applied sciences

MDPI

Article

Building Unmanned Store Identification Systems Using YOLOv4 and Siamese Network

Shi-Jinn Horng [1,2,*] and Pin-Siang Huang [1]

[1] Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei 10607, Taiwan; m10815063@mail.ntust.edu.tw
[2] School of Computer Science, University of Technology Sydney, Sydney, NSW 2007, Australia
[*] Correspondence: horngsj@yahoo.com.tw; Tel.: +886-2-2737-6700

Abstract: Labor is the most expensive in retail stores. In order to increase the profit of retail stores, unmanned stores could be a solution for reducing labor cost. Deep learning is a good way for recognition, classification, and so on; in particular, it has high accuracy and can be implemented in real time. Based on deep learning, in this paper, we use multiple deep learning models to solve the problems often encountered in unmanned stores. Instead of using multiple different sensors, only five cameras are used as sensors to build a high-accuracy, low-cost unmanned store; for the full use of space, we then propose a method for calculating stacked goods, so that the space can be effectively used. For checkout, without a checking counter, we use a Siamese network combined with the deep learning model to directly identify products instantly purchased. As for protecting the store from theft, a new architecture was proposed, which can detect possible theft from any angle of the store and prevent unnecessary financial losses in unmanned stores. As all the customers' buying records are identified and recorded in the server, it can be used to identify the popularity of the product. In particular, it can reduce the stock of unpopular products and reduce inventory.

Keywords: unmanned store; anti-theft; low-cost; deep learning; Siamese network

check for updates

Citation: Horng, S.-J.; Huang, P.-S. Building Unmanned Store Identification Systems Using YOLOv4 and Siamese Network. *Appl. Sci.* **2022**, *12*, 3826. https://doi.org/10.3390/app12083826

Academic Editors: Rubén Usamentiaga and Byung-Gyu Kim

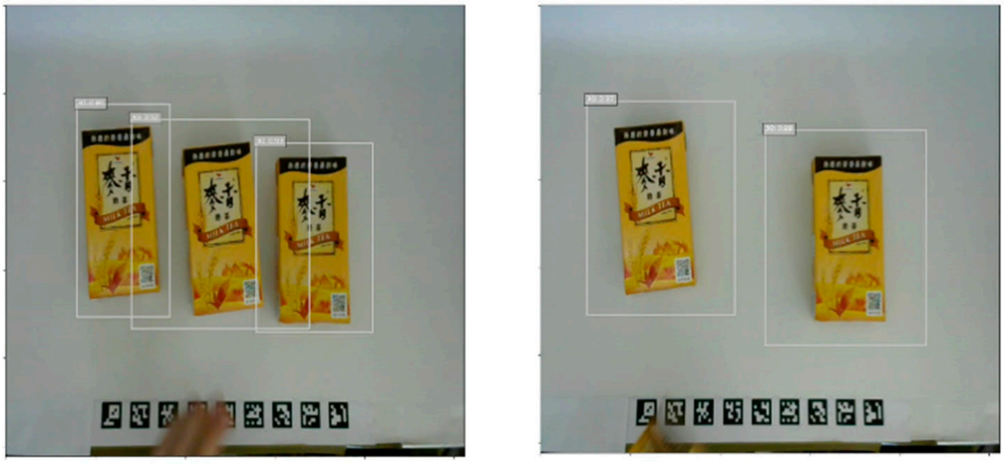
Received: 28 January 2022
Accepted: 7 April 2022
Published: 10 April 2022

1. Introduction

About two decades ago, it was hard to believe that a car can be self-driving; however, Tesla is in the mainstream of self-driving today. Compared to self-driving, the techniques used in unmanned stores are similar and will become popular in the near future.

As for saving the labor, there are many chain stores that are going to invest in unmanned stores, including FamilyMart and 7ELEVEN in Taiwan, Amazon in the United States, Alibaba Group in China, Bingo Box, Xingli, Guoxiaomei, F5 Future Store, etc. A shortage of funds is not serious for e-commerce stores but is fatal to small startups. How to balance the cost and the profit will currently be the biggest challenge for unmanned stores. In this paper, we focus on the premise of low cost and high recognition accuracy of products to solve the problems faced in unmanned stores. As for reducing the number of sensors used in the store, we only used five cameras to solve the problems of product identification, theft detection, and stacked product identification, and finally, a stable and reliable unmanned store system was successfully constructed.

In the unmanned store, in order to recognize the behavior of the customer, we define the "event" as identification. Figure 1 shows the images before the event and after the event. As can be observed, based on the images before the event and after the event, we can deduce how many products were taken out or put back by the customer through classification technique [1]. Who triggered the event? The shopping car list of the one who triggered the event would be modified. How to recognize the customer by face recognition technique is another key point in the unmanned store [2,3].



(**a**) (**b**)

Figure 1. Images associated to events. (**a**) Before event and (**b**) after event.

The contributions of this paper are summarized in the following:

1. We built the most light-weighted unmanned store identification systems for the unmanned store. We combined multiple models in sync to create this system, which covers the integration of multiple deep learning models and computer vision technologies. This will create a high-accuracy, low-cost unmanned store, and solve the problems that stores often encounter.
2. Most stores encountered problems such as having to retrain the deep learning model once a new product had arrived, and were not able to calculate the total amount accurately once the products were being stacked. The YOLO's model and the Siamese Network combined together can recognize new products without retraining the system.
3. Using two cameras, the system can calculate the total amount among the stacked products.
4. In terms of securing customer identification, we use the human posture estimation model to identify the shopping behavior of customers and to identify the customers. In addition, we have achieved the shopping scenario of taking the goods and leaving without a cashier.

All techniques used in this paper will be described in detail in the following sections.

2. Related Work

We all know that deep learning is a good technique in image classification, identification, and clustering, and all techniques developed previously could be used for the unmanned stores [1–3]. In order to enhance the correctness used in unmanned stores, some companies such as Amazon Go used a lot of micro sensors, sound detectors, an echo positioner, and a face recognizer, in order to try to reduce errors to be as small as possible. However, sensors are quite expensive and very harmful for the construction of unmanned stores. Instead of using a large number of sensors, in this paper, we built unmanned stores with low cost and high accuracy.

A number of ways for object detection have been proposed [4]. Tehsin et al. [5] applied a particle swarm optimization of correlation filters for object recognition. Yousaf et al. [6] applied a filter and morphological operation to enhance the edges of the image for object detection. Awan et al. [7] proposed a difference of the Gaussian (DoG)-based logarithmically preprocessed EEMACH filter to manage multiple distortions in a single training instance and enhanced the recognition performance. Zhou et al. [8] developed a Scale-Transferrable Detection Network (STDN) for detecting multi-scale objects in images. There are two major ways to recognize objects in deep learning models; one is the one-stage method and the other is the two-stage method. The former performs the positioning and classification directly, and the latter first extracts the candidate region of interest and then does the classification. Usually, the former runs faster than the latter but the latter has better accuracy than the former. The YOLO family, including YOLO [9], YOLO v2 [10], YOLO v3 [11], YOLO v4 [12], and SSD (Single Shot MultiBox Detection) [13], are the representatives of the former, and the representatives of the latter are RCNN [14], Fast RCNN [15], and Faster RCNN [16], respectively.

Traditionally, the deep learning neural models require a large number of samples to conduct learning during the training process; in practice, the lack of samples for training usually leads to poor performance. Koch [17] proposed the Siamese network to conduct a better similarity checking for objects with a small number of samples. We can then use the Siamese network in unmanned stores for classification.

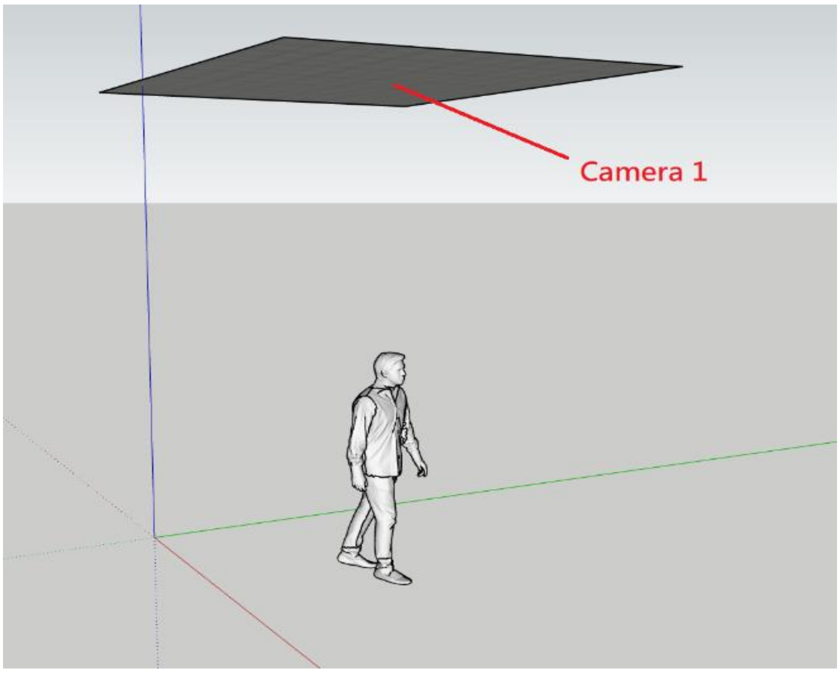
To recognize customer behavior more accurately, a posture prediction model developed by Cao et al. using OpenPose [18] is used to assist in identifying the physical behavior of customers. It first captures the 2D image and then the key points of the body are detected and help the body tracking algorithm to identify each pose at different angles. These poses are useful in identifying the behavior of a customer.

3. Preliminaries

We describe all the techniques to be used in our scheme in this section.

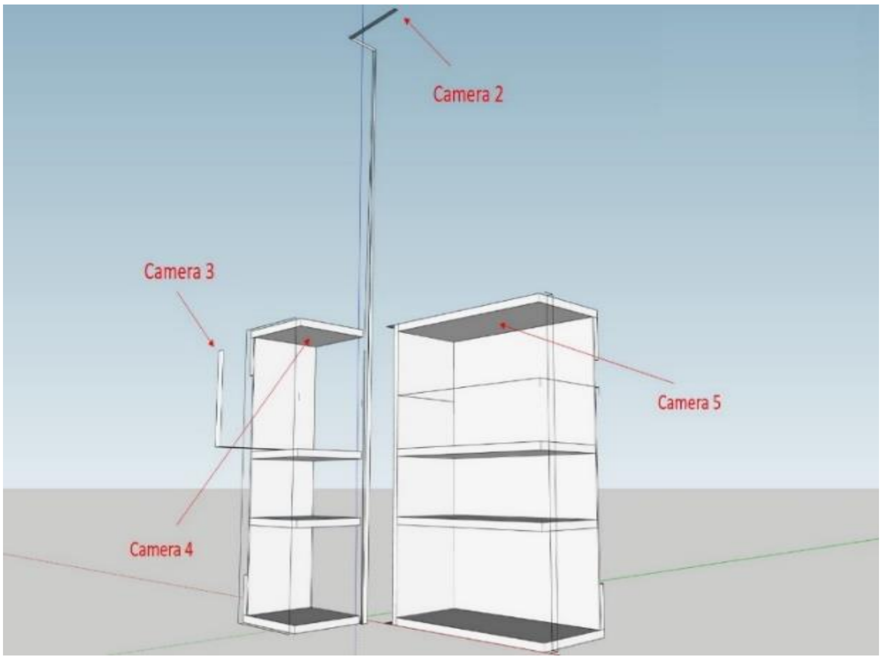
3.1. Equipment

For the hardware equipment, we only used five Logitech C270 webcams and one GTX 1660 Windows computer. One camera is used for the entrance/exit of the store, as shown in Figure 2a. The rest of the four cameras are located inside the store. Camera two is installed on the top. Cameras three and four are responsible for handling the stacked products, and camera five is used to identify products on the platform, as shown in Figure 2b.



(**a**)

Figure 2. *Cont.*



(b)

Figure 2. Hardware architecture: (**a**) Camera 1 is located in the entrance/exit; (**b**) the locations of the rest of the four cameras are inside the store.

3.2. Marker

An ArUco marker [19] is a synthetic square marker that is composed of a wide black border and an inner binary matrix determining its identifier (id). The marker size determines the size of the internal matrix. For instance, a marker size of 4 × 4 is composed by 16 bits.

An easy way to detect whether a hand is entering the merchandise cabinet or not is using the infrared sensor. When there is an object blocking the infrared sensor, a disturbance occurs and then it is recognized as an event and an image is immediately taken. On the contrary, when the infrared sensor is back to its normal condition, it indicates that the hand had left the shelf. It means that an event was over and another image is taken again. Instead of using infrared sensors, in order to reduce the cost, we use a marker to detect the movement event and put markers all over the entrance of the merchandise cabinet. The marker will be obscured while there is a hand passing it and will identify whether someone is going to take items. When the marker appears again, it means that some items were taken away. Figure 3 shows the detail.

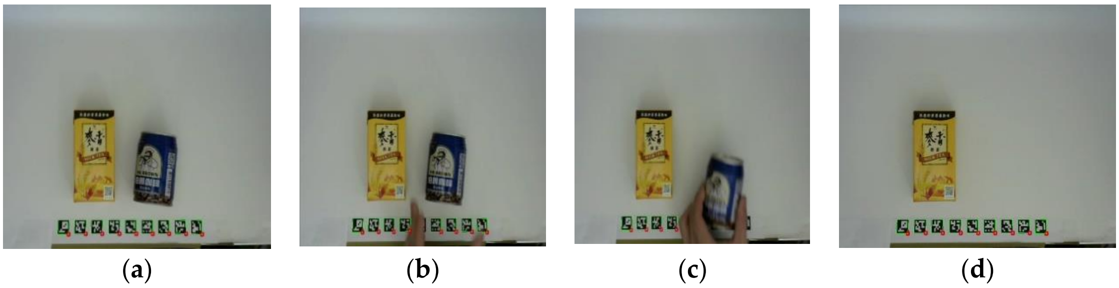


Figure 3. Marker function. (**a**) Normal condition; (**b**) Hand passing; (**c**) An item is moving; (**d**) Item taken.

3.3. Object Detection

In deep learning, as we mentioned previously, object detections are classified into a one-stage method and two-stage method, respectively. For the sake of real time, instead of using the two-stage method, we used the one-stage method. We used YOLOv4 developed by Alexey Bochkovskiy [12] for object detection; not only it is faster but its accuracy can reach a state-of-the-art level. The loss function used in YOLOv4 can be observed in [12].

3.4. Siamese Network

Siamese network is useful for object identification. The single-sample learning method is to give the Siamese network two images at the same time and let it compare whether the two input images belong to the same category or not. It then can be used to select which category is the closest to the current input object.

The main steps of single-sample learning are described in Figure 4. First, two images, X1 and X2, are fed into the twin network f(x), where it shares the same weights with the same architecture. These two images are then going into two of the same networks, respectively; a 4096-dimensional output vector is obtained for each network. Then, the difference, f(x1) − f(x2), is fed into the fully connected layers for identification. We used the Sigmoid function for identification; if the output is closer to zero, it means that both are more distinctly different; otherwise, both are quite similar.

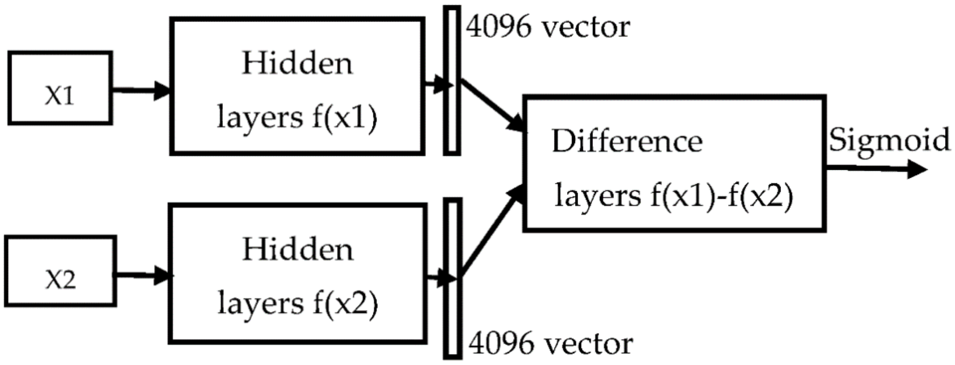


Figure 4. Siamese network model.

The loss function used in one-shot learning is Triplet Loss. Let $f(x)$ transform x into a high-dimensional space to form a vector. $f(x^a)$, $f(x^p)$, and $f(x^n)$ are the vectors converted from the anchor, negative sample, and positive sample, respectively.

The distance between the negative sample and anchor should be less than the distance between the positive sample and anchor with a self-determined constant α, where $||f(X^a) - f(X^p)||_2^2 + \alpha > ||f(X^a) - f(X^n)||_2^2$.

The loss function is then defined in Equation (1).

$$Loss = ||f(X^a) - f(X^p)||_2^2 - ||f(X^a) - f(X^n)||_2^2 + \alpha \tag{1}$$

3.5. Person Re-Identification

A pedestrian re-identification is called a person re-identification (ReID), or PREID for short. The technology that uses computer vision to determine whether there are specific pedestrians in images or movies is an image retrieval problem. A pedestrian image can be used to search in cross camera situations. Usually, the collected pedestrian images used for searching are stored in the Gallery database, and each image is associated with an ID and named g_{id}. We named each pedestrian identity as p_i in the query. During the recognition process of ReID, the pedestrian identity p_i will be compared to each identity stored in the Gallery database for similarity, $sim(p_i,\ g_{id})$. If the similarity is lower than the threshold, it means that this person is likely to appear for the first time, and then a new ID will be given to this person and added to the Gallery database; otherwise, the pedestrian g_{id}, whose similarity is the maximum, will be considered the output of the identity of the pedestrian by Equation (2):

$$\underset{g_{id}}{Largmax}\ (\ sim(p_i, g_{id}))\ , p_i \in \text{ Query} \tag{2}$$

There are many versions of pedestrian re-identification. Currently, MGN [20] and SCR [21] are widely used. Many models are derived from these two architectures. Instead of using MGN [20] and SCR [21], the architecture used in this paper is that which is developed in [22] and this architecture is the current state-of-the-art model in the PREID field.

3.6. Open Pose

In order to prevent the unmanned store from theft, we used the OpenPose [18] human posture estimation model to judge customer behavior. Instead of using the dual-lens body detection camera (such as Kinect) and infrared sensors, OpenPose only needs the RGB image as the input; it can detect key points, including elbows, wrists, knees, and other body positions. In particular, it can reach a good recognition rate from a bird's-eye view.

Figure 5 shows using OpenPose to determine purchase status. How to use OpenPose to determine the purchase status of customers in the unmanned store, prevent the store from theft, and improve the accuracy of the staggered pickup will be discussed in detail in the next section.

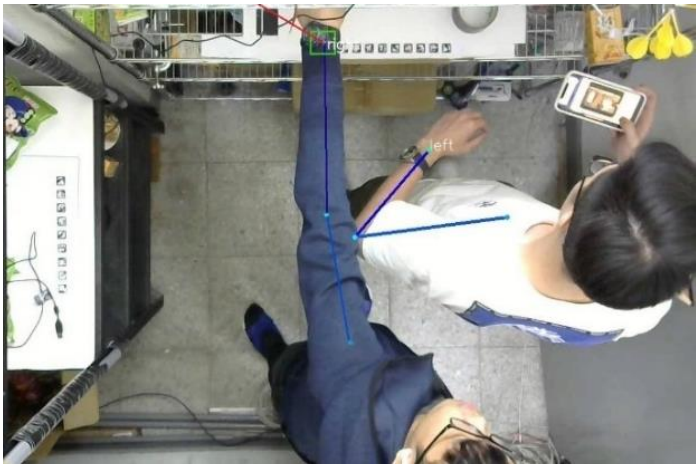


Figure 5. Using OpenPose to determine the purchase status of customers.

4. Proposed Scheme

The shopping process is stated in Algorithm 1 and it consists of five major steps. In step one, while the customer enters into the unmanned store, a photo is taken, and the customer's information is created and stored in the Gallery database using the YOLO v4 person detection model. In step two, the customer is shopping. When the customer puts his hand in the merchandise cabinet, the hand will be detected by the marker, and after the hand leaves the cabinet, the number of items taken or returned will be calculated by the YOLO item detection model. In step three, the customer may take some items that just arrived at the store, which have not yet been trained in the YOLO item detection model—these items can be identified by the Siamese Network. In step four, the customer who took the items will be identified by the OpenPose and PREID Network. In step five, step two to step four are repeated until the customer leaves the store and finishes the checkout. Instead of using the time complexity to demonstrate the performance, a practical execution time of each model taken in the unmanned store system is shown in Table 1; it takes only 372 ms for the whole system.

Algorithm 1: Shopping Process Algorithm

1. Use YOLO v4 to identify the customer and put the customer's information in the Gallery database while he/she enters the unmanned store.
2. When the marker detects a hand entering the cabinet, a shopping event is started, and photos are taken during this period. When the marker detects a hand leaving the cabinet, the number of items are calculated by the YOLO item detection model as the shopping event ends.
3. Use the Siamese Network to identify the items which were not trained in YOLO item detection model.
4. Use OpenPose and the PREID Network to identify which customer took the items.
5. Repeat step 2 to step 3 until the customer leaves the store and finishes checkout.

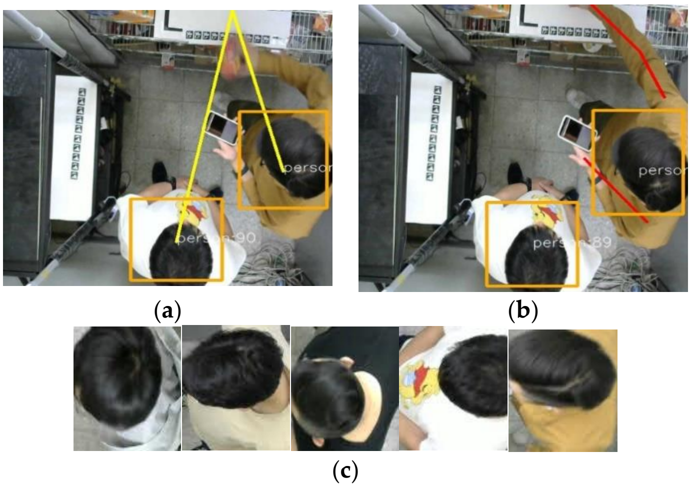
Table 1. The execution time of each model.

Algorithm	YOLO v4	Openpose	PREID Network	Siamese Network	Marker Detection
Speed (ms)	45	77	100	120	30

4.1. Analysis of Unmanned Store Events

When an event is triggered, we will receive three images and the location of the merchandise cabinet. The three images are: the store's top view image, the product image when the event is triggered as shown in Figure 1a, and the product image after the event is triggered, as shown in Figure 1b. Through this information, we can judge which customer took which item in the store.

We use the images taken before and after, put them into the YOLO v4 model, and the model will respectively calculate the categories and quantities of these two images; the difference in quantity between the two images will be the quantity purchased (or put back) into the merchandise cabinet by the customer. From the top view image obtained directly above the store, the identity of the customer can be judged. First, we use the position of the trigger commodity cabinet in the image to find the closest customer frame as the event trigger through the shortest Euclidean distance or we use the OpenPose method, as shown in Figure 6a,b, respectively. After obtaining the image of the trigger, it is compared to the images of all customers in the store one by one by using the Lightweight PRIED network to find out which customer triggered the event, and the purchased goods are placed into the shopping cart list corresponding to the customer found, as shown in Figure 6c.

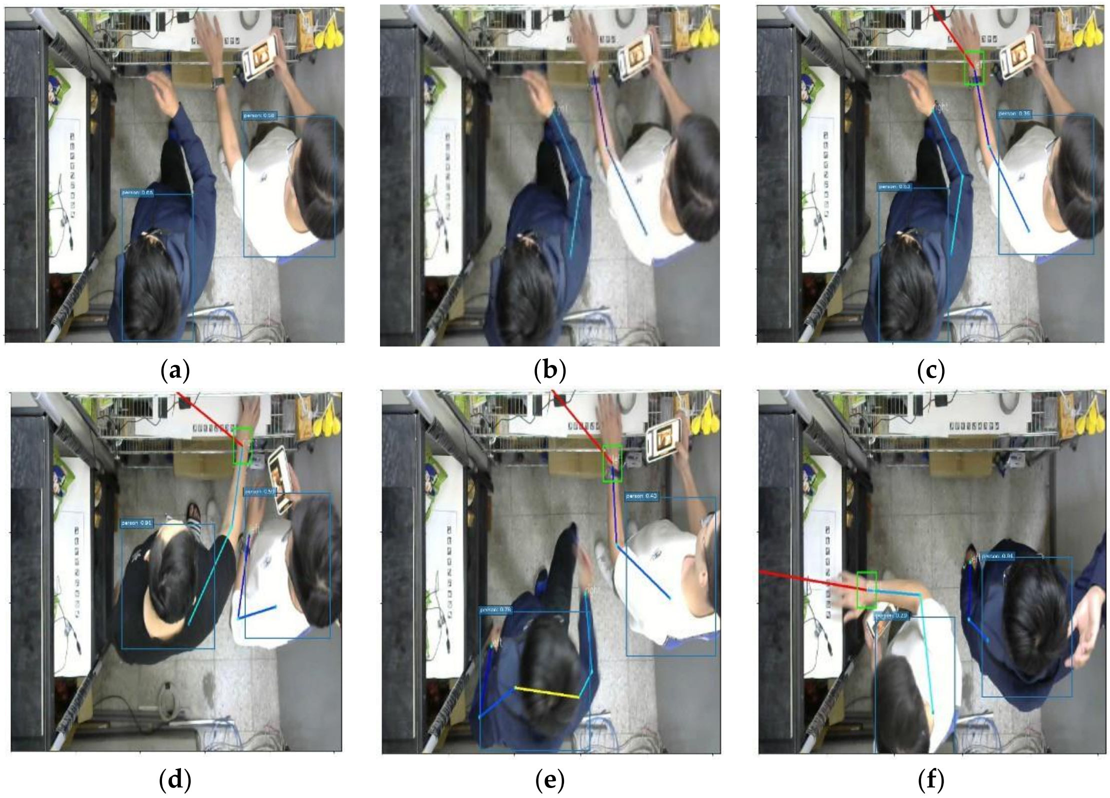


(a) (b) (c)

Figure 6. (**a**) Locating the event trigger by Euclidean distance; (**b**) locating the event trigger by OpenPose; (**c**) find the customer ID in the store.

4.2. Customer Identification and Anti-Theft System Based on OpenPose, YOLOv4, and PREID Network

In order to correctly judge the identity of the customer, we used three models; the first model obtains the bounding box of people, which can return the customer location and the size of the bounding box, as shown in Figure 7a; the second model is the OpenPose model, which uses the same input image as the first model, and the output is the key positions of each customer in the store (such as wrists, shoulders, knees, etc.). As the store only needs to recognize the arm as a condition for judging the customer's image, it only needs to recognize six key points of the customer, the left wrist, left elbow, left shoulder, right wrist, right elbow, and right shoulder. For example, if there are currently two customers in the store, there will be four arms and 12 key points. In addition, the model will only detect one of the arms because the customer is partially obscured, such as the person on the left in Figure 7b.


(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 7. (**a**) Person detection; (**b**) OpenPose; (**c**) find the customer; (**d**) triggered behind; (**e**) triggered front; (**f**) no trigger.

Next, we use the known position of the merchandise cabinet and the key points returned by OpenPose to find the key point of the wrist with the closest Euclidean distance; we use this wrist to find the key point of the person's shoulder, and then use the position of the shoulder and the customer's position in the image of the person detection model. The shortest Euclidean distance for the image position obtains the customer image that triggers the event, as shown in Figure 7c.

OpenPose can identify many different situations in the store. For example, customers who are far away from the merchandise shelf reach out for items in the store. Through the links of key points, we can also find out the customers who triggered the event behind, as shown in Figure 7d. Here are some normal shopping situations, as shown in Figure 7e,f.

After we have the image of the person who triggered the event, we use the image as the input to Person REID, compare the similarity with all customer images in the store, and select the person with the highest similarity as the trigger of this event.

4.3. New Item Identification System Based on YOLO v4 and Siamese Network

In order to facilitate the explanation of the system architecture, we call products that have not appeared in the training set as new products, and products that have been trained as old products.

To be able to successfully identify whether a product is a new product, first we will use the second model to identify the existing 30 product categories, as shown in Figure 8a. The second model is used to perform object detection; that is, it only detects the location and size of the object but does not classify the detected objects. Figure 8b shows the object detection. Next, the intersection of all recognized frames of these two models is taken so that the frames recognized by the models will be put in one image, as shown in Figure 8c.

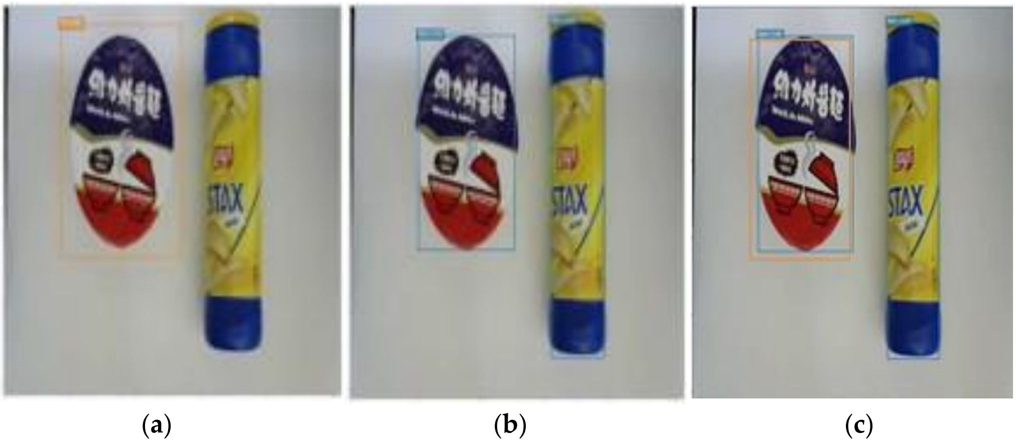


Figure 8. (**a**) Classification model; (**b**) object detection; (**c**) IoU of two models.

The two models predict with the same image; there will be many overlapping boxes, which are calculated as IoU (Intersection over Union). When IoU is greater than 0.7, the candidate box is removed and regarded as an old product; otherwise, it is regarded as a new product. Using this structure, as shown in Figure 9, we can classify products. In Figure 8c, the instant noodle on the left is judged as an old product, and the potato chip on the right is judged as a new product.

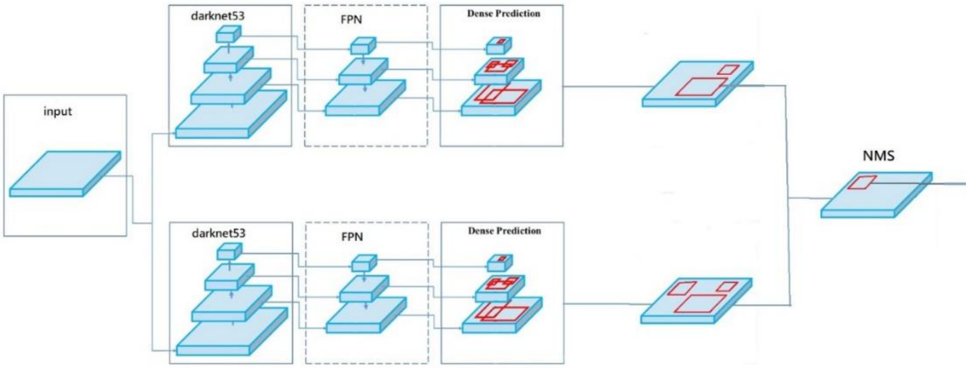


Figure 9. Classification model and object detection mode.

Suppose there is a new product; the product image is resized to 106*106, it is used as an input to the Siamese network, and the other input of the Siamese network is from the image of all new product categories. The two images will eventually generate two vectors. We can then calculate the similarity of these two generated vectors and select the highest one from new product categories as the category of the product. In this way, we can identify new products without retraining the model.

4.4. Product Identification System Based on YOLO v4 Stacking and Height Occlusion

For the deep learning model, the target object must appear completely in the image before the product can be identified. As long as the product is slightly obscured, the product may be identified incorrectly. Simple object identification cannot solve the problem of stacked products due to occlusion. We designed a set of methods that can solve the problems of stacked goods and height occlusion, which can effectively improve the utilization of space and can also greatly reduce the construction cost of unmanned stores.

First of all, we designed an identification model for stacking products and height occlusion; this model has six goods categories, but is extended into eight categories because of two added categories, "Top of Bottle" and "Top of Can", which are used to solve the occlusion problem mentioned later, such as bottled Coca-Cola. Aluminum can tops, Pringles tops, bottled beverage tops, etc. are all products of this type. In this way, once the product itself is blocked, the number of top covers can still be used to calculate the number of products.

There are three calculation methods for commodity appearance, as shown in Table 2. The first calculation method can be directly identified through the YOLO model. As the training sample contains many occlusions and stacking situations, after a large amount of data enhancement and training, a model that can identify more than half of the occluded products is formed, as shown in Figure 10a.

Table 2. Identification methods used for height occlusion and stacking.

Method	Commodity Category
Directly identify the product and calculate the quantity	Noodles, packaged snacks, and boxed snacks, as shown in Figure 10a
Calculate the top	Bottled drinks and Pringles, as shown in Figure 10b
Dual camera calculation	Iron, aluminum cans, and aluminum foil packages, as shown in Figure 10c

The second calculation method is because some products have long and narrow shapes, only the top can be identified when they are densely arranged. Therefore, the calculation method is to calculate the number of recognized top caps, such as bottled beverages and canned snacks. In this way, there are several top covers which are equivalent to the quantity of several commodities, as shown in Figure 10b.

The third calculation method requires two cameras to identify stacked products, which are iron cans and aluminum foil packages, as shown in Figure 10c. First, the top of the product is regarded as a category, referred to as "top covers", using two cameras. From top views, the number of products is calculated from the top angle, but it is impossible to determine whether the products are stacked. The side view image can be used to judge whether the product is on the first layer or stacked on the second layer. The number of "top covers" is discovered by looking down on the top-view camera, and then using the side camera to adjust the camera angle to find the stacked products.

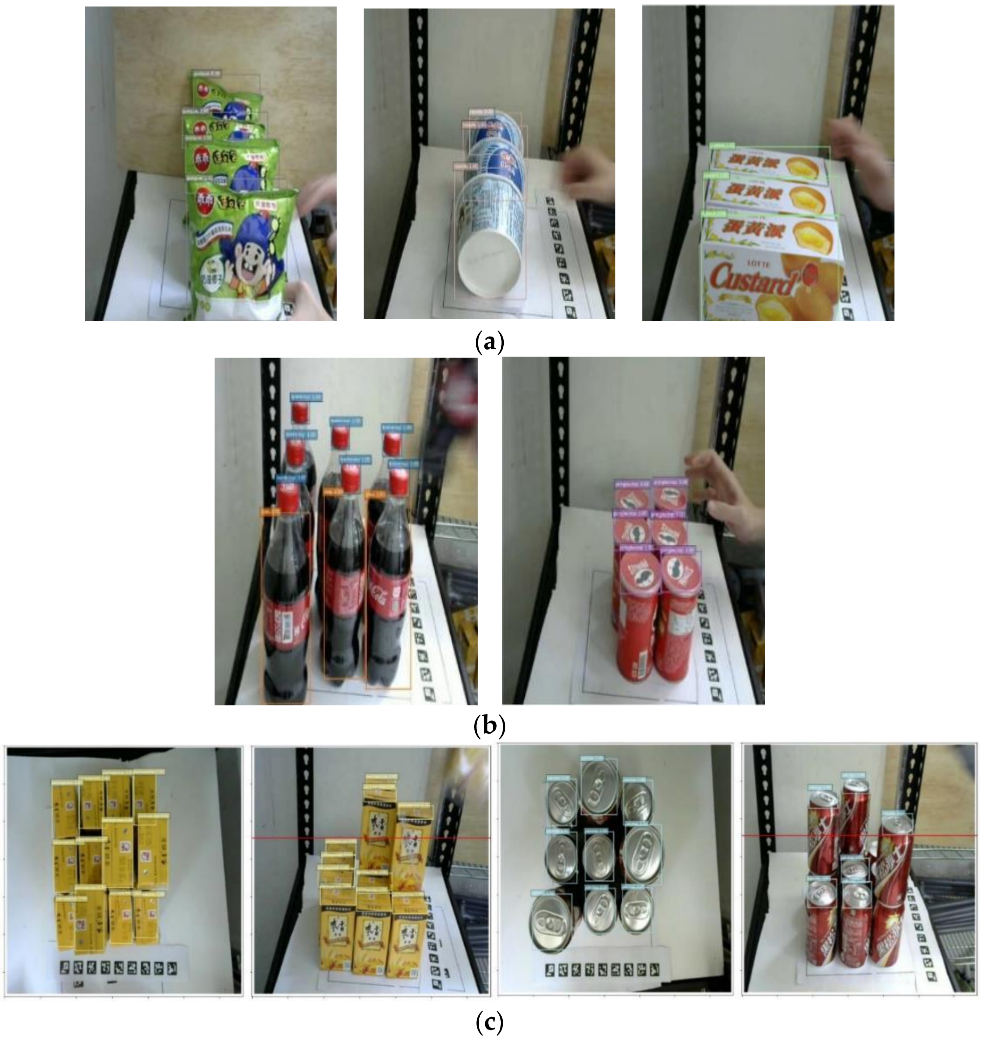
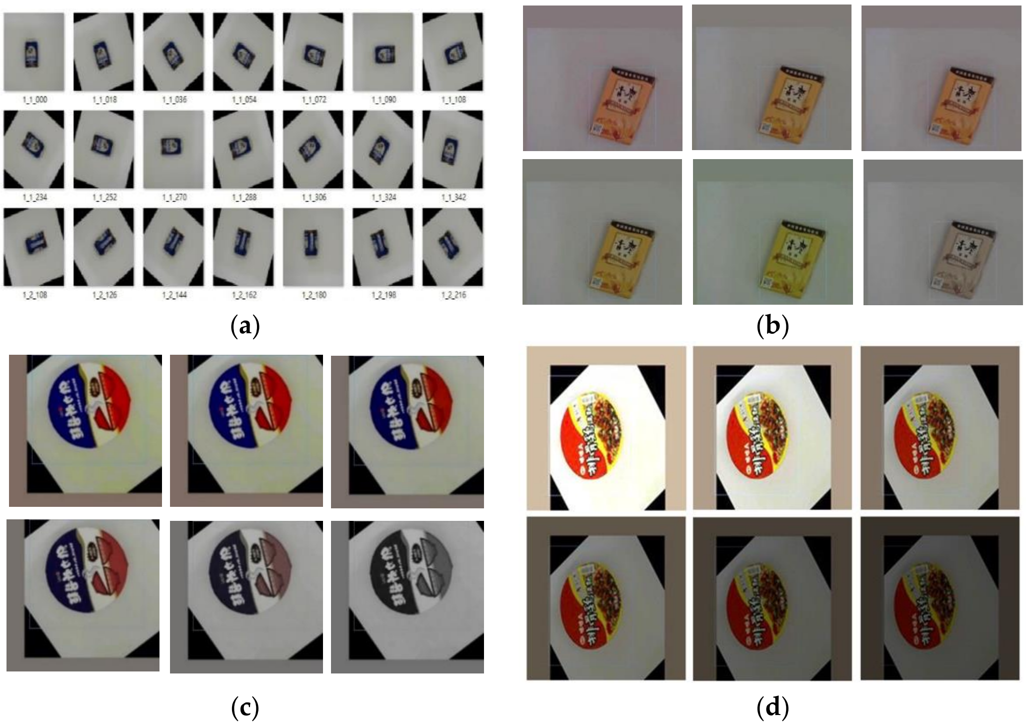


Figure 10. Identifying products in different categories. (**a**) Packed products; (**b**) top view for dense products; (**c**) top and side views using two cameras.

5. Results

According to human beings, when an object undergoes a certain deformation, it is still recognizable by human eyes and brain perception; for machine learning, it may not be that easy. Therefore, data augmentation is an important thing in order to improve the performance and robustness of the model. The following are the data augmentation methods used in this paper. Rotation is used to generate a new image from the original training sample every 15 degrees, thus one photo will generate 24 additional rotated images, as shown in Figure 11a. Images of different hues are randomly generated, because products often appear similar in characteristics but different colors. For example, the appearance characteristics of Maixiang milk tea and Maixiang black tea are almost the same except for the color hue. If the color range is adjusted to be too large, it may be mistakenly judged as the same product. To solve this problem, we should adjust the hue to plus or minus 2.5%, and images with different hues would be generated from the products.

Therefore, the hue is adjusted to plus or minus 2.5%; images with different hues will be randomly generated from the original product image, as shown in Figure 11b. The saturation is set to plus or minus 10%, and images with different saturations are randomly generated in this interval, as shown in Figure 11c. The exposures are set to plus or minus 10%, and by randomly generating the exposures, the product can be more easily identified when the light is reflected, as shown in Figure 11d. The image sizes are randomly enlarged or reduced by 30%, so that the product can be identified when it is far or near.



(**a**) (**b**)

(**c**) (**d**)

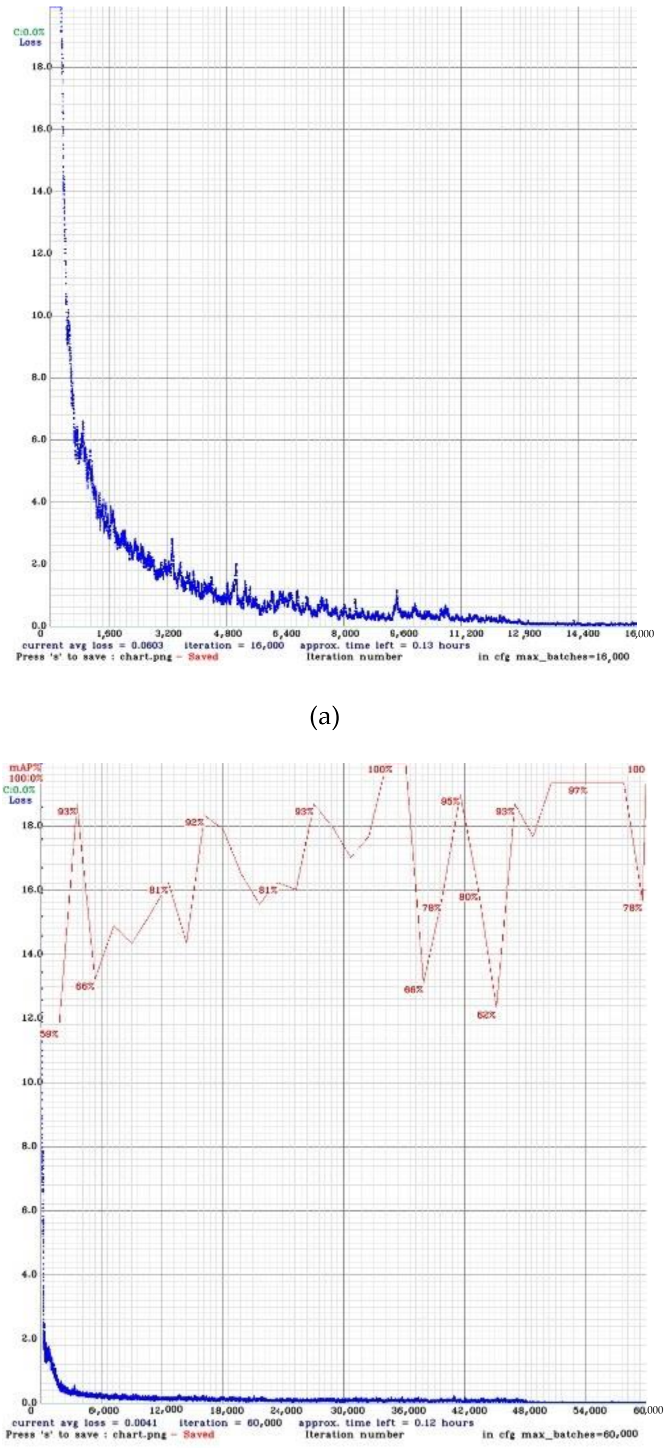
Figure 11. Data augmentation. (**a**) Rotation; (**b**) hue; (**c**) saturation; (**d**) exposure.

The training process in YOLO v4 is shown in Figure 12. Figure 12a is for the person detection model, Figure 12b is for the 30 items detection model, and Figure 12c is for the stacking and height occlusion items detection model. The number of iterations is depicted in the horizontal axis, the vertical axis is the training loss, and every 1000 iterations will be calculated once with the value of mAP. During the training, we set the learning rate to 0.0001, and data augmentation and the dropout layer [23] are used to prevent over-fitting. There are several tests for each of the three YOLO v4 models in the store. The results are shown in Table 3:

Table 3. The results of YOLO v4 model.

YOLO v4 Model	Testing Data Description	Testing Set	mAP
Person detection	Customer's overhead image	50 images	1
30 items detection	30 non-stacking products	60 images	0.96
Stacking and height occlusion items detection	6 products	50 images	0.98

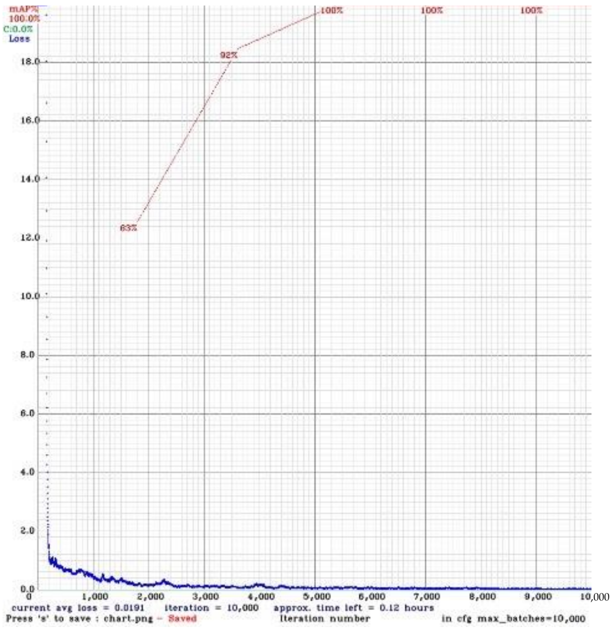
To train the PREID model, we used 2066 different images acquired from 27 people in total. The marked data are manually processed. The same image of the same person is placed in the same folder, and any two data are randomly selected, and the similarity is compared and calculated. Triplet loss [24] and cross entropy (CE) loss [25] are the two loss functions used. We set the learning rate to 0.0006 and use Adam [26] as the optimizer, where the batch size is set to 16, and the training result is shown in Figure 13a.



(a)

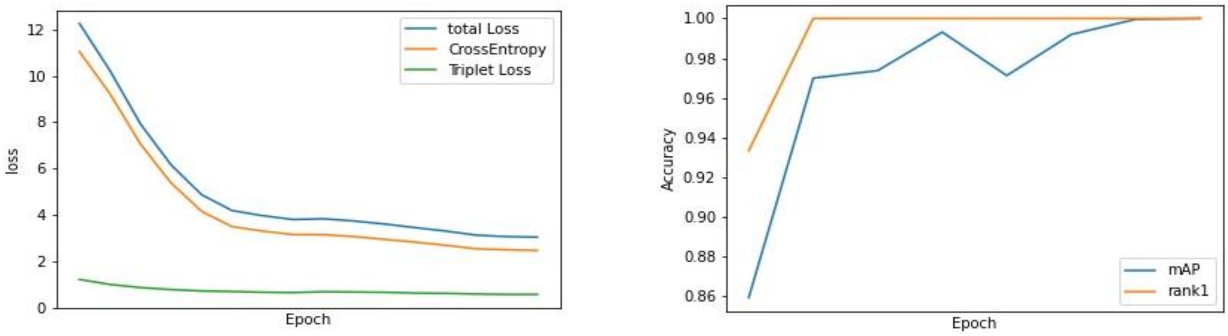
(b)

Figure 12. *Cont*.



(c)

Figure 12. Training models. (**a**) person detection model (**b**) 30 items detection model (**c**) stacking and height occlusion detection model.



(**a**)

(**b**)

Figure 13. PREID (**a**) training loss (**b**) validation.

In PREID, we often use the first hit (rank-1) to measure. If the query corresponding to the gallery has the highest similarity, the query meets the first hit, and we can also use rank-n as the measurement standard. Figure 13b shows the rank-1 and mAP performance of the validation set during the training process. For the testing set, we use the maximum storage capacity of six people as the test target and test 36 images. The rank-1 is 0.9722. Table 4 shows the result.

Table 4. Pedestrian re-identification model accuracy.

Model Name	Testing Data Description	Rank-1
Lightweight PREID	There are 36 pedestrian images in Query, which should be classified into 6 different pedestrian categories	0.9722

We also test the stability of the entire system based on actual shopping. We asked customers to enter the store to simulate the actual picking/returning to the shelf. With one shopping event as a unit, we tested 600 events. An event is correct when one puts the correct product category and quantity into one's shopping cart with one's correct customer ID. We divide incorrect cases into two kinds, object-detection error and person-identification error, and the total accuracy is 94.5%. The results are shown in Table 5:

Table 5. Test of the entire system.

Object-Detection Error	Person-Identification Error	Correct Cases	Testing Set (Accumulated)	Accuracy
3	1	96	100 events	0.96
8	5	187	200 events	0.935
11	6	283	300 events	0.943
13	8	379	400 events	0.947
18	11	471	500 events	0.942
20	13	567	600 events	0.945

In order to prove that adding the OpenPose anti-theft system does make the store's theft rate lower, it is compared to the method using the minimum Euclidean distance metric using the same 100 image test, including interlaced shopping, stolen goods, normal shopping, etc. In this case, adding OpenPose can cope with more complicated situations and improve accuracy. The results are shown in Table 6.

Table 6. Accuracy rates using Euclidean distance and OpenPose anti-theft system.

Method	Testing Data Description	Testing Set	Accuracy
Minimum Euclidean distance	Image directly above the unmanned store field	100 images	0.86
Anti-theft system architecture			0.98

Compared to Amazon Go [27], our system uses less hardware cost, but with a compatible accuracy. All unmanned stores have a camera sensor; in addition, Amazon Go has other sensors, including infrared sensors, volume displacement sensors, light curtains, and weight sensors. 7ELEVEN and FamilyMart use less sensors then Amazon GO. With less equipment, we can achieve a scenario of cost reduction. Indeed, if other sensors, such as the weight sensor, are added to the system in the future, the accuracy can be further improved. Table 7 shows the comparison.

Table 7. Cost analysis of this paper and other unmanned stores.

Unmanned Store	Camera	Infrared Sensors	Volume Displacement Sensors	Light Curtains	Weight Sensors	RFID
Amazon GO	v	v	v	v	v	
7ELEVEN	v	v				v
FamilyMart	v	v			v	
OURS	v					

6. Conclusions

This paper proposes an unmanned store system, combined with the architecture of multiple deep learning models with limited resources, to solve the problems that existing in unmanned stores. The combination of YOLO v4 and the Siamese network is the core of this paper. The whole system can recognize customers, products, and quantities correctly during shopping and achieve the goal of "take it and go".

Compared to Amazon Go [27], our system uses less hardware cost but with a compatible accuracy. Indeed, if other sensors, such as the weight sensor, are added to the system, the accuracy can be further improved. All customers' buying records are recorded and the popularity of a product can be instantly mined and shown. Some unpopular products can be used for promotion. Hence, the system can help reduce the stock of unpopular products and reduce inventory.

Author Contributions: S.-J.H. was the supervisor administrating the project; he reviewed and edited the manuscript. P.-S.H. carried out the investigation, analysis, and validation and wrote the original draft. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Ministry of Science and Technology under contract number 110-2221-E-011-125-, 110-2218- E-011 -006 –MBK, 111-2218-E-011-011-MBK and by the "Center for Cyber-physical System Innovation" from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Duan, M.; Li, K.; Liao, X.; Li, K. A Parallel Multi classification Algorithm for Big Data Using an Extreme Learning Machine. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 2337–2351. [CrossRef]
2. Duan, M.; Li, K.; Li, K.; Tian, Q. A Novel Multi-task Tensor Correlation Neural Network for Facial Attribute Prediction. *ACM Trans. Intell. Syst. Technol.* **2021**, *12*, 1–22. [CrossRef]
3. Duan, M.; Li, K.; Ouyang, A.; Win, K.N.; Li, K.; Tian, Q. EGroupNet: A Feature-enhanced Network for Age Estimation with Novel Age Group Schemes. *ACM Trans. Multim. Comput. Commun. Appl.* **2020**, *16*, 1–23. [CrossRef]
4. Zou, Z.; Shi, Z.; Guo, Y.; Ye, J. Object detection in 20 years: A survey. *arXiv* **2019**, arXiv:1905.05055.
5. Tehsin, S.; Rehman, S.; Bin Saeed, M.O.; Riaz, F.; Hassan, A.; Abbas, M.; Young, R.; Alam, M.S. Self-Organizing Hierarchical Particle Swarm Optimization of Correlation Filters for Object Recognition. *IEEE Access* **2017**, *5*, 24495–24502. [CrossRef]
6. Yousaf, M.R.; Rehman, S.; Dawood, H.; Guo, P.; Mehmood, Z.; Azam, S. Saliency based object detection and enhancements in static images. In *Lecture Notes in Electrical Engineering, Proceedings of the International Conference on Information Science and Applications, Macau, China, 27 March 2017*; Springer: Singapore, 2017.
7. Awan, A.B.; Rehman, S.; Bakhshi, A.D. Composite filtering strategy for improving distortion invariance in object recognition. *IET Image Process.* **2018**, *12*, 1499–1509. [CrossRef]
8. Zhou, P.; Ni, B.; Geng, C.; Hu, J.; Xu, Y. Scale-transferrable object detection. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 528–537.
9. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
10. Redmon, J.; Farhadi, A. Yolo9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
11. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
12. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.Y.-M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
13. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
14. Kido, S.; Hirano, Y.; Hashimoto, N. Detection and classification of lung abnormalities by use of convolutional neural network (cnn) and regions with cnn features (r cnn). In Proceedings of the International IEEE Workshop on Advanced Image Technology (IWAIT), Chiang Mai, Thailand, 7–9 January 2018; pp. 1–4.
15. Girshick, R. Fast r cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
16. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r cnn: Towards real time object detection with region proposal networks. *arXiv* **2015**, arXiv:1506.01497.
17. Koch, G.; Zemel, R.; Salakhutdinov, R. Siamese neural networks for one shot image recognition. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 7–9 July 2015; Volume 37.
18. Cao, Z.; Hidalgo, G.; Simon, T.; Wei, S.E.; Sheikh, Y. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 172–186. [CrossRef] [PubMed]
19. Detection of ArUco Markers. Available online: https://docs.opencv.org/3.4/d5/dae/tutorial_aruco_detection.html (accessed on 28 April 2021).

20. Wang, G.; Yuan, Y.; Chen, X.; Li, J.; Zhou, X. Learning discriminative features with multiple granularities for person re-identification. In Proceedings of the 26th ACM International Conference on Multimedia, Seoul, Korea, 22–26 October 2018; pp. 274–282. [CrossRef]
21. Chen, H.; Lagadec, B.; Bremond, F. Learning discriminative and generalizable representations by spatial channel partition for person re-identification. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 2483–2492.
22. Herzog, F.; Ji, X.; Teepe, T.; Hormann, S.; Gilg, J.; Rigoll, G. Lightweight Multi-Branch Network for Person Re-Identification. *arXiv* **2021**, arXiv:2101.10774.
23. Quispe, R.; Pedrini, H. Top-DB-Net: Top DropBlock for Activation Enhancement in Person Re-Identification. *arXiv* **2020**, arXiv:2010.05435.
24. Dong, X.; Shen, J. Triplet loss in siamese network for object tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
25. Zhang, Z. Improved adam optimizer for deep neural networks. In Proceedings of the IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), Banff, AB, Canada, 4–6 June 2018.
26. Wang, X.; Han, X.; Huang, W.; Dong, D.; Scott, M.R. Multi-similarity loss with general pair weighting for deep metric learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 5022–5030.
27. Kirti, W.; Wukkadada, B.; Nadar, V. Just walk-out technology and its challenges: A case of Amazon Go. In Proceedings of the 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 11–12 July 2018.