelf. Geo-fences are also defined for regions of interest, such as entry, exit and checkout areas. These areas along with the shelves and cameras seldom change.

In addition ISACS requires the knowledge of the location of the cameras and their orientation. This can be obtained via traditional methods for multi-camera calibration. These methods produce a map of relative position and orientation of the cameras without scale. To further obtain the scale and align the two world coordinate systems, ISACS uses a visible *anchor vector* in the form of a sheet of paper with a ArUco pattern, with a fixed size of 20cm, laid on the floor during the calibration step [3].

6.1.3 Inventory Monitoring

ISACS uses live inventory monitoring, achieved through weight sensors, in order to compute the list of products that a person is shopping. ISACS uses a similar approach as *FAIM* [28], that is, each shelf is instrumented with multiple plates, suspended over 2 weight sensors each, along the depth axis (See Fig. 6.4). Depending on the width of the shelf, each shelf contains 4, 6, 9 or 12 sensing plates. Each sensing plate is 12cm-wide and each sit contiguous to each other. This one-sized narrow sensing plate allows for ease of manufacture-ability as well as low cost given that they will not have much weight sit on top of it, allowing for a lower maximum capacity load cell thus higher weight resolution without expensive ADCs (Analog-to-Digital Converters).

ISACS does not use computer vision to further improve the object identification. From prior work results [28], we have observed that the small added value of computer vision comes at the expense of very high computation needs, which limits this approach from scaling to a store level deployment. Furthermore this computer

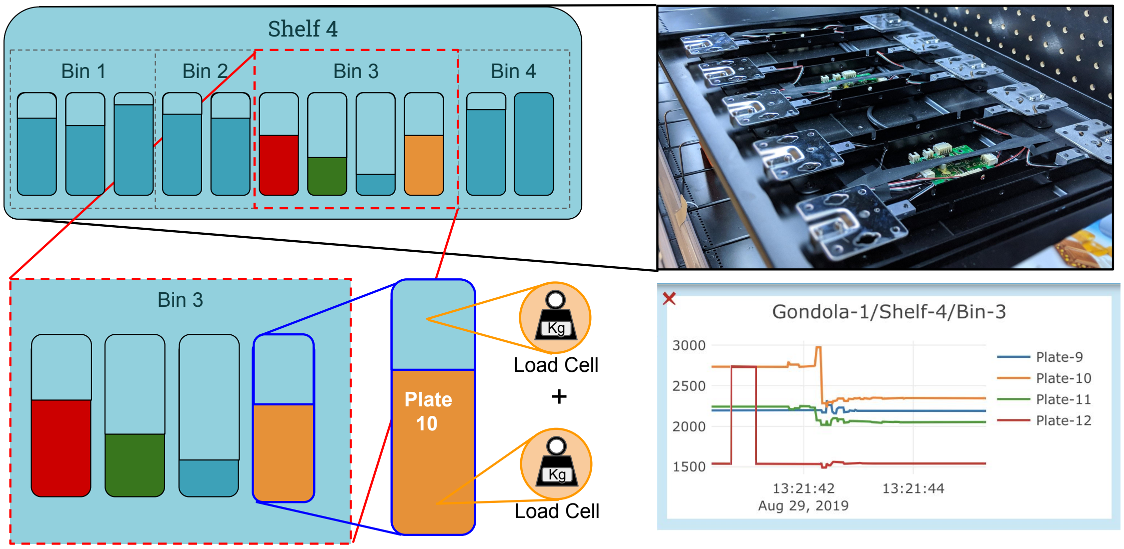


Figure 6.4: ISACS's weight sensor shelves showing a deeper view into how a shelf is divided into multiple bins, each with multiple plates, each with 2 load cells. Top Right image shows a real image of the shelves used with the load cells exposed for clarity. Bottom Right image shows a plot of the resulting measurements used in ISACS.

vision addition only provides extra information in the case the prior knowledge of the location of the objects is not deterministic. In other words computer vision for inventory monitoring only plays a role when objects are either misplaced or sitting in the same sensing plates as other different products. From experience we have observed that misplacements seldom occur in an organized convenience store. **So *ISACS* enforces that each sensing plate has only 1 product type sitting on it**, placing a physical barrier (metal separator, see Fig. 6.5) between products, and that store employees properly re-stock and organize the store once a day.

Inventory monitoring in this context is measured as the quantity of products sitting on a shelf at any point in time. This means that when a customer or employee picks up a product, or puts it down (whether in the right or wrong place), ISACS tracks all of these changes and updates its inventory correspondingly. This leads to ISACS accommodating normal shopping behaviors, such as, picking up a product to further inspect it and putting it down without getting charged for that product.

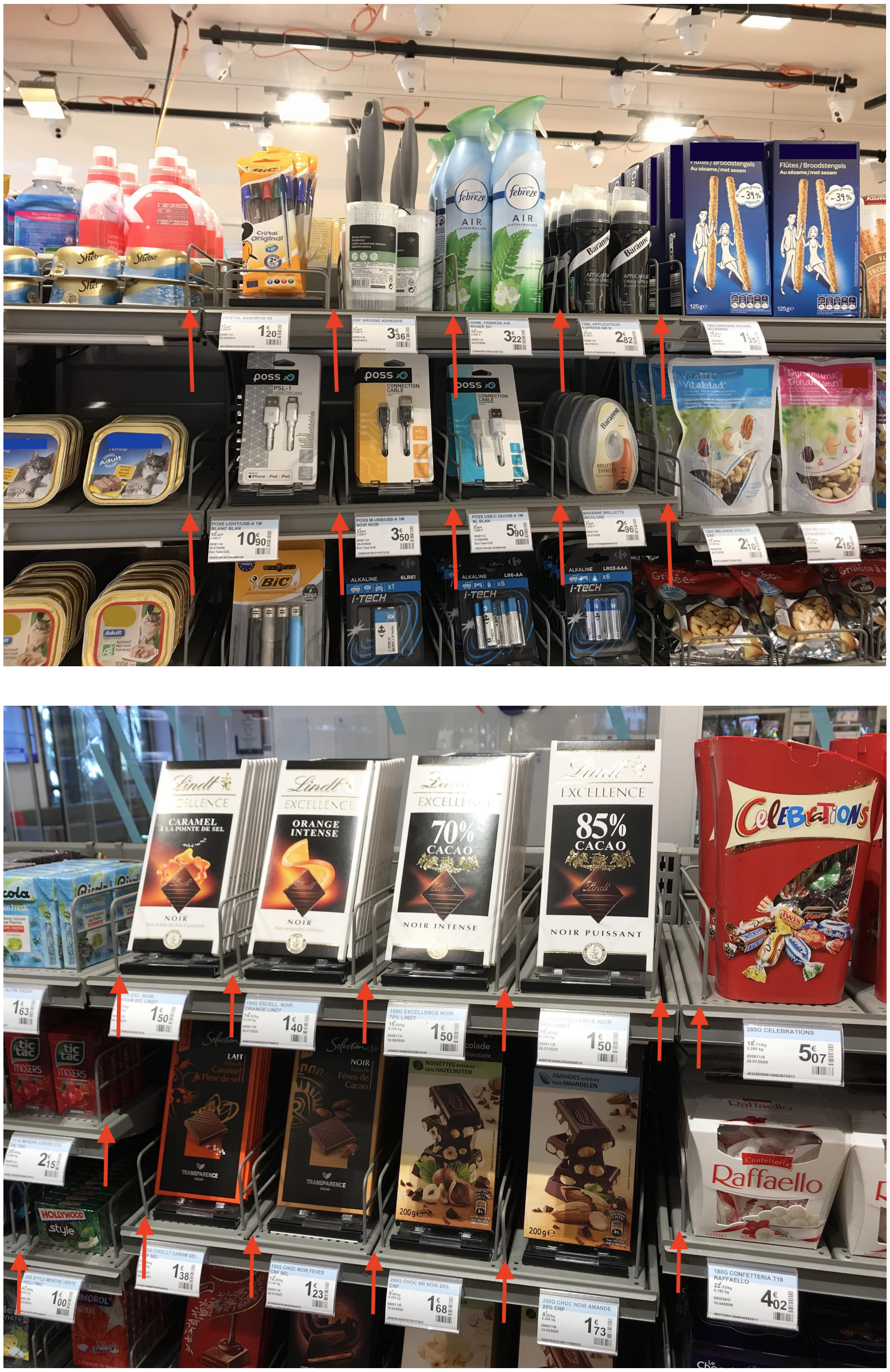


Figure 6.5: Typical gondolas in the store showing each bin separated by metal dividers (red arrows) containing only 1 type of product per bin with multiple items inside.

6.1.4 Weight Change Bin Detection

The triggering point that initiates the association process between a product and a person is the moment the person picks up, or puts down, a product from a shelf. ISACS detects this event by continuously processing the signals coming from the sensing plates. These plates are aggregated into *bins* and processed together as such (See Fig. 6.3b).

In prior work we have computed the mean and variance of the weight values over a sliding window, and classified it as *stable* or *active*. This way we can observe the weight values at the beginning and end of the *activity* and compute the difference in weight. Furthermore, we have observed from previous studies that the smaller the bin width the higher accuracy we obtain for item identification. So we made every bin in the store contain one product only, and cover the minimum consecutive amount of plates that is sufficient to support the product. This makes the event detection more robust to **multiple people** interacting with the same shelf simultaneously by treating each interaction separately. Collision cases still occur if 2 or more customers pick from the exact same *bin* simultaneously. This means that upon a weight change detection occurring, if there are multiple hands within the associated bin, the system has no valid way to detect which hand is the correct one and will decide in favor of the first one that arrived at that location. We have never observed such a case with real shoppers outside of our intentional tests.

Prior work defines mathematically $w^n_{g,s,p}$ as the weight on the p^{th} weight plate on shelf s, gondola g at discrete time n. We also defined $(\mathcal{L})$ as the Item Layout Model such that:

$$l_{g,s,p} = \{i \in \mathcal{I} \mid \text{product } i \text{ is stocked at plate } p \text{ on shelf } s \text{ and gondola } g\} \quad (6.2a)$$

$$\mathcal{L} = \{l_{g,s,p}\}, \ \ \forall g, \forall s, \forall p \quad (6.2b)$$

And $|l_{g,s,p}|$ as the total number of items at plate p on shelf $s$and gondola g. ISACS further defines a bin b as a set of consecutive weight plates $\{p^{g,s}\}$, defined according to the physical store model and constrains the location of the products such that $|l_{g,s,p}| = 1$. Given this we then compute the bin's aggregated weight as:

$$w^n_{g,s,b} = \sum\nolimits_{p \in b} w^n_{g,s,p} \quad (6.3)$$

Then, the bin's aggregated moving mean and variance are, respectively, $\mu^n_{g,s,b}$ and $\nu^n_{g,s,b}$:

$$\mu^n_{g,s,b} = \frac{1}{2\,\mathrm{N}_w + 1} \sum_{t=n-\mathrm{N}_w}^{n+\mathrm{N}_w} w^t_{g,s,b} \tag{6.4}$$

$$\nu^n_{g,s,b} = \frac{1}{2\,\mathrm{N}_w + 1} \sum_{t=n-\mathrm{N}_w}^{n+\mathrm{N}_w} |w^t_{g,s,b} - \mu^t_{g,s,b}|^2 \tag{6.5}$$

where N_w is the sliding window half-length in samples, which corresponds to 0.5s in our implementation ($2\,\mathrm{N}_w + 1 = 61$).

An event is detected according to Equations 6.6:

$$\text{Event begins on gondola } g \text{ and shelf } s\text{, bin } b\text{:} \quad \nu^t_{g,s,b} > \varepsilon_\nu, \forall t \in [n_i,\, n_i + \mathrm{N_h}) \tag{6.6a}$$

$$\text{Event ends on gondola } g\text{, shelf } s\text{, bin } b\text{:} \quad \nu^t_{g,s,b} \leq \varepsilon_\nu, \forall t \in (n_e - \mathrm{N_l},\, n_e] \tag{6.6b}$$

$$\text{Temporal consistency:} \quad n_e > n_i \tag{6.6c}$$

where $\mathrm{N_h}$ and $\mathrm{N_l}$ correspond to the minimum length the weight variance has to exceed or fall short of the threshold ε_ν in order to detect the initial and ending timestamp of an event. Based on some initial experiments we empirically set the values to $\mathrm{N_h} = \mathrm{N_l} = 30$ (0.5s) and $\varepsilon_\nu = 0.01\,kg^2$. Once an event has been detected, the Weight Change Bin Detection module determines the event weight difference $\Delta\mu$ for every bin for which the event occurred.

6.1.5 Multi-human 3D Pose Estimation

Understanding customer motion throughout the store is done through images captured from cameras placed on the ceiling of the store, given this is the place with the most unobstructed view into the people in the store. In Section 6.4 we discuss the camera placement and its impact in the accuracy of the system. In this section we describe the Multi-human 3D pose estimation approach used by ISACS.

The multi-human 3D pose estimation pipeline is broken down into 2 main components: human pose estimation and 3D tracking. The human pose estimators compute the set of keypoints: $\{\mathrm{k_{head}}, \mathrm{k_{neck}}, \mathrm{k_{shoulderL}}, \mathrm{k_{shoulderR}}, \mathrm{k_{elbowL}}, \mathrm{k_{elbowR}}, \mathrm{k_{wristL}}, \mathrm{k_{wristR}}\}$, $k_n \in \mathcal{R}^2$ for all humans present in an image using the same approach as in

[11, 15]. These 2D key points are computed for all cameras in the store and passed on to the 3D tracking component. At this stage the 3D tracking uses a temporal and spatial affinity metric to match the multiple keypoints across different views. Finally by using the camera placement provided by the Physical store model the 3D pose tracker triangulates the multiple views into separate consistent 3D people [15]. (See Fig. 6.3c) ISACS's human pose estimator was trained using the CMU panoptic dataset [45] and enhanced with data collected and annotated from several other deployments provided by AiFi Inc. These deployments included a diverse set of racial, gender, age and clothing. However they were collected from real stores. This means that intentional odd patterns in clothing or masks were not contemplated in these datasets.

We've measured the accuracy of our tracking model in two ways: consistency of shopper identification across its shopper journey (between entering and exiting), and location accuracy of the joints predicted. By counting the number of people coming into the store and coming out of the store, and looking at the consistency of the trajectory we were able to determine that across the 2 weeks dataset (Detailed in Sec. 5.3) we only had 12 incorrect ids tracked out of 1874 transactions, this is approximately 99.36% accuracy in consistent identity tracked. The location of the joints however was measured empirically by our testers. This measurement was done by placing a hand in the center of every bin in the store and verifying if the visualization of the 3D projection matched the center location of the bin. In this case the spatial accuracy of our tracking solution was within 1 bin length of the expected bin. This translates to an average accuracy of approximately 10cm when not occluded. All measurements with respect to shoppers mentioned in this chapter are relative to this tracking system. Section 6.2 further details the impact of the measurements taken by the tracking system in the matching of people to products taken.

While ISACS mainly leverages the approach presented in [15], it improves that approach by adding physical contextual information to the tracker. By leveraging the *Physical Store Model* we improved the people detection accuracy by initializing and eliminating people tracks only at the entrance and exit of the store. This prevented the tracker from generating contextual impossible 3D tracks, such as, someone appearing in the middle of a store or on top of a gondola. It further allows for a person to be missed in some frames, due to occlusions or inaccurate model detection, and

Even our human pose estimation model provided confidence level for the prediction of each joint, our system did not fully leverage this information. In the case the joint was predicted with a confidence level of above 80% ISACS would use that prediction. However, a more promising approach would be to use the confidence level of each joint prediction when doing the matching between products and humans. This information could be useful when deciding between two competing hands picking a certain object.

6.2 Multiple Human to Multiple Products Matching

The previous section described the inventory monitoring and multi-human 3D pose estimation pipelines that ISACS relies on. Here, we detail how ISACS aligns the inventory triggering mechanism with the pose information (Section 6.2.1) and further leverages the pose information and the physical store model to select the correct people interacting with the shelves (Sections 6.2.2) and how this selection is then combined with the proximity of the event to emit a prediction of people interacting with the shelf (Sections 6.2.3-6.2.4).

6.2.1 Visual Event Timing

After an event is detected and triggered by the Weight Change Bin Detection module ISACS *looks* at who was interacting with that shelf at the event time. However, timing between the sensor readings and the video requires synchronization. While all the cameras are able to synchronize via NTP (Network Time Protocol), the custom weight sensing hardware is not. ISACS addresses this issue by making use of a 3D Pose Buffer.

3D Pose Buffer

In order to properly match the timing of the event with the right set of estimated 3D people in the store, ISACS keeps a running buffer of 3D Pose estimated people. This buffer ensures that a fast motion of pick up/put down can still be detected an accounted for.

We define mathematically p_i^n as the set of 3D keypoints (Section 6.1.5) defining the i^{th} person at discrete timestamp n. We also defined ($\mathcal{P}$) as the location of all people in the store at any given time, such that:

$$p_i^n = \left\{ \begin{matrix} k_{\text{head}}, k_{\text{neck}}, \\ k_{\text{shoulderL}}, k_{\text{shoulderR}}, \\ k_{\text{elbowL}}, k_{\text{elbowR}}, \\ k_{\text{wristL}}, k_{\text{wristR}} \end{matrix} \right\}, \quad k_m \in \mathcal{R}^3 \tag{6.7a}$$

$$\mathcal{P} = \{p_i\}, \ \forall i \tag{6.7b}$$

And $|\mathcal{P}^n|$ as the total number of people in the store, at any given time n.

Event Trigger

We define the moment an event is detected from Weight Change Bin Detection (Section 6.1.4) as n_i and the moment it is triggered n_t, such that $n_t > n_i$. These 2 times are never exactly the same given the delay created by the detection mechanism.

6.2.2 People Pose Selection

Once detected an event, in order to accurately match multi-humans with multi-products ISACS assigns different likelihood to the 3D Poses in the buffer, based on the timing and physical distance of each 3D Pose to the triggering bin's location.

From the *3D Pose Buffer* ($\mathcal{P}$, Eq. 6.7b) and the time the event was detected n_i (Eq. 5.4b) the **People Pose Temporal Selection** estimates the likelihood of a person on $\mathcal{P}$ belonging to that event. Such that:

$$\text{Event Time Window:} \quad \mathcal{E} = \{n : n_i - \epsilon_t < n < n_t\}, \forall n \in \mathcal{P} \tag{6.8a}$$

$$P_T\left(p \mid \mathcal{E}, \mathcal{P}\right) = \begin{cases} 1/\left|n_i - n\right| & \forall n \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \tag{6.8b}$$

This means that the closest a person is to the detected timestamp n_i, the higher the likelihood that that person is the one that created the event. ISACS considers the event to have a duration of ϵ_t. Although an event is detected using the weight sensors, the motion of a pickup/put back spans a larger time window than the measure sensing window. We empirically set the value of $\epsilon_t = 3.5$ seconds based on initial experiments.

The **People Pose Spatial Selection** further estimates the same likelihood using the spatial proximity of the detected person to the events location (Section 6.1.2), such that:

$$P_S(p \mid B, \mathcal{P}) = \begin{cases} 1 & B(p_i^n(\text{wrist*})) = b, \forall i, \forall n \in \mathcal{P} \\ 1 - \frac{min^*(|p_i^n(\text{wrist*}) - B_c^b|)}{\sum_{i \in \mathcal{P}} |p_i^n(\text{wrist*}) - B_c^b|} & B(p_i^n(\text{wrist*})) \neq b, \forall i, \forall n \in \mathcal{P} \end{cases} \tag{6.9}$$

This Equations assigns a probability of 1 to a certain person (p) in case the location of either hand of a person is estimated to be inside the bin (b) that generated the event. It further assigns a normalized decaying likelihood to the closest hand of each person inside the buffer.

Extruded Bins. When calculating the spatial probability of each pose for each event in the entire dataset, we have observed that $P_S(p|B, \mathcal{P}) = 1$ rarely occurred. This happened because when the event takes place the hands of the person are inside the shelf –which most of the time is occluded by the upper shelves (except on the top shelf)– leading the 3d Pose estimation to incorrectly predict the position of the hands. Therefore we have extruded the size of the bins towards the front of the shelf by $\epsilon_x = 10$cm. This ensured that we could capture the position of the hand while it was coming in and out of the bin, and consider it inside the volume. We show the results of these approaches in Section 6.3.

6.2.3 Proximity Buffer and Matching Prediction

In this stage of the pipeline ISACS filters out all estimated people from the 3D Pose Buffer that have a likelihood of 0 based on Equations 6.8 and 6.9. This creates a much smaller buffer, the **Proximity Buffer**, which contains the estimated people that are close to the event in both time and space. Finally the Eqs. 6.8 and 6.9 are combined, such that:

$$P^i_{\text{pose}} = P_S(p \mid B, \mathcal{P}) \cdot P_T\,(p \mid \mathcal{E}, \mathcal{P}) \tag{6.10}$$

This Equation creates a list of likelihoods for every 3D Pose inside the Buffer $(\mathcal{P}, Eq.6.7b)$.

Hand inventory information during put backs. During put backs, ISACS leverages an extra piece of information: the inventory in the hands of the people. This means that once a Put Back occurs, at this stage, ISACS removes from the likehood model p^i_{pose} all i's that do not have the products for which the *Inventory Monitoring pipeline* has predicted, leaving only those who could have placed that item in the shelf, in the first place. We show the results of the approaches in Section 6.3.

6.2.4 Maximum Distance Thresholding

The last step in making a prediction on *Who* interacted with the shelves is a thresholding on maximum distance. This means that predictions where $|p^i(k_{wrist*}) - B_c| > \epsilon_s$ are ignored.

This threshold exists to remove events that are created based on ambient noise on the sensors or any faulty signal artifact. In this case if a sensor triggers an event but there is no person close to the shelf we can safely assume that this is a false trigger and ISACS ignores the event. We've defined empirically $\epsilon_s = 2$meters, given that most people will not be physically able to pick any items from the shelves if they are standing 2 meters away from it.

6.3 Real-world evaluation at an operating convenience store

In this Section we present our implementation of ISACS, the experimentation setup, the metrics used to evaluate the performance of our multiple human to multiple product matching algorithm, as well as the natural experimentation results. We further demonstrate how the store setup (proper placement of the products/shelves) can affect the performance of the system, and the steps taken to address these issues.

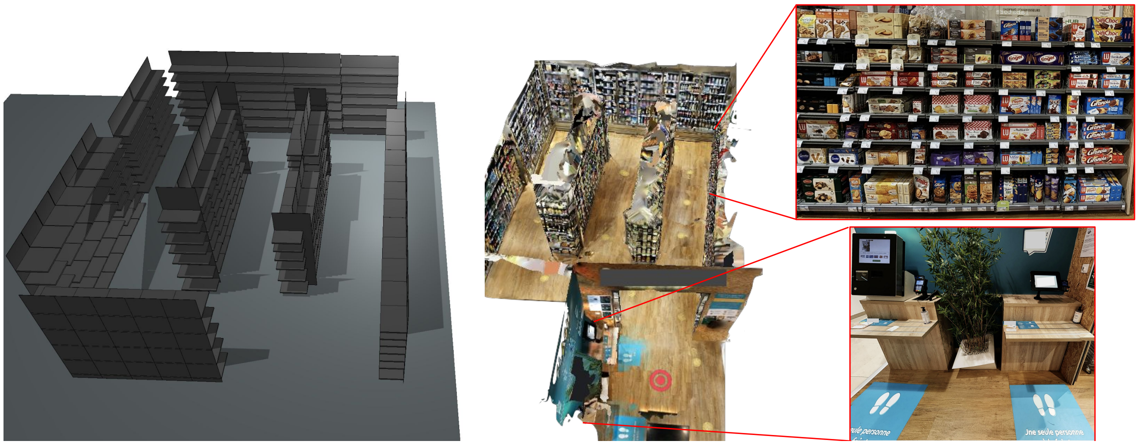


Figure 6.6: 3D physical store model (left) and 3D rendering of the store, with a focus on the payment terminal and 3 gondolas showing the density, distribution and product assortment (right). This store has 52 gondolas, with 6 to 10 shelves each, 6 large refrigeration units with double open doors and 7 shelves each. These were decided and deployed by Carrefour which demonstrates the ability of ISACS to scale to a store level deployment.

6.3.1 System Implementation Designed for Retailers

We approached this deployment with the intention of ensuring that the store would be designed by retailers and deployed by retailers, so that we would observe the real life implications of ISACS. We have partnered with Carrefour SA to deploy ISACS in a real setting. Due to privacy limitations the store was deployed inside of Carrefour's Headquarters, in France, with access only to employees or visitors of the company. The employees were informed and gave consent to be recorded in this store.

The store was designed in line with Carrefour's convenience store format. This means the number of products and variety truly reflects a real store, as well as its disposition in the store.

Store Layout

The store has 52, 0.5 meter wide, gondolas with 6 to 10 shelves each. There are 6 large refrigeration units with double open doors and 6-7 shelves each. This store is divided into several sections: fresh food, condiments, alcoholic drinks, regular drinks, snacks, candy, chips and convenience items. (See Fig.6.6).

All shelves are instrumented with the weight sensors defined in Sec. 6.1. In the ceiling there are 52 IP cameras, laid out according to Sec.6.1. All cameras and sensors are connected via Ethernet to a cabinet that holds our servers. These servers are not connected to the internet, and only connect to the retailer's POS (point of sale), so that ISACS can provide the receipt to the payment terminal.

Store Entry and Exit.

This is an open store setting, where there is no gate/door to enter the store, so anyone can get inside without any limitations. In order to present a receipt to a customer that shops in an autonomous store there has to be a place where a matching between the sensed person and the real person occurs. Therefore we have placed 2 payment terminals at the exit. One for self checkout, in case the person does not wish to be a part of the autonomous checkout experience, and a second one with a autonomous terminal. This terminal operates just like the self checkout providing similar payment methods, but automatically shows the receipt as the person approaches the terminal, using ISACS.

In our system we leverage the fact that customers are used to going to a checkout station at the end of their shopping journey to match the predicted receipt with their payment method. *ISACS* uses a virtual geo-fence around the checkout area to match the sensed customer with the real customer. It then presents the receipt in the payment terminal inside that area. This way the association is done at the moment of payment, reducing the need for any entry blocking mechanism. Similar to a self-checkout station where the customer does not need to scan the products. In most cases, when the cart is correct, the shopper chooses their method of payment (credit/debit card, loyalty card, apple/android pay, etc.), the shopper would then execute the payment and leave with the purchased items. In the cases where the predicted receipt was incorrect, the shopper would be able to ask for help (there was always a person by the payment terminals to assist shoppers). The employee providing help would enter a correction mode, and scan any missing items, or remove any extra items. This correction is then registered with ISACS and used for accuracy measurements.

6.3.2 Allowed Customer Shopping Journeys

There is a vast list of shopping journeys that customers can take inside a store. From taking a cart, or basket, and filling it up with items, or just carrying them around. Coming in alone or with their family. These are just a few of the options possible. ISACS focuses on the journeys present in convenience stores which are the ones in which the customer most values speed. This means that each customer is tracked individually and is charged by the items that he, or she, picks up from the shelves and does not put them back on any shelf. Passing items between people is outside the setting of ISACS. This means that unless the customer leaves items in the store in a place that is not a sensored shelf (i.e. floor or give them to someone else) ISACS will correctly predict what items the customers are carrying out of the store. In this context people tend to either carry the objects in their hands or put than into a small bag.

6.3.3 Experimental Setting and Dataset Characteristics

Due to the nature of the operating store, data privacy and infrastructure requirements –approximately 1.5TB/operating day of local storage capacity– we have not been able to evaluate our most recent approaches against all of the experiment. However we recorded a total of 2 weeks of data and have evaluated our multi-people matching with multiple objects in its multiple versions. In these 2 weeks of data there were 1874 transactions, each with approximately 5 interactions per transaction. There are a total of 7840 pick ups and 2093 put backs. These are real shoppers entering the store purchasing products for their own use, so no instructions were given to them and the number of items, duration of the trip and behavior is completely natural.

For the same data privacy and store requirements reasons this chapter does not address the impact of *customer education* throughout the length of the experiment. It was impossible for ISACS to track recurring customers given that every new customer that entered the store and became visible by the cameras was "initialized" as a new "anonymous track" without any association to an account. At the moment of payment, the "anonymous track" that stood in the geo-fenced area in front of the payment terminal was then sent to the retailer's payment system and no storing of the payment identity is ever made in the ISACS system.

This dataset was collected on the first 2 weeks of October 2020, using version 6 being operated (See Tables 6.1 and 6.2). Every interaction of the shoppers with the shelves was recorded and manually labelled with the following information: timestamp, which person interacted, how many people surrounding the event, distance of the people to the event, product(s) interacted with, quantity of products picked/putdown, basket prior to interaction, and final basket. The final basket was further validated with the POS information provided by the retailer.

During these two weeks the shoppers were not instructed any differently than before, leading to their natural behavior which was present for the totality of the experiment (13 months).

6.3.4 Multi-human to Multi-product Matching Metrics

In this chapter we evaluate the accuracy of the matching algorithm between multiple people and multiple products. We define average matching accuracy as, Avg. Match Accuracy for short, as:

$$\text{Avg. Match Accuracy} = \frac{\#\text{ correct person predicted}}{\#\text{ events}}(\%) \tag{6.11}$$

Its complement, the average matching error, can then be easily defined as:

$$\text{Avg. Match Error} = \frac{\#\text{ incorrect person predicted}}{\#\text{ events}}(\%) \tag{6.12}$$

Furthermore ISACS performance is dependant on the behavior of the customer, this means that there is a performance difference in the case a customer does multiple sequential events or multiple customers do a single event. In case of matching errors the latter would affect multiple receipts however the first would only affect 1 receipt. Therefore we further measure accuracy by looking at the final receipt accuracy and final item accuracy as such:

$$\text{Avg. Receipt Accuracy} = \frac{\#\text{ correct predicted receipts}}{\#\text{ receipts}}(\%) \tag{6.13a}$$

$$\text{Avg. Item Accuracy} = \frac{\#\text{ correct items receipts}}{\#\text{ items purchased}}(\%) \tag{6.13b}$$

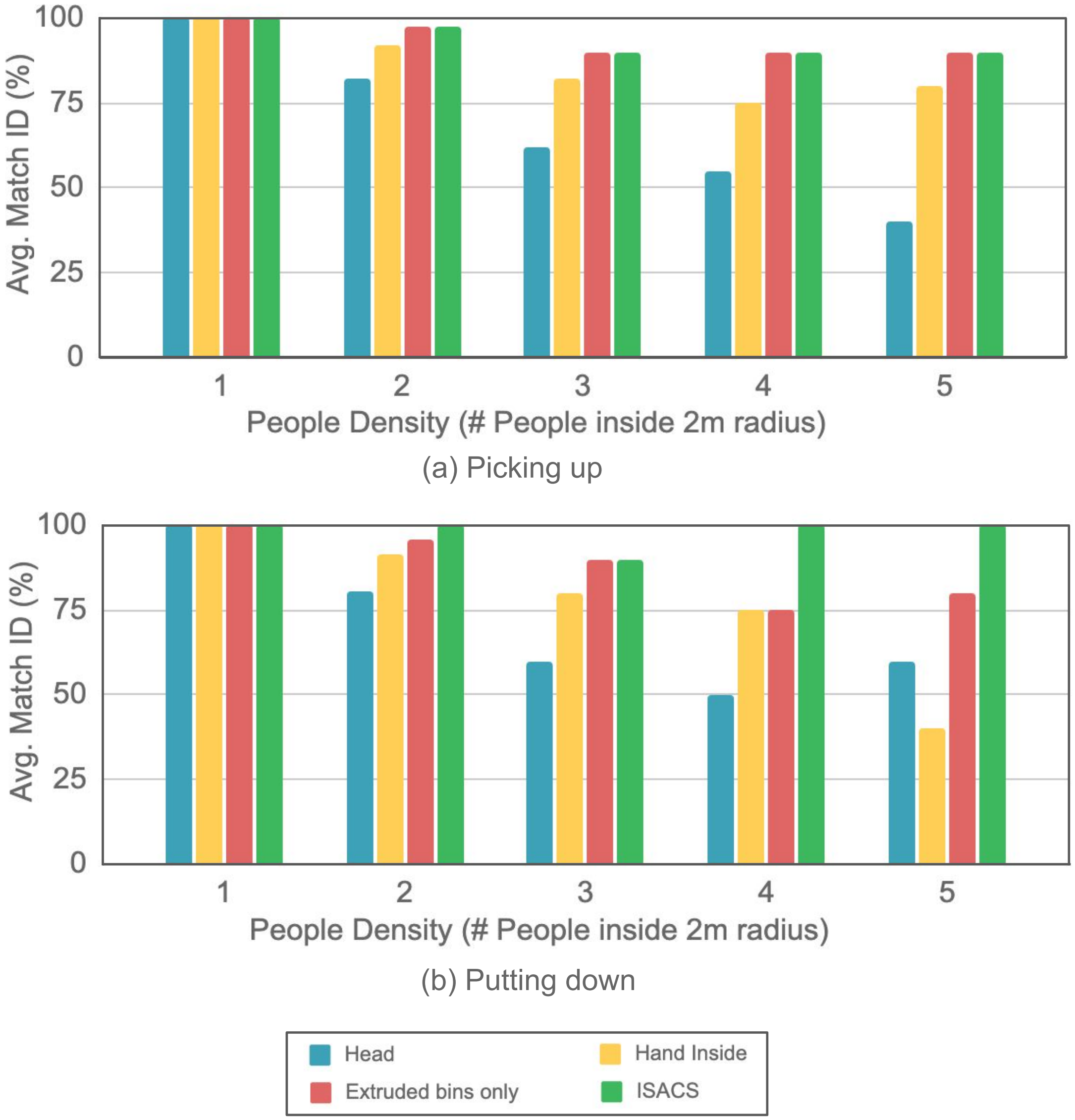
It is worth noting that the metrics in Equation 6.13 are not only dependent on the matching algorithm but are also affected by the accuracy of the Multi-Human 3D Pose Estimation and the Inventory Monitoring pipelines. This is therefore reflective of the overall ISACS framework performance. In the next subsection, we analyze our experiment results and the dependency of ISACS's Avg. Match Accuracy on different customer behaviors.

Ground truth was collected through the POS (point of sale) of the store. Furthermore, in order to ensure the quality and accuracy of the data there was an employee at all times present at the POS verifying the shoppers transactions and educating the customers. Given there were only 2 registers at the exit, and the small average basket size in this store, it was possible to maintain reliable transaction data.

6.3.5 Experiments at a Convenience Store Level Deployment

In order to fully evaluate the system in a real-world setting, we recorded the "store level" shopping behavior described above, and present the results here. We want to understand how the density of people around a pickup and a put down and the distance of the people affect the matching performance for ISACS.

One of the parameters that affects the performance of the matching algorithm is the amount of people surrounding the event as it happens. We define the number of people surrounding a event as the number of people within a radius of 2 meters of the center of the bin being interacted with. In Figure 6.7a we can see that when a person picks up items the association with the *head* (in blue) of that person suffers the most with the increase in density around the event. This accuracy drops from 82.35%, when there are 2 people, to 40% when there are 5 people around. However when leveraging the hand keypoints for the matching (in yellow) we observe a smaller effect of the density and a clear increase in accuracy, from 92.11% with only 2 people to a 80% with 5. The effect of density also affects the ability to proper estimate the hand location using the 3D Multi-Human Pose Estimation as it increases the natural occlusions in a small area. Furthermore the moment an event occurs is when the hand of the customer is mostly occluded by the upper shelves as it is inside the bin. Because this phenomenon occurs almost in all shelves, except in the top ones, the Extruded Bins only (Section 6.2.2) approach yields a higher accuracy with 97.27%



(a) Picking up

(b) Putting down

Figure 6.7: Multi-human to Multi-product Matching accuracy for different people densities around a: (a) *pick up event*, (b) *put down event*. The number of people are counted within a radius of 2 meters around the bin location of the event. In (a) ISACS's performance decreases with the increase in density. However in (b) ISACS's performs better by leveraging the information about the inventory the people have.

with only 2 people and 90% with the highest density observed, 5 people. During pick ups, the information provided by the inventory at hand does not add value to the decision, hence the accuracy observed with ISACS (in green) equals the Extruded Bins only approach (in red).

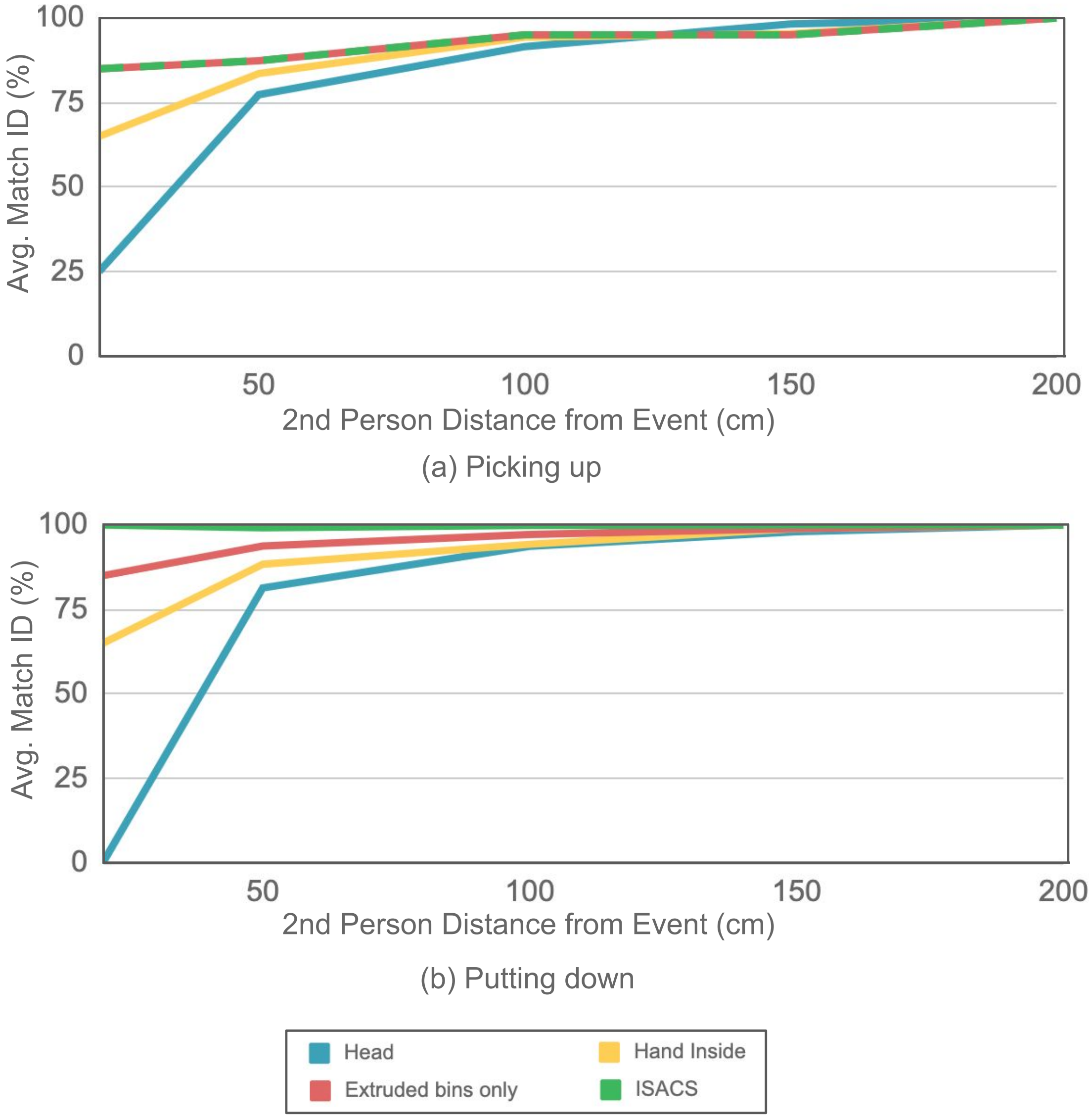
It is important to note that the number of events that occur at higher densities is much lower. There were only 10 pick ups and 5 put backs with 5 people around; and only 4 put backs with 4 people around. This happened during a demonstration of the store, and it is not a common occurrence in the dataset.

In Figure 6.7b we can see the equivalent metric for when people are putting items down. We can see that the accuracies follow an equivalent pattern to Fig. 6.7a, however in this case ISACS's performance gets an accuracy increase due to the use of the inventory at hand (Section 6.2.3) as a another source of information for deciding the person. ISACS's accuracy got to 100% in most cases except in the case of 3 people where it got to 90%. These failure cases occurred when 2 or 3 people are shopping together and pick the same products and then 1 of them puts them back. In this case the inventory at hand adds no value, leading to the same accuracy as the Extruded Bins approach.

Furthermore we observe on Figure 6.7b the *head* approach (in blue) performed at 60% while the *hands inside* only performed at 40%. This is observed only across 5 events. After investigation this occurs because in 1 case that hands of the person putting down the item are not seen by any angle when inside the shelf while the heads are clearly visible. The *Extruded bins* approach solves this situation by matching the right person slightly before the hand is occluded, preforming at 80% with 5 people around.

However it isn't only people density that affect the performance of the matching algorithm. The distance at which people stand from the event when multiple people are interacting can also significantly impact performance. As shown in Fig. 6.8a, **when multiple people interact with the shelves we consider the distance of the closest person to the event that *did not* perform the pick up or put down**. What we observe is that when people are very close to each other, closer than 50cm, we see a significant out-performance from ISACS (85% under 20cm and 87.4% between 20-50cm), when compared to the *head* (25% under 20cm and 77.3% between 20-50cm) and *hands inside* (65% under 20cm and 83.6% between 20-50cm) approach. During pick ups, there is no advantage of using the inventory information of what the person has in their hands (Section 6.2.3), therefore this approaches performance equals the one of the *Extruded bins only.*

On the other hand, during put downs, ISACS performance is unaffected by the distance of the closest person, performing consistently above 99.2% (See Fig. 6.8b).



(a) Picking up

(b) Putting down

Figure 6.8: Multi-human to Multi-product Matching accuracy for multiple people when: (a) *picking up*, (b) *putting down* close to each other. The distance is measured from the center of the *bin* being interacted with to the head of the closest person *not* interacting with the bin. In (a) ISACS's accuracy increases as people get further apart. However in (b) ISACS's accuracy is not affected by distance due to the added inventory information at the hand of the person.

6.4 Lessons in deploying stores with real customers

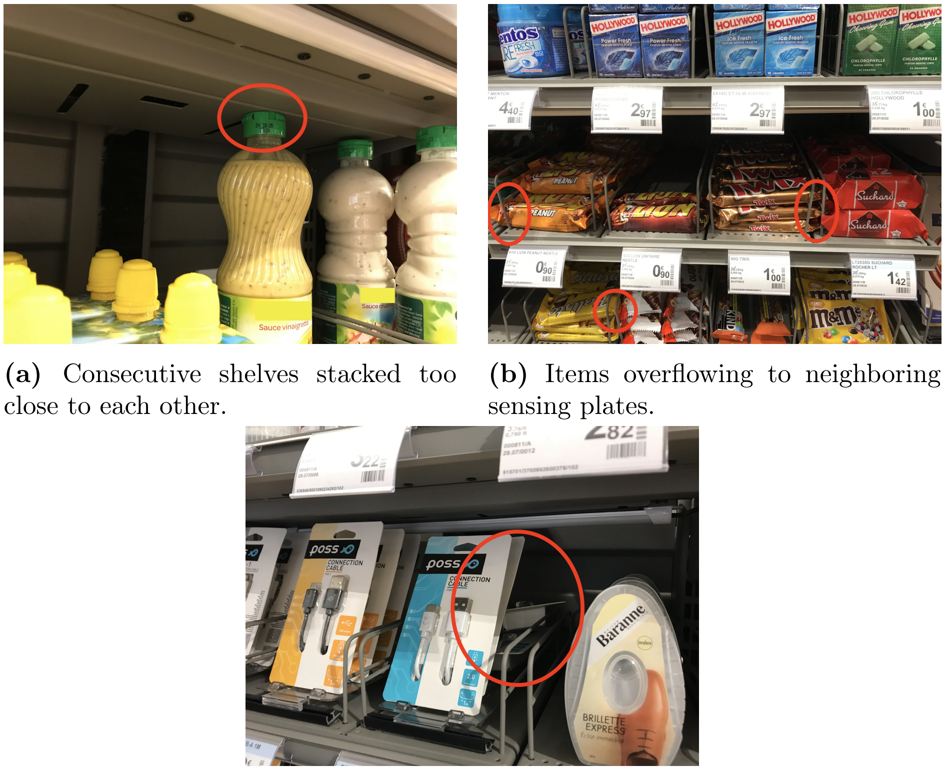
We have deployed and evaluated ISACS in an operating convenience store covering 800 square feet with 1653 distinct products, and more than 20000 items in a store owned and operated by Carrefour SA, in Massy France.

Over the course of 13 months, we gathered data and experience that allowed us to iterate over the hardware and software of the deployment, to withstand the realities of an operating convenience store and improve the autonomous store setup for higher accuracy. In this section we describe the design decisions taken to reduce the impact of setup or operations, based on our deployment experience as well as the lessons learned through multiple iterations of this project.

6.4.1 Unexpected ISACS's Operation Challenges

When deploying ISACS we realized that the Item Weight Model was challenging to obtain. Getting the operators of the store to weight every single item was not feasible. So we have initially approximated the weight distribution of the items by weighing 3 items of each type and generating a distribution for each type. This worked well for pre-packaged goods. However fresh food, condiments and light weighted items became difficult to identify and count accurately due to a much **higher variance in the weight distribution**. In order to accurately count and predict these items we have manually identified these *high variance* items and changed their weight distribution model accordingly. Furthermore we have trained the operator to identify these items and label them as such. Up until version 5 ISACS had a fixed variance threshold applied to every *Weight Change Bin Detection* (described in Sec. 6.1.2). After the first COVID lockdown, the *high variance* issue started to became more apparent given that the retailer started to sell more items that had higher variance, such as cups with pieces of cut fruit. These quickly became the most selling items, severely impacting the accuracy of the system.The ability to set a particular product as a *high variance* product was introduced to the restocking application in version 6 (See Table 6.2). This meant that employees prior to restocking would define products that had higher than 10% variance across units as *high variance* products.

Another requirement of ISACS is an understanding of the placement/location of the products in each shelf ($\mathcal{L}$, Section 6.1.4). This information is commonly available for large retailer which standardize their store layouts for store operations efficiency. However in this experimental store we were faced with a **continuously changing items' layout**. We observed fluctuations, in version 1 & 2 (See Table 6.1



(a) Consecutive shelves stacked too close to each other.

(b) Items overflowing to neighboring sensing plates.

(c) Items falling behind the shelf into neighboring sensing plates.

Figure 6.9: Unexpected Challenges from real-world deployment due to normal retailer operations.

and Fig. 6.10), of the accuracy of ISACS due to the changing of the items' layout without the respective change of the location model, $\mathcal{L}$ model in the system. This was solved by training the operator to change the $\mathcal{L}$ model by using a simple mobile application that allowed them to switch out a product, or place it in a different shelf by simply scanning it on the phone. The mobile application used did not change the system's performance. Ultimately "synchronization" between the location model, $\mathcal{L}$, and the physical reality was the impacting factor to the system. Once trained, the restocking employees were able to ensure an accurate sync between $\mathcal{L}$ and reality.

As time passes and the employees operate the store (restock and change product locations) more regularly we observed another decrease in accuracy. This was due to fact that the employees in order to increase the number of items available for sale, increasing the density, started to hit certain limitations of the system. Such as, placing items so close to each other that the physical separators between *bins* would

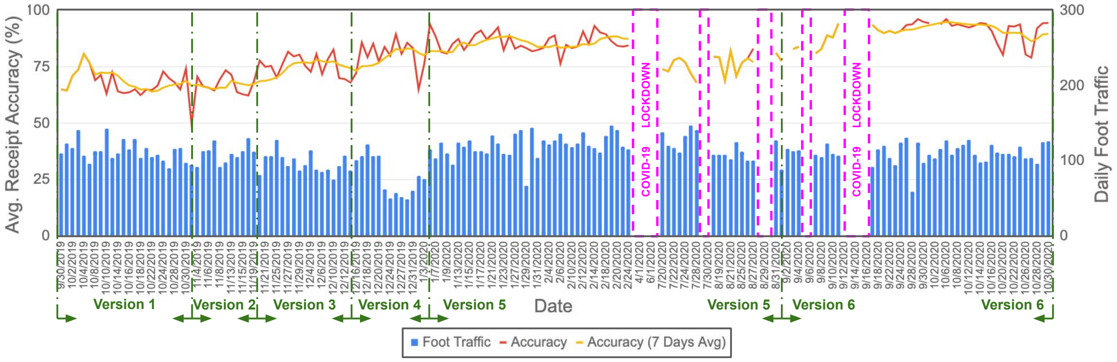


Figure 6.10: Receipt accuracy across 1 year of ISACS's deployment. The store was locked down due to COVID-19 from 24-Feb-2020 up to 20-Jul-2020, as well as, 4 smaller periods representing in light purple. ISACS's achieved up to 96.4% Receipt accuracy over the lifetime of the deployment.

actually not prevent **the object from laying partial weight on neighboring sensing plates** (See Figs. 6.9b-6.9c). This lead to incorrect weight measurement and inaccurate item counting. Further, the proximity of the shelves, in height, was decided based on the minimum distance possible so that the customer can see the product but also the retailer can fit as many as possibly vertically, notice how vertically *snug* the objects are in Fig. 6.6. This lead to situations where objects ended up *squeezed* in between 2 shelves, leading to a **increase in weight measurement due to the weight of the upper shelf** (See 6.9a). These issues were resolved with proper employee training and monitoring. You can observe these accuracy swings and steadily increases as we address the issues in Fig. 6.10.

Finally, ISACS does not take into account the restocking procedures. This initially meant that as an employee would restock the store, the system would be tracking them attempting to understand what they would pick up and put back. However employees behavior is far from a normal shopper: employees pick up a full shelf worth of goods to look at the expiration date, put back items that are not yet in the system to be monitored and also re-arrange the products orientation in the shelf to have the branding facing the client in a neat way. For the purpose of measuring the system we removed the carts generated for the employees.

6.4.2 Uncontrolled Customer Behavior

Ultimately the biggest challenge that ISACS faces is the uncontrolled behavior of the customers. There is a particularly challenging effect when deploying such a store: **the curiosity effect**. When customers are faced with this store for the first time they try to understand it, by stressing the system to the limits. **Pressing the sensors up and down, trying to hide the objects, running through the store, shopping extremely close to each other are just a few of the example behaviors we have observed.** This effect takes place more strongly in the beginning of the experiment. As time passes and the customers get used to the experience we have observed that those behaviors almost completely disappear.

In order to understand the impact of the accuracy presented in Section 6.3 we have measured the behavior of the shoppers, across the following dimensions:

- Average number of products in a receipt
- Average density of shoppers when interacting with a shelf
- Average distance of close-by shoppers to the interacted shelf

These variables affect ISACS performance both in people matching as well as inventory detection and counting. Figure 6.11 shows the number of people present –density– when an event (pick up/put down) occurs. We can observe that **most events occur with under 3 people close-by**. Figure 6.12a presents the distance of the close-by shoppers when an event occurs. This is showing that if an event has more than 1 person close-by, that person will be no further than 2 meters and as close as 20 centimeters. Notice the red line showing the threshold picked for 6.1, where people outside the radius of *2 meters* are not considered to be matched by ISACS.

6.4.3 Hardware Resilience

There are three types of hardware equipment that need to operate continuously in order to for ISACS to perform at its best: cameras, weight sensors and servers. While cameras and servers are quite a mature product easily available and maintainable the weight sensing shelves are not.

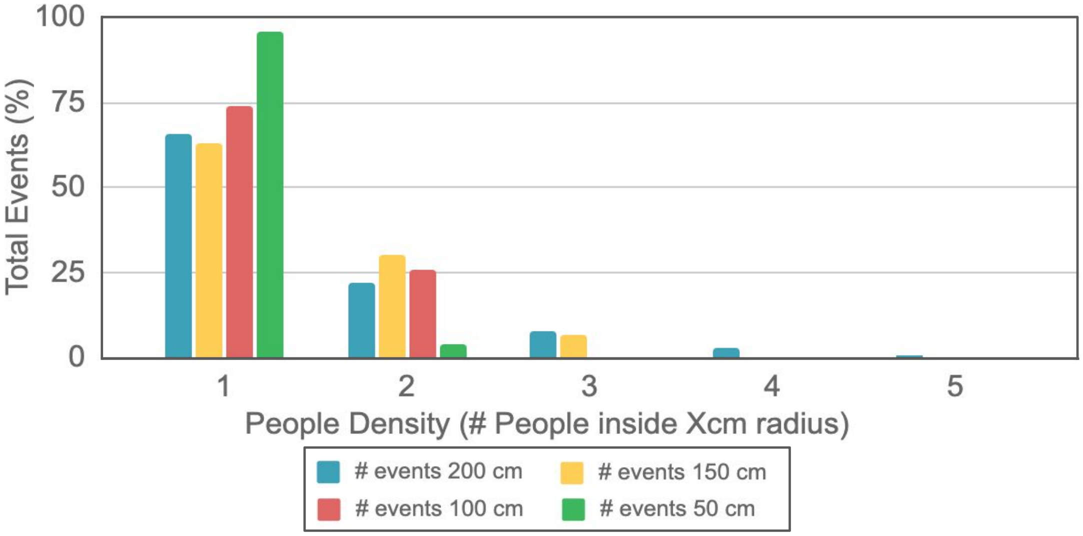
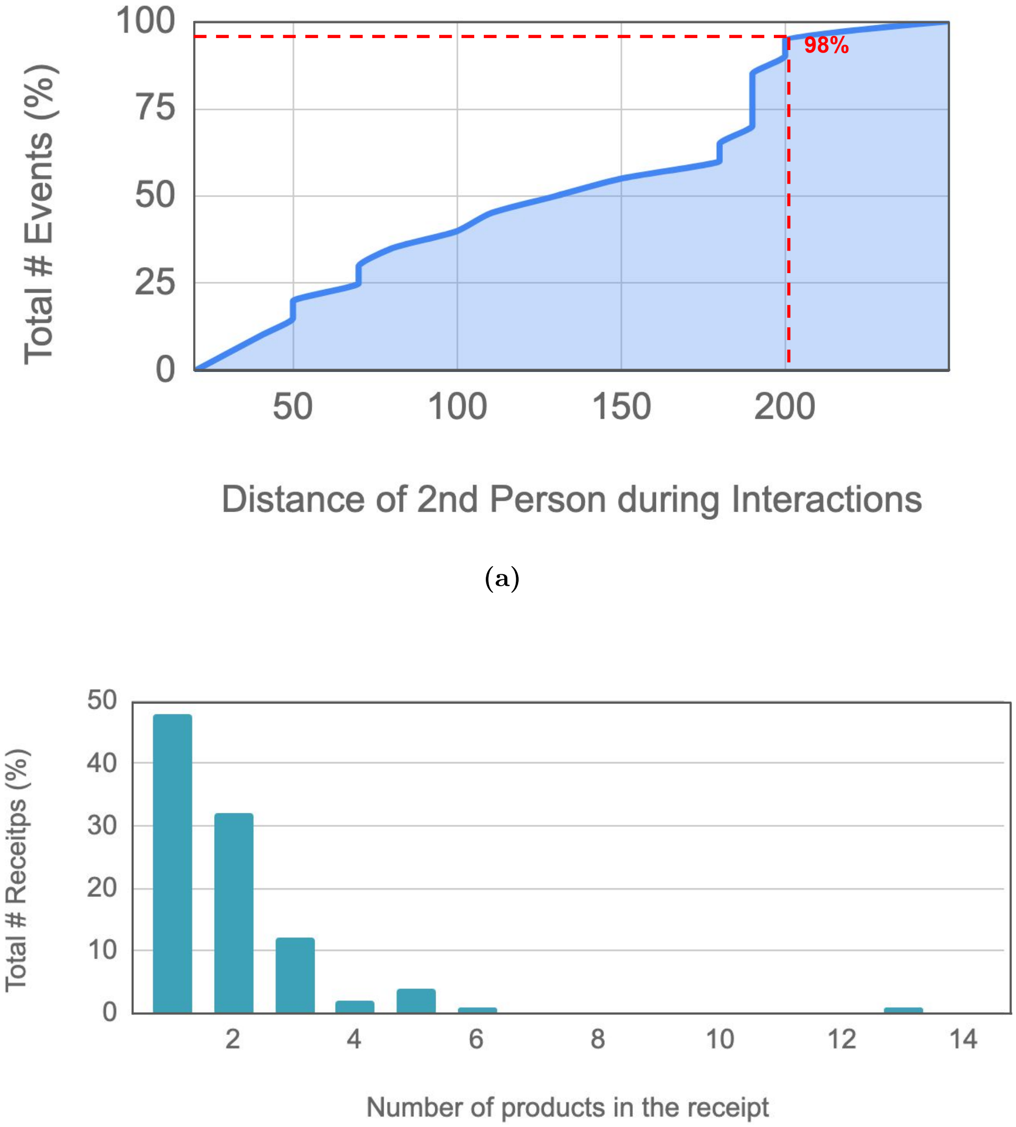


Figure 6.11: People density around a pick up or put down event during 1 year of operating ISACS. People density is measured as the % of events found with people within 200, 150, 100 and 50 cm (each color add up to 100%). Note that most interactions are with a single person, up to a maximum of 5 people within 2 meters.

Weight Sensing Shelves

This store contained 580 sensing plates, each of them connected to a control box, at the bottom of each gondola, which in turn was connected to the local network via Ethernet.

Given the lack of maturity of these custom sensing shelves, we faced some initial challenges in operating them. Particularly we observed that some shelves mistakenly would swap the sensing plate id with a different plate, for a few seconds, and then return to the original id (See Fig. 6.13). This would occur more often inside the refrigeration units given **the sensitivity of shelves to the temperature fluctuations**. The result of this issue would be 'ghost' weight detections of the products siting on those sensing plates. If by chance a customer was close-by he or she would get those objects attributed to them. This was observed before version 5 (See Table 6.1). This issue was caused by a design flaw in the sensing HW, where the identifier of the plate –*plateID– was being measured as an analog pin on the controller for each plate, based on a combination of resistor dividers. This meant that*



(a)

(b)

Figure 6.12: Deployment statistics of a 1 year real world operations. (a) Average distance between people when multiple people interact with the shelves at the same time (98% of the multiple people interaction happen under 2 meters). (b) Number of products per receipt. Average number of products taken is 1.83

depending on the tolerances of the resistors the plateID *would change if the resulting voltage would dip, or spike, due to the change in resistance caused by temperature effects.*

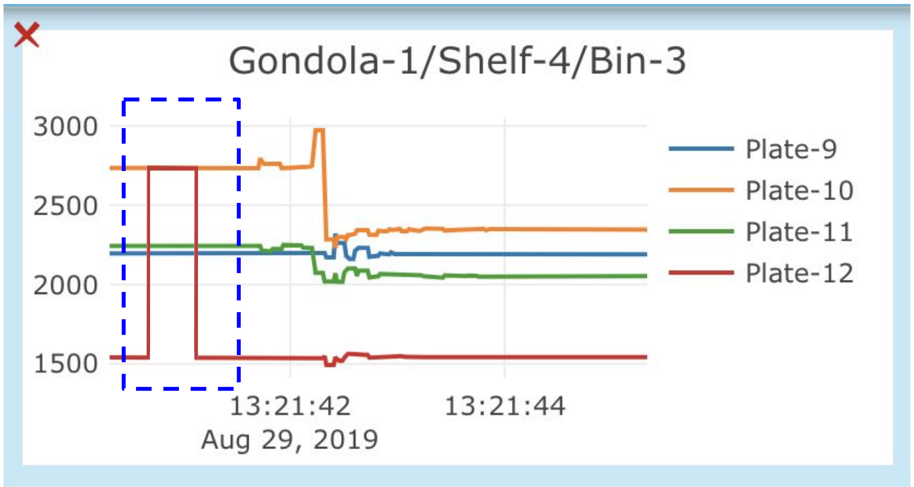


Figure 6.13: ISACS's weight sensor signal demonstrating the swapping of plate IDs being reported. Notice the inside the blue dashed box that plate 12 (in red) jump to the same line of plate 10 (in orange). This occurs because plate 10's id started to report as if it was plate 12's causing incorrect weight changes detected.

Upon realization of this issue, 2 solutions were developed: the *Max distance thresholding* –Sec. 6.1, and a packet reconstruction technique. The first approach removes any association of a person who would be too far to actually have been able to pick the products, while the second one would attempt to revert the ID swap before the signal reached the Weight Change detection module. This was accomplished by observing the continuity of the incoming signals and matching a instantaneous discontinuity with another discontinuity on sensing plate of the same shelf. These solutions were applied in Version 5 (See Table 6.1). A more proper solution, but highly costly, would have been to redesign the sensing HW to store its internal ID digitally and not be subject to environmental factors, and subsequently swap all shelves.

Furthermore, during the first 3 weeks of the deployment, prior to opening of the store, the shelves required significant attention and fixing due to the transport impact on the shelves. Many internal connections were loose, and some control boxes were not consistently working. However after the first iteration of shelf hardware fixing, these shelves remained consistently working for the following years with very minimal maintenance.

Table 6.1: Version history of ISACS throughout the 13 months of deployment, showing how multiple approaches of camera selections and the equivalent software changes required to accommodate the hardware changes.

ISACS	Hardware	Date Start	Date End
Version 1	34 RGB cameras	09-30-2019	10-30-2019
Version 2	v1 + 15 Top Down Intel RealSense D435	10-30-2019	11-19-2019
Version 3	Angled 34 RGB	11-19-2019	12-12-2019
Version 4	52 RGB Cameras 34/15 (Angled/Top down)	12-12-2019	1-5-2020
Version 5	v4 + plateID reconstruction	1-5-2020	8-31-2020
Version 6	Version 5	8-31-2020	10-30-2020

Table 6.2: Version history of ISACS throughout the 13 months of deployment, showing when different versions of ISACS were deployed and their respective durations.

ISACS	Software	Date Start	Date End
Version 1	Head keypoint tracking	09-30-2019	10-30-2019
Version 2	v1 + Hand Depth Processing	10-30-2019	11-19-2019
Version 3	v1 + Keypoint tracking with Hand prediction	11-19-2019	12-12-2019
Version 4	v3 + Top Down RGB Keypoint tracking for hands	12-12-2019	1-5-2020
Version 5	v4 + event distance threshold	1-5-2020	8-31-2020
Version 6	v5 + High variance items	8-31-2020	10-30-2020

Camera Selection and Placement

In placing cameras we have to consider 2 main aspects: people tracking and people interaction with shelves. For people tracking it is best to place cameras in the ceiling, given this is the place with the most unobstructed view into the people in the store. An interesting trade off to consider when instrumenting an autonomous store is the camera specification, that is, which king of visual sensor to use. Whether a structure light sensor,a time of flight sensor, stereo sensors or a simple 2D RGB sensor. With the recent advances in computer vision 2D RGB sensors are becoming more and more powerful. Coupling this with their ease of use and accessibility we have decided to use them for our application.

The cameras setup of this store went through 5 iterations as we experimented with the best performing setup. As detailed in 6.2 ISACS uses the head and hands of the person to make its matching prediction between the customers and the products. With the advancements in the state of the art on people tracking we upgraded the camera setup throughout the following versions:

Version 1 - Camera Setup. Initially we deployed cameras with that would cover every volume of the store, when empty, with at least 3 angles. This was a requirement for tracking people accurately through the triangulation technique present in [15]. This resulted in 34 cameras spread out throughout the ceiling of the store.

Version 2 - Depth and RGB Setup. As we started to observe the need for better matching between people and products, we realized the need for better understanding of the movement of the hands of the customers. Therefore we deployed 2 types of cameras: depth cameras –Intel RealSense d435–, and regular RGB cameras. The depth cameras were intended to enhance the hand position of the person as they interacted with the shelves. These were placed top-down in the ceiling aiming just in front of each gondola, in order to have the most unobstructed view of the place of interaction with the shelf. While the others followed the principle layed out in *Version 1.*

Version 3 - Angled RGB Camera Setup. Given the added complexity in depth cameras –added computation, network requirements, physical device stability– we decided to remove them and compute the hand location based solely on RGB 2D images. This was achieved using the approach in [15] with cameras placed on the ceiling angled between 20-70 degrees. This angled produced the required side images of people to which the models available were trained with.

Version 4 - Top Down and Angled RGB Camera Setup. We finally reached our last version by adding top down RGB cameras to our system, given that better models and more available data emerged and allowed the tracking modules to perform better with those angles, giving an particularly useful point of view. Top down cameras are the most useful to understand *who* is interacting with a shelf given that they suffer the least from occlusions.

Servers

ISACS runs on a server with 4 NVidia Tesla T4 GPUs. These hardware accelerator cards are used for running human detection from all of the available cameras in the store. In order to process the required frames for Multi-Human 3D tracking the GPUs are fully occupied. These servers stay in a cabinet inside the store, to allow for very low latency and provide the receipts instantly as the customer approaches the payment terminal. Due to the stability of server technology, these servers did not require any maintenance during the entire duration of the experiment.

6.4.4 ISACS Iterative Deployment Process

We have iterated through several approaches during the length of the experiment. These can be seen in Table 6.2 and Figure 6.10. This figure shows the impact of the changes in approach in the accuracy of the receipts.

It is worth noting that the daily fluctuation of accuracy occurs due to the uncontrolled nature of the human behavior, both from the employees as well as from the customers. When the store setup was properly re-mapped ($\mathcal{L}$, correctly estimated), or a new approach was deployed, we can only confidently observe the improvements over a multi day trend. The orange line represents the 7 day trailing accuracy. In this line we can observe the upwards trend of the accuracy given every new iteration of the approach.

The process of upgrading the system was lead by the previous versions' results. Each version would run for a minimum of 1 month to validate the accuracy impact of that version's changes. Any upgrades to the system were done during closed hours of the store. This meant that the deployment process would not impact the normal operations of the store.

Over the course of 1 year of operation, ISACS achieved a receipt daily accuracy of up to 96.4%. Which translates to a 3.5x improvement over prior reported self-checkout accuracies.

6.5 Privatized Retail Systems Discussion

Amazon has shown to the world how autonomous stores can look like through their Amazon's Go technology. These stores provide an opportunity for the customer to shop in the store by downloading the Amazon application scanning the app to enter and "just walk out", with their receipt arriving within the next few hours [24].

While this has validated the customer experience of "just walking out" it unfortunately has been all done under closed doors, preventing the broader scientific community from leveraging the learnings of such stores and expanding more broadly beyond Amazon. Furthermore there are several media articles speculating how Amazon operates their Go's stores [106]. These [82] claim that Humans are leveraged in the cloud to perform the final checkout for the customer, essentially displacing the cashier workload to a remote one.

Furthermore, Amazon Go's stores favors a particular demographic group: upper-class, tech-savvy, and younger generation. The requirement of an application to enter and shop and the delayed receipt upon exit becomes quite restrictive for a more broad community to adopt. Shoppers who are more budget conscious usually come with the mindset of how much they will spend for their shopping and require validation before leaving the store. American consumer patterns are changing and discount retailers are seeing an increased adoption from a larger population sample which is more budget conscious [122]. To serve the broader community ISACS's presents an in-store receipt, despite the delayed receipt by Amazon. This is driven by the fact that in-store discounts and the "smart-shopper feelings" towards pricing act as a major component of the emotional response affecting shoppers' behavior to favor in-store price confirmation [18, 77, 96].

6.6 Autonomous Checkout Background and Related Work

ISACS's proposes a system that generates a receipt autonomously, i.e. no human interaction, while the customer is inside the store by matching multiple people with multiple item pick up/put down events. To achieve this, ISACS's system contributes to multiple fields of research, such as: multi-human tracking, object recognition and tracking, methods for identifying retail products, 3D scene reconstruction and sensor fusion. In this section we discuss existing works in these fields and how ISACS's leverages these prior works to enable Autonomous retail.

6.6.1 Autonomous Retail Enablers

The recent advancements of computer vision research in object recognition help enable applications such as autonomous stores. These advancements can be grouped into three main approaches: vision-only, a combination of computer vision with additional sensors, or a combination of computer vision with sensors and human product recognition.

Vision only

These solutions face two critical challenges: data availability for training and occlusions. The latest work using deep learning techniques to do object recognition –Self-attention-based [119], Mask R-CNN [41], Center-Net [25], NormalNet [116], FoveaBox [50]– require a large amount of labelled data on every individual product for training which is highly labor intensive and quickly becomes impractical. Some solutions have alleviated this problem by automating this process either through the use of sensors [91] or semi-supervised human labelling [107].

An added challenge for vision-only approaches are objects such as fruits and vegetables. There are works such as [30] that focus on identification of these kinds of products. Due to their visual variable nature, the approach is only accurate for a small subset of fruits and vegetables. Furthermore this only identifies the item, but doesn't natively handle accurate counting or items sold by the weight –as is the case for most fruits and vegetables.

The above approaches are also very computationally intensive, which can be limiting either in the excessive hardware provisioned in the store, or with the cost of such computation in the cloud. Approaches like [34] attempt to simplify the problem by leveraging Optical Character Recognition (a lower computationally intensive approach), to identify the objects by their label. RGB-D cameras have also been used to attempt at reducing computation and increasing accuracy of object detection [43] at a prohibitive higher infrastructure cost.

Furthermore these approaches focus mainly on locating and classifying objects in 2D images [95], which provides no information about the motion of the object, and the owner of the object.

Vision with Additional Sensors

Works such as [28, 56, 89, 90] combine however multiple 2D views with additional weight sensors, similar to ISACS, to generate an understanding of which objects are being picked up or put back down. However these works are limited by only one person interacting with one item at a time and are limited to a small amount of products, preventing them to scale to a store-level deployment. Other sensors have been explored for object identification such as RFID [60, 84, 88, 125], vibration sensors for people detection and tracking [67, 68] and inertial wearable sensors [92] for product to people matching, however these solutions have not shown to be practical in terms of cost, deployment (requiring customer to wear a device) or are not accurate enough for the low margins a store needs to operate.

Vision, Sensors and Human Product Recognition

Given the high accuracy required to operate a profitable convenience store, companies like Amazon have leveraged humans-in-the-loop in their cashier-less stores [1, 82, 106]. This approach is quite restrictive: delaying the delivery of the receipt prevents shoppers who are budget conscious from verifying their receipt before leaving the store, affecting the lower end of the social economic spectrum. American consumer patterns are changing, and discount retailers are seeing an increased adoption from a larger population sample that is more budget conscious [122]. COVID-19 has further pushed this behavior and driven shoppers' behavior towards a more targeted spending [17]. Furthermore, the majority of shoppers make purchases based on in-store discounts and the "smart-shopper feelings" towards pricing, discounts and promotions, act as a major component of the emotional response affecting shoppers' behavior to favor in-store price confirmation [18, 77, 96]

6.6.2 Retail Technology

There are several works in the computer vision, robotic and sensing fields that have pushed forward the retail technology domain. Robots that grab different kinds of objects are faced with similar challenges in identifying objects [49]. These solutions provide visual understanding of the objects in a *fast* manner even withing a cluttered environment [57]. Others provide a multi-view approach for object pose understanding, in this case for the purpose of robotic manipulation [53].

Counting and tracking people without interfering with their normal behavior is a relevant problem that spans beyond the Autonomous Checkout domain. There are works [13, 15, 23, 59] that focus on leveraging cameras to track people continuously across multiple camera views [15], focused on dense environments [59] or on the ability to reconstruct the 3D motion of people [104] in order to understand their behavior.

While each of these works alone does not address the autonomous checkout domain they share common challenges in visual understanding of the scene (store), objects (products) and people (shoppers).

6.7 Chapter Summary

In this chapter, we presented ISACS (In-Store Autonomous Checkout System) for Retail. Utilizing a centimeter accurate physical model of the store, multi-human 3D pose estimation and live inventory monitoring through weight sensors, ISACS is able to match multiple people interacting with multiple products, maintain an accurate store setup and provide a receipt within 2 seconds. ISACS has a receipt accuracy of up to 96.4%, without a human in the loop, which is a 3.5x reduction in error compared to the 86% accuracy reported for self-checkout stations.

This chapter demonstrates the performance of ISACS in a convenience store where most transactions are done at an individual level. Therefore, the assumption present in the chapter of: *the last customer who removes a product form a store's* *shelf and leaves with it is the customer charged for that product* is reasonable. However, it would be possible to relax this constraint by focusing on events that occur in the entire store (i.e., floor or between people), rather than solely on the shelves. This however presents an added challenge of where to focus the sensors and cameras in order to properly identify these events, something that we intend to address in our future work.

ISACS is the first fully autonomous system deployed in a convenience sized store that leverages computer vision and sensors to provide a in-store receipt, without relying on humans-in-the-loop.

Chapter 7

Conclusion

In this work I proposed a framework for combining multi-modality sensing with scene modeling in order to improve human-object characterization demonstrated in the context of autonomous retail.

I've shown that multi-sensing modalities such as weight and vision can be combined with physical models of the scene to improve the identification of the objects. This was show to outperform state of the art, in retail stores, by 2x error reduction in object identification. Furthermore this work demonstrates that the sensors and cameras can be combined with a model of the interactions of the humans with the objects to enhance the matching between humans and their respective interactions even in a highly dense environment. This approach outperformed prior work by 3x reduction in erroneous automatically generated receipts. And finally I propose as my future work to investigate the capabilities of combining a model of the relationships between objects and humans to improve the human-object interaction characterization under the constraint of dynamic objects.

Bibliography

[1] Amazon. Amazon.com: Amazon go, https://www.amazon.com/. Accessed: 2019-04-10. Cited on pages 35, 39, 64, and 100.

[2] ECR An. Self-checkout in retail: Measuring the loss, 2018, Acessed: 2021-11-12. Cited on page 65.

[3] Gwon Hwan An, Siyeong Lee, Min-Woo Seo, Kugjin Yun, Won-Sik Cheong, and Suk-Ju Kang. Charuco board-based omnidirectional camera calibration method. *Electronics*, 7(12):421, 2018. Cited on page 70.

[4] Adrian Beck. Self-scan checkouts and retail loss: Understanding the risk and minimising the threat. *Security Journal*, 24(3):199–215, 2011. Cited on pages xiv, 9, 58, and 59.

[5] Sima Behpour, Kris M Kitani, and Brian D Ziebart. Ada: Adversarial data augmentation for object detection. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1243–1252. IEEE, 2019. Cited on page 30.

[6] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. *2016 IEEE International Conference on Image Processing (ICIP)*, Sep 2016. Cited on page 30.

[7] Miroslaw Bober. Mpeg-7 visual shape descriptors. *IEEE Transactions on circuits and systems for video technology*, 11(6):716–719, 2001. Cited on page 46.

[8] Eleonora Bottani, Massimo Bertolini, Antonio Rizzi, and Giovanni Romagnoli. Monitoring on-shelf availability, out-of-stock and product freshness through rfid in the fresh food supply chain. *International Journal of RF Technologies*, 8(1-2):33–55, 2017. Cited on page 34.

[9] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. Albumentations: Fast and flexible image augmentations. *Information*, 11(2), 2020. Cited on pages xii, 27, and 28.

[10] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. In *arXiv preprint arXiv:1812.08008*, 2018. Cited on page 43.

[11] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018. Cited on page 75.

[12] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. Cited on page 43.

[13] Marco Carraro, Matteo Munaro, Jeff Burke, and Emanuele Menegatti. Real-time marker-less multi-person 3d pose estimation in rgb-depth camera networks. In *International Conference on Intelligent Autonomous Systems*, pages 534–545. Springer, 2018. Cited on page 101.

[14] Ching-Hang Chen, Ambrish Tyagi, Amit Agrawal, Dylan Drover, Stefan Stojanov, and James M Rehg. Unsupervised 3d pose estimation with geometric self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5714–5724, 2019. Cited on page 30.

[15] Long Chen, Haizhou Ai, Rui Chen, Zijie Zhuang, and Shuang Liu. Cross-view tracking for multi-human 3d pose estimation at over 100 fps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3279–3288, 2020. Cited on pages 59, 75, 96, and 101.

[16] Francois Chollet. Building powerful image classification models using very little data. *Keras Blog*, 2016. Cited on pages 27 and 28.

[17] McKinsey & Company. Redefining value and affordability in retail's next normal, https://www.mckinsey.com/industries/retail/our-insights/redefining-value-and-affordability-in-retails-next-normal. Accessed: 2021-05-15. Cited on page 100.

[18] B. C. Cotton and Emerson M. Babb. Consumer response to promotional deals.

Journal of Marketing, 42(3):109–113, 1978. Cited on pages 98 and 101.

[19] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019. Cited on page 30.

[20] Daan de Boer. The impact of rfid-deployments on out-of-stocks in various apparel stores. *TU Delft*, 2018. Cited on page 34.

[21] Statista Research Department. Retail e-commerce sales worldwide from 2014 to 2023(in billion u.s. dollars), https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/. Accessed: 2020-11-15. Cited on page 63.

[22] Statista Research Department. Total retail sales worldwide from 2018 to 2022(in trillion u.s. dollars), https://www.statista.com/statistics/443522/global-retail-sales/. Accessed: 2020-11-15. Cited on page 63.

[23] Junting Dong, Wen Jiang, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Fast and robust multi-person 3d pose estimation from multiple views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7792–7801, 2019. Cited on page 101.

[24] Meiling Du. Examining the user experience of amazon go shopping — just walk out, Accessed: 2019-11-15. Cited on pages 65 and 98.

[25] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. Cited on pages 10, 61, and 99.

[26] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Modeling visual context is key to augmenting object detection datasets. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 364–380, 2018. Cited on pages 22 and 30.

[27] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1301–1310, 2017. Cited

on page 30.

[28] João Falcão, Carlos Ruiz, Shijia Pan, Hae Young Noh, and Pei Zhang. Faim: Vision and weight sensing fusion framework for autonomous inventory monitoring in convenience stores. *Frontiers in Built Environment*, 6:175, 2020. Cited on pages 16, 70, and 100.

[29] João Diogo Falcão, Carlos Ruiz, Adeola Bannis, Hae Young Noh, and Pei Zhang. Isacs: In-store autonomous checkout system for retail. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(3):1–26, 2021. Cited on pages 16 and 17.

[30] Frida Femling, Adam Olsson, and Fernando Alonso-Fernandez. Fruit and vegetable identification using machine learning for retail applications. In *2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pages 9–15. IEEE, 2018. Cited on page 99.

[31] Teresa Fernandes and Rui Pedroso. The effect of self-checkout quality on customer satisfaction and repatronage in a retail context. *Service Business*, 11(1):69–92, 2017. Cited on page 65.

[32] Forrester. Consumers cringe at slow checkout, Accessed: 2019-11-15. Cited on page 63.

[33] Dini Fronitasari and Dadang Gunawan. Palm vein recognition by using modified of local binary pattern (lbp) for extraction feature. In *2017 15th International Conference on Quality in Research (QiR): International Symposium on Electrical and Computer Engineering*, pages 18–22. IEEE, 2017. Cited on page 61.

[34] Petia Georgieva and Pei Zhang. Optical character recognition for autonomous stores. In *2020 IEEE 10th International Conference on Intelligent Systems (IS)*, pages 69–75. IEEE, 2020. Cited on page 100.

[35] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. Cited on page 46.

[36] Peter Godlewski. Automated inventory system and method, January 19 2010. US Patent 7,650,298. Cited on page 60.

[37] Grabango. Eliminate lines and save people time, https://grabango.com/. Ac-

cessed: 2019-04-10. Cited on page 64.

[38] Craig B Green. Weight sensing apparatus and methods for inventory control, April 14 2009. US Patent 7,518,068. Cited on page 60.

[39] Thomas W Gruen, Daniel S Corsten, and Sundar Bharadwaj. *Retail out-of-stocks: A worldwide examination of extent, causes and consumer responses*. Grocery Manufacturers of America Washington, DC, 2002. Cited on page 34.

[40] Lili Guo, Dan Xu, and Zhenping Qiang. Background subtraction using local svd binary pattern. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 86–94, 2016. Cited on page 43.

[41] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. Cited on pages 10, 43, 46, 61, and 99.

[42] Stefan Hinterstoisser, Vincent Lepetit, Paul Wohlhart, and Kurt Konolige. On pre-trained image features and synthetic images for deep learning, 2017. Cited on page 30.

[43] Tomáš Hodaň, Xenophon Zabulis, Manolis Lourakis, Štěpán Obdržálek, and Jiří Matas. Detection and fine 3d pose estimation of texture-less objects in rgb-d images. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4421–4428. IEEE, 2015. Cited on page 100.

[44] Mahdi Hussain. Convenience store loss prevention - the complete guide, https://petrooutlet.com/blog/posts/convenience-store-loss-prevention-guide/. Accessed: 2019-11-15. Cited on page 64.

[45] Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Godisart, Bart Nabbe, Iain Matthews, et al. Panoptic studio: A massively multiview system for social interaction capture. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):190–204, 2017. Cited on page 75.

[46] Rena Kamoda, Mayumi Ueda, Takuya Funatomi, Masaaki Iiyama, and Michihiko Minoh. Grocery re-identification using load balance feature on the shelf for monitoring grocery inventory. In *Computers workshop (CwC)*, page 13. Citeseer, 2012. Cited on page 61.

[47] Whoi-Yul Kim and Yong-Sung Kim. A region-based shape descriptor using zernike moments. *Signal processing: Image communication*, 16(1-2):95–102, 2000. Cited on page 54.

[48] Marcus Klasson, Cheng Zhang, and Hedvig Kjellström. A hierarchical grocery store image dataset with visual and semantic labels. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 491–500. IEEE, 2019. Cited on page 62.

[49] Ellen Klingbeil, Deepak Rao, Blake Carpenter, Varun Ganapathi, Andrew Y Ng, and Oussama Khatib. Grasping with application to an autonomous checkout robot. In *2011 IEEE international conference on robotics and automation*, pages 2837–2844. IEEE, 2011. Cited on page 101.

[50] Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, Lei Li, and Jianbo Shi. Foveabox: Beyound anchor-based object detection. *IEEE Transactions on Image Processing*, 29:7389–7398, 2020. Cited on pages 10, 61, and 99.

[51] Ronald B Larson. Supermarket self-checkout usage in the united states. *Services Marketing Quarterly*, 40(2):141–156, 2019. Cited on page 65.

[52] Tian Li and Hongtao Zhang. Information sharing in a supply chain with a make-to-stock manufacturer. *Omega*, 50:115–125, 2015. Cited on page 34.

[53] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, 2018. Cited on page 101.

[54] Chen-Hsuan Lin, Ersin Yumer, Oliver Wang, Eli Shechtman, and Simon Lucey. St-gan: Spatial transformer generative adversarial networks for image compositing, 2018. Cited on pages 22 and 30.

[55] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. Cited on pages xiii and 47.

[56] Lizheng Liu, Bo Zhou, Zhuo Zou, Shih-Ching Yeh, and Lirong Zheng. A smart unstaffed retail shop based on artificial intelligence and iot. In *2018 IEEE 23rd International workshop on computer aided modeling and design of*

communication links and networks (CAMAD), pages 1–4. IEEE, 2018. Cited on pages 10 and 100.

[57] Ming-Yu Liu, Oncel Tuzel, Ashok Veeraraghavan, Yuichi Taguchi, Tim K Marks, and Rama Chellappa. Fast object localization and pose estimation in heavy clutter for robotic bin picking. *The International Journal of Robotics Research*, 31(8):951–973, 2012. Cited on page 101.

[58] Weijie Liu, Nishuang Liu, Yang Yue, Jiangyu Rao, Cheng Luo, Hang Zhang, Congxing Yang, Jun Su, Zhitian Liu, and Yihua Gao. A flexible and highly sensitive pressure sensor based on elastic carbon foam. *Journal of Materials Chemistry C*, 6(6):1451–1458, 2018. Cited on page 60.

[59] Xiaobai Liu. Multi-view 3d human tracking in crowded scenes. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 3553–3559, 2016. Cited on page 101.

[60] Xiaochen Liu, Yurong Jiang, Kyu-Han Kim, and Ramesh Govindan. Grab: Fast and accurate sensor processing for cashier-free shopping. *arXiv preprint arXiv:2001.01033*, 2020. Cited on pages 10 and 100.

[61] Graciela Lara López, Adriana Peña Pérez Negrón, Angélica De Antonio Jiménez, Jaime Ramírez Rodríguez, and Ricardo Imbert Paredes. Comparative analysis of shape descriptors for 3d objects. *Multimedia Tools and Applications*, 76(5):6993–7040, 2017. Cited on page 61.

[62] Sehla Loussaief and Afef Abdelkrim. Deep learning vs. bag of features in machine learning for image classification. In *2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*, pages 6–10. IEEE, 2018. Cited on page 61.

[63] Bangalore S Manjunath, J-R Ohm, Vinod V Vasudevan, and Akio Yamada. Color and texture descriptors. *IEEE Transactions on circuits and systems for video technology*, 11(6):703–715, 2001. Cited on page 46.

[64] Melissa A Matthews and Tanya M Horacek. Vending machine assessment methodology. a systematic review. *Appetite*, 90:176–186, 2015. Cited on page 10.

[65] Christian Metzger, Jan Meyer, Elgar Fleisch, and Gerhard Tröster. Weight-sensitive foam to monitor product availability on retail shelves. In *Interna-*

tional Conference on Pervasive Computing, pages 268–279. Springer, 2007. Cited on page 60.

[66] Matthew Meyer. Out of stock sensor, March 1 2016. US Patent 9,275,361. Cited on page 60.

[67] Mostafa Mirshekari, Jonathon Fagert, Shijia Pan, Pei Zhang, and Hae Young Noh. Step-level occupant detection across different structures through footstep-induced floor vibration using model transfer. *Journal of Engineering Mechanics*, 146(3):04019137, 2020. Cited on pages 10 and 100.

[68] Mostafa Mirshekari, Shijia Pan, Jonathon Fagert, Eve M Schooler, Pei Zhang, and Hae Young Noh. Occupant localization using footstep-induced structural vibration. *Mechanical Systems and Signal Processing*, 112:77–97, 2018. Cited on pages 10, 59, and 100.

[69] Eduardo de Araujo Moretti, Rosley Anholon, Izabela Simon Rampasso, Dirceu Silva, Luis Antonio Santa-Eulalia, and Paulo Sérgio de Arruda Ignácio. Main difficulties during rfid implementation: An exploratory factor analysis approach. *Technology Analysis & Strategic Management*, 31(8):943–956, 2019. Cited on page 34.

[70] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 891–898, 2014. Cited on page 30.

[71] Adis Imam Munandar and Biantara Albab. Disruption of the workforce in the digital era: A smart store case study. In *Managing Learning Organization in Industry 4.0*, pages 116–120. Routledge, 2020. Cited on page 9.

[72] Kazuya Murao, Junna Imai, Tsutomu Terada, and Masahiko Tsukamoto. Recognizing activities and identifying users based on tabletop activities with load cells. In *Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services*, page 39. ACM, 2015. Cited on page 60.

[73] Kazuya Murao, Junna Imai, Tsutomu Terada, and Masahiko Tsukamoto. Activity recognition and user identification based on tabletop activities with load

cells. *Journal of Information Processing*, 25:59–66, 2017. Cited on pages 40 and 60.

[74] Jeanne Nagle. *Working with Tech in Retail*. The Rosen Publishing Group, Inc, 2020. Cited on page 9.

[75] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016. Cited on page 43.

[76] Stephen O'Hara and Bruce A Draper. Introduction to the bag of features paradigm for image classification and retrieval. *arXiv preprint arXiv:1101.3354*, 2011. Cited on page 61.

[77] Bryan (Forbes) Pearson. How retailers can maximize the power of coupons, https://www.forbes.com/sites/bryanpearson/2017/03/15/research-reveals-how-retailers-can-maximize-the-power-of-coupons/?sh=4873e6532f01. Accessed: 2019-11-15. Cited on pages 98 and 101.

[78] Xi Peng, Zhiqiang Tang, Fei Yang, Rogerio S Feris, and Dimitris Metaxas. Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2226–2234, 2018. Cited on page 30.

[79] Siyuan Qiao, Wei Shen, Weichao Qiu, Chenxi Liu, and Alan Yuille. Scalenet: Guiding object proposal generation in supermarkets and beyond. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1791–1800, 2017. Cited on page 62.

[80] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 413–420, 2009. Cited on page 26.

[81] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. *openreview.net*, 2016. Cited on page 20.

[82] Recode. Amazon's store of the future has no cashiers, but humans are watching from behind the scenes, https://www.recode.net/2017/1/6/14189880/amazon-go-convenience-store-computer-vision-humans. Accessed: 2019-04-10. Cited on pages 10, 35, 39,

64, 98, and 100.

[83] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. Cited on pages 46 and 61.

[84] Yacine Rekik, Evren Sahin, and Yves Dallery. Analysis of the impact of the rfid technology on reducing product misplacement errors at retail stores. *International Journal of Production Economics*, 112(1):264–278, 2008. Cited on pages 10 and 100.

[85] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. Cited on pages xiii, 26, 43, 46, 47, 54, and 61.

[86] Stephan R. Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks, 2017. Cited on page 31.

[87] Mahsan Rofouei, Wenyao Xu, and Majid Sarrafzadeh. Computing with uncertainty in a smart textile surface for object recognition. In *2010 IEEE Conference on Multisensor Fusion and Integration*, pages 174–179. IEEE, 2010. Cited on page 60.

[88] George Roussos. Enabling rfid in retail. *Computer*, 39(3):25–30, 2006. Cited on pages 10 and 100.

[89] Carlos Ruiz, Joao Falcao, Shijia Pan, Hae Young Noh, and Pei Zhang. Aim3s: Autonomous inventory monitoring through multi-modal sensing for cashier-less convenience stores. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pages 135–144, 2019. Cited on pages 16, 17, 65, and 100.

[90] Carlos Ruiz, Joao Falcao, Shijia Pan, Hae Young Noh, and Pei Zhang. Autonomous inventory monitoring through multi-modal sensing (aim3s) for cashier-less stores. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pages 395–396, 2019. Cited on page 100.

[91] Carlos Ruiz, Joao Falcao, and Pei Zhang. Autotag: visual domain adaptation

for autonomous retail stores through multi-modal sensing. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, pages 518–523, 2019. Cited on pages 16, 20, 61, and 99.

[92] Carlos Ruiz, Shijia Pan, Adeola Bannis, Ming-Po Chang, Hae Young Noh, and Pei Zhang. Idiot: Towards ubiquitous identification of iot devices through visual and inertial orientation matching during human activity. In *2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 40–52. IEEE, 2020. Cited on pages 10, 60, and 100.

[93] Faten Sandid and Ali Douik. Robust color texture descriptor for material recognition. *Pattern Recognition Letters*, 80:15–23, 2016. Cited on pages 46 and 54.

[94] Bikash Santra and Dipti Prasad Mukherjee. A comprehensive survey on computer vision based approaches for automatic identification of products in retail store. *Image and Vision Computing*, 86:45–63, 2019. Cited on page 64.

[95] Silvio Savarese and Li Fei-Fei. 3d generic object categorization, localization and pose estimation. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007. Cited on page 100.

[96] Robert M Schindler. The excitement of getting a bargain: some hypotheses concerning the origins and effects of smart-shopper feelings. *ACR North American Advances*, 1989. Cited on pages 98 and 101.

[97] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015. Cited on page 19.

[98] Albrecht Schmidt, Martin Strohbach, Kristof Van Laerhoven, Adrian Friday, and Hans-Werner Gellersen. Context acquisition based on load sensing. In *International Conference on Ubiquitous Computing*, pages 333–350. Springer, 2002. Cited on page 40.

[99] Amit Sharma, Archan Misra, Vengateswaran Subramaniam, and Youngki Lee. Smrtfridge: Iot-based, user interaction-driven food item & quantity sensing. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*,

pages 245–257, 2019. Cited on pages 60 and 61.

[100] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019. Cited on page 20.

[101] Andrews Sobral et al. Bgslibrary: An opencv c++ background subtraction library. In *IX Workshop de Visao Computacional*, volume 27, 2013. Cited on page 24.

[102] Divya Srivastava, Rajitha Bakthula, and Suneeta Agarwal. Image classification using surf and bag of lbp features constructed by clustering with fixed centers. *Multimedia Tools and Applications*, 78(11):14129–14153, 2019. Cited on page 61.

[103] Ahmed Talib, Massudi Mahmuddin, Husniza Husni, and Loay E George. A weighted dominant color descriptor for content-based image retrieval. *Journal of Visual Communication and Image Representation*, 24(3):345–360, 2013. Cited on page 54.

[104] Zheng Tang, Renshu Gu, and Jenq-Neng Hwang. Joint multi-view people tracking and pose estimation for 3d scene reconstruction. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2018. Cited on page 101.

[105] TechCrunch. Inside amazon's surveillance-powered, no-checkout convenience store, 2019. Accessed: 2019-04-10. Cited on page 10.

[106] TechCrunch. Inside amazon's surveillance-powered, no-checkout convenience store, https://techcrunch.com/2018/01/21/inside-amazons-surveillance-powered-no-checkout-convenience-store/. Accessed: 2019-04-10. Cited on pages 35, 39, 64, 98, and 100.

[107] Ervin Teng, João Diogo Falcão, Rui Huang, and Bob Iannucci. Clickbait: click-based accelerated incremental training of convolutional neural networks. In *2018 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pages 1–12. IEEE, 2018. Cited on pages 61 and 99.

[108] Alessio Tonioni, Eugenio Serro, and Luigi Di Stefano. A deep learning pipeline for product recognition in store shelves. *arXiv preprint arXiv:1810.01733*, 2018. Cited on page 62.

[109] Toan Tran, Trung Pham, Gustavo Carneiro, Lyle Palmer, and Ian Reid. A bayesian data augmentation approach for learning deep models. *arXiv preprint arXiv:1710.10564*, 2017. Cited on page 30.

[110] Shashank Tripathi, Siddhartha Chandra, Amit Agrawal, Ambrish Tyagi, James M Rehg, and Visesh Chari. Learning to generate synthetic data via compositing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 461–470, 2019. Cited on pages 22, 25, and 30.

[111] Vipin Tyagi. Mpeg-7: Multimedia content description standard. In *Content-Based Image Retrieval*, pages 85–100. Springer, 2017. Cited on page 46.

[112] Md Nasir Uddin, Mm Rashid, Mg Mostafa, et al. Automated queue management system. *Global Journal of Management and Business Research*, 2016. Cited on page 9.

[113] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 109–117, 2017. Cited on page 31.

[114] Malvina Vega. 15+ retail statistics 2020 and the future of shopping, Accessed: 2019-11-15. Cited on page 63.

[115] Denis Vuckovac, Pascal Fritzen, Klaus Ludwig Fuchs, and Alexander Ilic. From shopping aids to fully autonomous mobile self-checkouts-a field study in retail. *Association for Information Systems*, 2017. Cited on page 65.

[116] Cheng Wang, Ming Cheng, Ferdous Sohel, Mohammed Bennamoun, and Jonathan Li. Normalnet: A voxel-based cnn for 3d object classification and retrieval. *Neurocomputing*, 323:139–147, 2019. Cited on pages 10, 61, and 99.

[117] Hao Wang, Qilong Wang, Fan Yang, Weiqi Zhang, and Wangmeng Zuo. Data augmentation for object detection via progressive and selective instance-switching. *arXiv preprint arXiv:1906.00358*, 2019. Cited on page 30.

[118] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. Cited on page 20.

[119] Wenyong Wang, Yongcheng Cui, Guangshun Li, Chuntao Jiang, and Song

Deng. A self-attention-based destruction and construction learning fine-grained image classification method for retail product recognition. *Neural Computing and Applications*, 32(18):14613–14622, 2020. Cited on pages 10 and 99.

[120] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016. Cited on page 43.

[121] Xiu-Shen Wei, Quan Cui, Lei Yang, Peng Wang, and Lingqiao Liu. Rpc: A large-scale retail product checkout dataset. *arXiv preprint arXiv:1901.07249*, 2019. Cited on pages 54 and 62.

[122] Cale G. Weissman. Why discount stores are one of the fastest growing retail sectors, https://www.modernretail.co/retailers/why-discount-stores-are-one-of-the-fastest-growing-retail-sectors/. Accessed: 2021-05-15. Cited on pages 98 and 100.

[123] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017. Cited on page 28.

[124] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 374–382, 2019. Cited on page 20.

[125] Jian Zhang, Yibo Lyu, Thaddeus Roppel, Justin Patton, and CP Senthilkumar. Mobile robot for retail inventory using rfid. In *2016 IEEE international conference on Industrial technology (ICIT)*, pages 101–106. IEEE, 2016. Cited on pages 10 and 100.

[126] Zippin. Zippin. checkout-free technology, https://www.getzippin.com/. Accessed: 2019-04-10. Cited on page 64.

[127] Barret Zoph, Ekin D Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V Le. Learning data augmentation strategies for object detection. In *European Conference on Computer Vision*, pages 566–583. Springer, 2020. Cited on page 30.