

Intelligent Checkout System

Jiaming Li, and James Bozhkov

Abstract—The traditional checkout systems are mainly based on scanning the labels attached on the items which require manual labor, and thus inefficient. Nowadays, as deep learning reveals excellent performance towards object detection tasks, this technique has been integrated in assisting the speed of each checkout process. However, most research merely focuses on detecting merchandise, yet overlooked the connection with customers. In this research paper, we are proposing a structured intelligent checkout system that obtains the dexterity of checking a customer's identity and their items embedded by dual cameras. The first camera is installed in front of the customer to determine their identity. The second camera is installed overhead for detecting multiple merchandises according to their appearances by using the state-of-the-art object detection model Mask R-CNN trained on D2S supermarket products dataset. In the proposed system, face detection technique is used to allocate users' faces in the image gathered from the camera, and the facial features encoding techniques are utilized in determining the customers' individual identity. Besides, the deep learning methods are implemented for the usage of both identity anti-spoofing checking and product detection. In the end, our research has found that the combination of face recognition and the anti-spoofing model can be used to analyze people's identity. Besides, in the experiment, the Mask R-CNN model yields a 98.15% mAP when classifying items from 13 categories. The high accuracy and robustness produced from the object detection model makes the checkout system integration seamless. The contribution of our work is providing a fully automatic checkout system architecture that utilizes the contemporary deep learning and machine vision techniques for accelerating the checkout process and reducing manual labor.

Index Terms—Checkout System, Object Detection, Facial Recognition, Deep Learning, Mask R-CNN

I. INTRODUCTION

Self-checkout systems are meant to make each checkout process faster, and more efficient by reducing wait times. In other words, the introduction of self-checkout is geared towards reducing stress for both customers and retail staff, especially during the holiday, when stores experience greater peak times [1]. One research shows that for customers, the checkout experience is crucial, and can influence shopping decisions for almost 40% of

shoppers not loyal towards a single grocery location [2]. In contrast, this can suggest that merchants are willing to offer a more convenient checkout method, which can effectively retain customers and boost sales. With the rise of highly intelligent machines, being able to conveniently shop without the hassle of waiting in line has never been easier. Smart checkout systems date back to the early 1900's, when a man named Howard Schneider integrated the first retail self-checkout system, and nicknamed it "the service robot" [3]. Moreover, this had revolutionized the checkout system to a standard in which customers interact with a digitized machine able to construct the payment process seamlessly. Nevertheless, contact-less checkout systems can help reduce the interaction between social workers and customers, especially in a post-pandemic climate. To further illustrate, the integration of a smart checkout system is willing to eliminate the need of cooperation in order to protect the health of each customer from contracting germs, bacteria, etc. In general, customers are not required to input their payment methods, loyalty cards or scan merchandise directly, as in other words, eliminate the inevitability for manual process. Consequently, the goal for smart checkout systems is to protect and provide a faster, pleasant experience when shopping, without having to worry about slow processing times and extensive delays.

Due to the COVID-19 pandemic, contact-less services are profoundly divine towards protecting the health of customers and staff. There is an estimation that the use of contact-less payments has grown by 20 percent since the start of the pandemic [4]. Nevertheless, big grocery store chains and school stores, where large gathering occurs, the self-checkout system is not yet extensively implemented or integrated using low efficiency [5]. Moreover, a highly efficient and reliable self-checkout system can help avoid the dilemma from large crowd gathering, nevertheless maintain proper social distancing from one another, that may prosper businesses and reduce the labor cost [6]. Furthermore, this can benefit any corporation, despite how large an organization is.

Nowadays, there are mainly two kinds of widely accepted checkout systems based on barcode and RFID respectively [7].

The barcode was first invented in 1949, by a man named Joe Woodland. His idea sparked over the adoption of barcoded commodity circulation, book management, postal management, and banking system [8]. To illustrate, a barcode is a graphic identifier that formerly represents a set of information by arranging black bars with white spaces, applying different widths according to certain types of coding rules. In contrast, the essence of a barcode is to demonstrate a group of numbers in a graph form, whereas this graph is beneficial by being promptly recognized from a scanner, and also read out the numbers' group. If the barcode is damaged, bent or cannot be read by a scanning gun, due to reflective reasons, there is an option to manually input each individual number. In fact, the usage of a barcode shall rely on the network along with the database. The scanned information contains solely plain numbers, besides the more detained information is queried from the system databases accepting each scanned number. On the contrary, in the supermarket checkout scheme, the price information is traced back onto the supermarket's internal database.

RFID, radio frequency identification, technology is reposed of a tiny radio transponder, a radio receiver and a transmitter. Furthermore, RFID utilizes electromagnetic fields to identify items consisting of attached labels or tags [9]. To demonstrate, RFID technology can identify merchandise in batches, even over long distances. If a sensor is installed alongside the market's exit, any merchandise that passes through the checkpoint is adequately identifiable. Based on RFID technology, Bocanegra et al. proposed a self-checkout system called RFGo which enhanced the automatic checkout process [10]. RFGo consists of a customized RFID reader, capable of observing multiple tags simultaneously. As a result, the observed data is sent onto a neural network for analysing the identification from each type of merchandise, while also determining its count. Their experiments presented that RFGo is adequately determining each type and quantity of merchandise accurately amidst a short time stamp. As a result, this can be deployed to places such as clothing stores, etc. However, for the rental stores, properly attaching an RFID id onto each individual merchandise, immensely increases the store owner's overall cost. Thus, this technique is not properly suitable for some retail stores' checkout systems.

In order to provide a solution for contact-free services with high efficiency, we have proposed an architecture for a self-checkout cashier system based on object detection and facial recognition, which has the ability to

immensely improve checkout efficiency. To demonstrate, this system has the ability to nevertheless scan a person's identity, determine if they subsist a rewards card, and even allow each individual to complete transactions using their identity, similarly to Apple's concept on the App Store.

This proposed smart self-checkout system is designed for any store that faces a problem where customers wait in long lines during checkout. If those stores adopt our system, customers will no longer be required to scan each item's barcode one by one at the counter. Instead, when items are placed below the camera, an object detection technique determines the category and amount of items, stationed for each proprietary scan alongside the calculated total price. Besides that, a facial recognition technique is utilized to procreate a connection between each customer along with their payment method. After each customer is registered in the store, they can save their preferred payment method onto our system. Therefore, next time the system will verify the identity of each customer, and the system can automatically process each checkout employing the saved payment method. To demonstrate, whenever a customer enters the store, they are able to quickly checkout by using their previous saved payment method and registered identity for a faster customer experience, reducing the amount of hassle previously experienced. Moreover, this miscellaneous payment process can be fully accomplished from the system resources. Even after the COVID-19 pandemic, the proposed system is still able to improve the efficiency of payment transactions and liberate human beings from tedious work conceived by today's standards. The contributions of our work include:

- 1) Design an Intelligent self-checkout system for utilizing contemporary machine vision techniques counting the latest object detection model.
- 2) Propose a user identity checking solution, in addition to an anti-spoofing feature for protecting the user's ownership.
- 3) Provide a contact-free checkout solution that reduces the risk of transmitting COVID-19.

The remainder of this paper is organized as follows. Related works are discussed in Section II. This contains work that inspired the proposed methods. Then, Section III introduces the techniques we used and background information. Afterwards, Section IV is the backbone of our paper, we illustrate the proposed architecture in more detail. Section V evaluates the proposed system and

provides a few future recommendations. Lastly, Section VI concludes the paper.

II. RELATED WORKS

In this section, different approaches and techniques, which are used for face recognition and object detection are explored. In addition, the automatic checkout systems utilizes machine vision techniques, which are proposed and reviewed by former researchers.

A. Face Detection

In order to recognize faces in video streams, first and foremost, allocate faces that appear in the frame. To demonstrate, there are two kinds of face detection methods: holistic features based and local feature based approaches [11]. Consequently, the local feature mainly uses the prior knowledge for treating faces as a combination of organ features, which can then detect the face according to features from the eyes, eyebrows, mouth, and other organs alongside the position relationship between one another. According to the holistic features based method, the face can be regarded as a whole pattern. From a statistical point of view, the face pattern space is constructed through a large number of face image samples, and furthermore, face existence is judged towards a similarity degree accordingly. Most face detection systems extend the Viola and Jones face detection model [12], mainly through the AdaBoost [13] learning algorithm, that imply Harr features and cascading classifiers [14]. For example, Zhang et al. presents a method, in which background subtraction is combined with Viola-Jones face detector for detecting and differentiating human faces from video sequences. Since this method only focuses on the foreground region, the proposed approach greatly reduces the computational cost and lowers the false positive rate [15].

Besides that, there are many other algorithms, which can detect faces from applying other features. For example, Fröba and Külbeck presented an algorithm, which applies local edge-orientation features, and as a result, achieved a 93% detection rate capable of running in real time [16]. Zhang et al. proposed a method that implies Multi-block Binary Patterns(MB-LBP) ability for face detection [17]. In other words, their experiments showcased that the proposed method can illustrate a 15% higher correct rate at a given false alarm rate of 0.001 compared to harr-like features, and 8% higher than the original LBP feature, consequently compressing less training time, due to MB-LBP being a smaller feature.

The above methods can detect each face by applying handcrafted features. With the continuous development of deep learning technologies and hardwares, researchers have started learning the outcome of convolutional neural networks in solving detection problems. Zhu et al. presented a face detection approach using region based CNN called Contextual Multi-Scale Region-based Convolutional Neural Network(CMS-RCNN) [18], in which contain regions of proposed components and regions of interest detection components for overcoming challenges from traditional face detection models, including heavy facial occlusions, extremely low resolutions and strong illumination. At present, the mainstream face detection methods are based on deep learning, and mainly include R-CNN series detection algorithms, consisting of R-CNN [19], fast-RCNN [20], YOLO [21], and SSD [22].

B. Face Recognition

In general, face detection is geared mainly towards determining whether there is an identifiable face, and how to recognize an identity problem in consideration towards the process of feature extraction. Furthermore, feature extraction aims to extract features from the face area itself, which then discriminates and compares the acquired face features, capable of distinguishing between different persons [23]. Traditionally, a general way for facial recognition scenarios tends to aim at extracting handcrafted features resulting from inputted 2D images in order to obtain a digital representation or encoding. Afterwards, this method compares known face encodings towards more similar ones. For instance, Ahonen et al. presented a face recognition method using a Local Binary Pattern(LBP) [24]. At first, the face area is divided into small areas, whereas, LBP histogram is first extracted and then connected for characterizing face images. Then, the face is recognized by the nearest neighbor classifier. Nowadays, researchers have proposed various deep learning architecture methods for extracting facial features and recognizing faces. Sun et al. presented a facial recognition model using deep neural networks called DeepID3 [25]. To demonstrate, this network first extracts features from the face region, which are then passed onto an additional Joint Bayesian model for face verification and identification. Consequently, in their experiment, their model achieves a 99.53% LFW face verification accuracy and 96.0% LFW rank-1 face identification accuracy. In conclusion, deep learning based face recognition is able to implicit expressions from faces with good robustness. However, its operation process involves more neurons, which can take longer.

C. Object Detection

Nevertheless, object detection is an underlying problem in the field of computer vision and image processing. On the contrary, it has recently been a hot topic on application and theory. The primary goal for object detection is to accurately locate categories and position information for various targets from an image or image sequence [26]. Moreover, traditional object detection algorithms rely on handcrafted features, such as Scale-invariant Feature Transform(SIFT) [27] and Histogram of Oriented Gradients(HOG) [28]. In theory, this can lead to lower accuracy, slower speeds and poor real-time performance. In recent years, deep convolutional neural networks have been more and more applied in the field of computer vision, thanks to the rapid development of deep learning where tremendous progress has been made in the field.

In 2014, R-CNN proposed by Grishick et al utilized CNN to be compatible with object detection [19]. To demonstrate, R-CNN first uses a selective search method that generates about 2000 search suggestion boxes for the input image, in an area that may contain objects. Next, different suggestion boxes are resized onto the same size, 227 by 227. After that, AlexNet [29] is used for extracting features from the suggestion boxes. Finally, service the SVM [30] classifier for determining the goal type and position.

One year later, Ren and others put forward a Faster R-CNN [31]. This network uses a region generation network(RPN) for integrating and generating candidate regions, feature extraction, target classification and position regression, all onto one model, which can recognize end-to-end target detection network training. Meanwhile, with the improvement of detection speed, Faster R-CNN becomes the first quasi-real-time deep learning target detection network.

By sparse sampling the original image, YOLO object detection algorithm calculates the loss of coordinates and categories in the central point region that contain the object [21]. To illustrate, in their experiment, the detection speed can be as fast as 45 frames, meeting real-time demand. YOLO algorithm is based on the global information of image detections. Compared with the Fast R-CNN algorithm, the error rate of error detection has been greatly reduced, whereas the generalization ability is strong. However, the YOLO algorithm only detects target objects whose center point falls on the grid. On the other hand, when there are multiple object center points in a grid, the phenomenon from poor position accuracy

and low recall rates are transpired.

In addition, several studies have demonstrated the possibility of using computer vision to implement checkout systems. Some of them have even been commercialized.

Amazon launched a new type of contactless grocery store, Amazon Go on January 22, 2018, without necessarily installing a checkout counter [32]. To demonstrate, each customer walks into the store, by scanning their phone onto a metallic plate that contains a unique QR Code, after which they go to the desired items, take the items, and then walk out without having to wait in line for checking out. After exiting the store, the payment is automatically charged and processed onto the customer's Amazon account. Consequently, Amazon refers to this seamless shopping experience as "Just Walk Out". On the contrary, the technologies being utilized in Amazon Go are similarly compared towards driverless cars that comprise computer vision recognition, multi-sensor fusion, and deep learning algorithms [33]. Amazon Go revolutionized the exploration of future technology, and as a result demonstrated how machine vision can change our lives. Generally, the cost figure of a single Amazon Go store could range within millions of dollars, whereas the technology is not yet widely available on a global scale [34].

Meanwhile, in order to generate a deep learning based checkout system, for serving a broader customer group, some researchers are proposing an affordable solution based on image classification and object detection. Femling et al. represented a system with a capacity to identify fruits and vegetables over lapsing, abides employing convolutional neural networks(CNNs) [35]. Furthermore, the proposed system contains a camera set for the Raspberry Pi, focused on a scale in front of the camera. To demonstrate, users can place a single preferred choice of fruit or vegetable onto a scale where a photo is shot and processed by a CNN classifier to which it can determine the specified fruit, or vegetable. One the weight and correspondent item type is detected, the system can automatically generate the subtotal. This corresponding system is lightweight, as it can run on a Raspberry Pi, on the other hand, it is not adequately efficient as it can only process one type of item amid each checkout.

Wu et al. proposed an architecture for an intelligent self-checkout system that uses an object detection model that proposes the determination type and count of purchased merchandise so that the total is automatically calculated [36]. In their study, the object detection model YOLO [21], is utilized to detect each type and bounding

box of definitive items. In general, this detected information can be used for calculating the subtotal, even if multiple items coincide within the image, whereas the proposed system can calculate the overall subtotal once. Moreover, as a checkout system, it is still necessary for customers to interact. For example, determine the identity of a customer in addition to automatically charging their payment accounts. In summary, the efficiency of checkout systems are greatly improved with the addition of new features.

III. TECHNIQUES USED IN THE STUDY

Our plan is to aid the checkout process for stores with the help of computer graphics. Currently, the checkout process is roughly divided into three steps:

- 1) Scan the merchandise in order to obtain the category, quantity, and price.
- 2) Customers may present their membership card, if they have signed up.
- 3) Customers get informed about their subtotal, then immediately after, excerpt a payment method in order to considerably complete their payment.

Nevertheless, the first step can be accomplished using an object detection model, which has been proven by [36]. Without a doubt, the main goal of object detection is to figure out all of the interesting targets, or objects, in the image to better determine their categories and positions, that is one of the core problems in the field of machine vision. Furthermore, we can use the object detection model for analyzing each type and number of goods purchased from customers, and then obtain the commodity price information from the backend database system for calculating the total amount due. In order to implement step 2 and 3, the customer's identity must be verified. Consequently, none of the researchers have implemented this process in their related works. However, the checkout efficiency can be further improved if the process is handled by the machine. Some common ways of verifying an identity include: fingerprint identification, voice recognition, retinal plus iris plus facial scanning, and recognition, similar to common practices found on modern computational devices [37]. Amidst the exception of voice recognition, all of the methods can be implemented using computer vision technology. Otherwise, fingerprint, face and iris recognition all require an additional device for collecting information. Meanwhile, face recognition turns out to be the least valuable, and most efficient method.

To train our model, we need to find a dataset containing merchandise of the market with labeled type

and bounding box information. The reason why we don't generate our dataset is that to train a robust and generalized model, the dataset needs to be large enough. While collecting photo and labeling items can cost a huge amount of time, we prefer to use existing dataset for training the object detection model.

After the model is trained, we will evaluate the model with several metrics. First we will test the time cost for our model to make predictions on one image, since our aim is speeding up the checkout process. Second, we will evaluate the accuracy of our model which includes mean Average Precision (mAP) for a given Intersections over Union (IoU) and the average precision for each item category.

In order to implement step 2 and 3, the customer's identity must be verified. However, none of the researchers have implemented this process in related works. However, the checkout efficiency can be further improved if the process is handled by the machine. Some common ways of verifying an identity include: fingerprint identification, voice recognition, retinal plus iris plus facial scanning, and recognition, similar to common practices found on modern computational devices[1]. Amidst the exception of voice recognition, all of the methods can be implemented using computer vision technology. Otherwise, fingerprint, face and iris recognition all require an additional device for collecting information.

Meanwhile, face recognition turns out to be the least valuable, and most efficient method. To demonstrate, facial recognition can be divided into three stages: face detection, face coding, and feature matching. Face detection is the process of finding faces from an image. This process eliminates irrelevant information so that the face encoding can focus on specific parts. To illustrate, face coding can be described as the process of extracting facial features from an image, and encode that information onto a digital combination. Typically, facial features are encoded into a series of numbers, which are usually represented as vectors. In the process of face matching encoded features, each vector is matched one by one from the database. If two vectors are similar enough, a match occurs to further signalize a connection from the faces' identity.

Moreover, the conventional face recognition does not check whether the face is "live" or only an image. Since the identity of the customer is associated with their payment method, we must conduct an anti-spoofing detection that inquires if the face obtained is from a real person or a photograph. Generally, identity detection

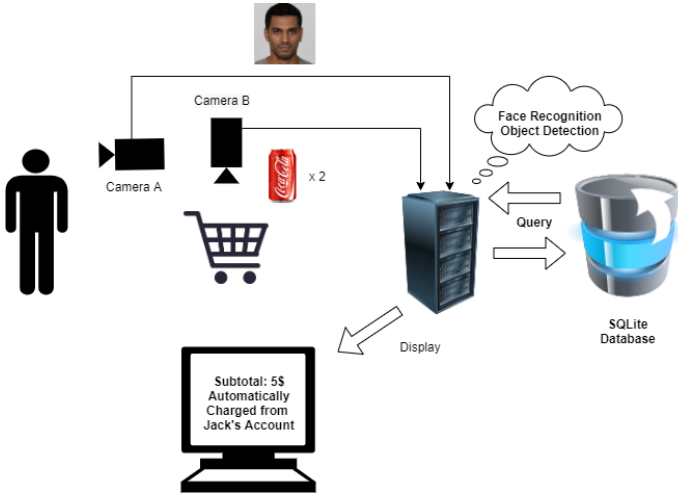


Fig. 1: Structure of Proposed Self-checkout System

only works if a face has passed the anti-spoofing detection. After the identity of the customer is determined, we can attain their membership, payment method and a further detailed private information from the database. In conclusion, the computer will then automatically deduct the paid amount from the member's account.

IV. PROPOSED METHOD

The proposed self-checkout cashier system is based on object detection and facial recognition techniques. In contrast, customers are no longer required to scan each individual items' barcode one by one at the counter. Instead, when a customer places each individual item below the camera, the object detection technique will actuate the category alongside the quantity of items based on each proprietary scan. In addition, the object detection technique will also calculate the total price. To demonstrate, facial recognition techniques utilize a connection between each individual customer and their payment method. After each customer registers, they have an option to save their payment method in our system for future preferences. And then next time, the system will verify the identity of each customer amongst processing each checkout using their saved preferences. Furthermore, the miscellaneous payment process can be fully accomplished by the system resources. Fig.1 illustrates the architecture of the proposed system.

A. Face Recognition

The face recognition function is implemented by a python library face_recognition [38]. It utilizes dlib [39]'s the state-of-the-art deep learning based face recognition tools for accomplishing various face recognition

related work, and to further simplify the process of methods calling. The first task for face recognition is face detection. To illustrate, this can detect the bounding boxes around each face in an image. There are two models available for face detection. The first model is based on Histogram of Oriented Gradients (HOG) [28], which is similar to a handcraft feature. The second model is based on Convolutional Neural Networks(CNN) [29]. Nevertheless, the HOG is faster but is less accurate than the CNN based model, whereas the CNN based model can be accelerated by GPU/CUDA devices. We have chosen the HOG based model as the default option. The location of the faces are represented as a 2D array. First of all, we need to determine the number of faces that appear on an image. Only the frame with a single face will be further processed.

Once the location of the face is resolved, afterwards the face is cropped from the original image and passed onto the anti-spoofing model. The anti-spoofing model we used is adopted from [40]. To demonstrate, this technique can be used to distinguish real faces from printed paper photos, display screen of electronic products, silicon mask, 3D human image, etc. Consequently, real faces along with fake faces contain drastic differences on the Fourier spectrum in frequency domain that are up to a certain extent, and can be classified by convolutional neural networks. In general, the pre-trained model is offered in their repository, therefore, we will apply their model directly.

Eventually, only the passing anti-spoofing face check will be sent onto the identity check model. On the other hand, in order to check for each users' identity, their face will need to be encoded for a machine readable feature vector. The input face crop aims to be encoded to a 128 dimensional face encoding vector. Afterwards, the face encoding will be compared with each known face encoding that is eventually stored in the database. Generally, the one containing the lowest distance that is less than the tolerance factor T will be regarded as the same face with the same identity. Otherwise, the customer is regarded as a first time user that is required to encode his/her face and save their personal information which includes their name, email address and payment method. In summary, the overall user identity verification process can be represented as a flowchart in Fig.2.

B. Object Detection

Moreover, we are going to use the object detection model for detecting the type and location of the products. With this information, we can justify the quantity each

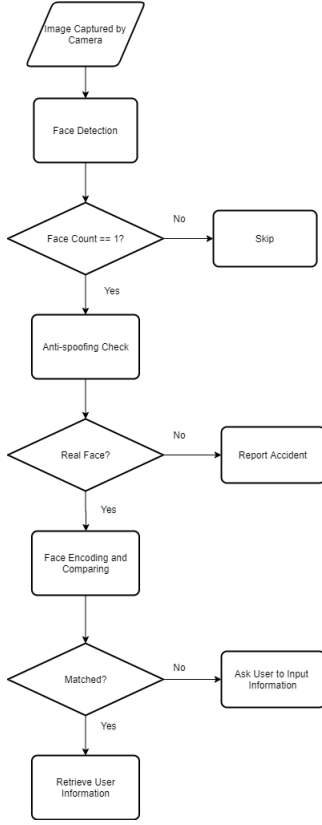


Fig. 2: The Flow Chart of User Identity Verification Process

customer shall pay, and can also mark the detected objects by presenting them. It is very critical to choose the model that best fits our system since there are plenty of object detection models, each designed for specific tasks. The accuracy rate and processing speed are two important metrics for evaluating the object detection model. In our scenario, the accuracy rate should be more important than the processing speed, because the accuracy rate determines whether our system can accurately calculate the subtotal, that is crucial. In addition, we don't need to process live video streams, but rather analyze individual images. On the contrary, the processing speed is less important when it comes to executing our object detection model. On the other hand, the GPU can vastly impel the neural network's image processing, as there is sufficient space for GPU to be installed on the checkout system, so we can ignore the impact on the processing delay. Anton Rigner evaluated the accuracy of three state-of-the-art object detection models for retail merchandise detection in his master thesis [41]. His result indicated that Mask R-CNN [42] outperformed RetinaNet [43] and YOLO V3 [44] on mean average

TABLE I: Performance Comparison among Mask R-CNN, RetinaNet, YOLO v3 on mAP

	Model (1920 x 1080)	mAP	mAP50	mAP75
[41]	Mask R-CNN	72.3	93.1	79.3
	RetinaNet	71.8	92.6	80.2
	YOLOv3	59.9	82.2	70.0

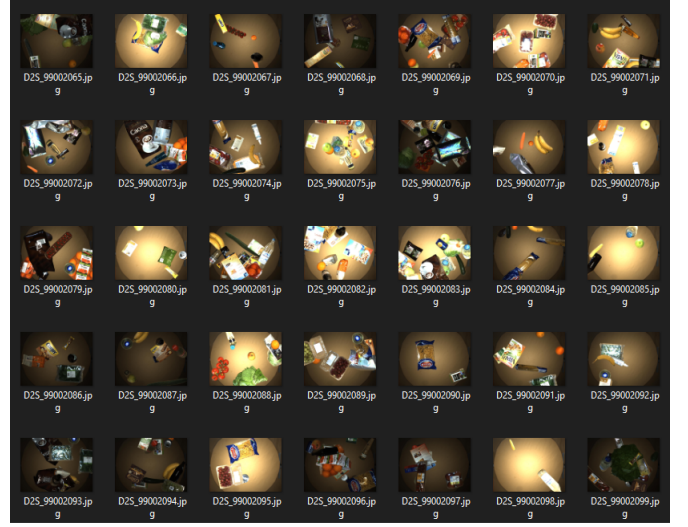


Fig. 3: Part of Images in D2S Dataset

precision(mAP). Table I shows the comparison on mAP for Mask R-CNN, RetinaNet, and YOLO v3.

C. Dataset

In order to train the object detection model, we are going to use MVTEC Densely Segmented Supermarket Dataset(MVTEC D2S) [45]. D2S contains 21,000 high-resolution images that belong to 60 categories. Furthermore, the objects include groceries and everyday products that are necessary for our project. Nevertheless, the same objects are photo oped from different angles and concealed by different lighting conditions in order to ensure the diversity of the training data. In addition, the dataset includes images with pixel-wise labels for all object instances that can be used for training the object detection model. Fig. 3 shows parts of the images in the D2S dataset.

D. Database Design and Automatic Checkout

In order to achieve an automatic checkout, the appropriate databases are needed to be designed and integrated onto the system. To illustrate, two tables: Customer, and Product are constructed to store both customer and merchandise information. Within the Customer table, there are 6 attributes corresponding to a customer's

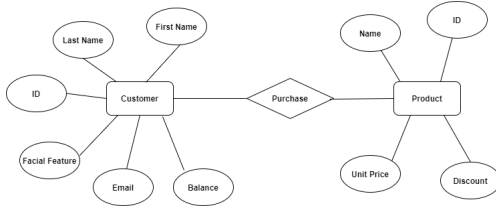


Fig. 4: Database Entity Relationship Diagram for Designed Database

information: first name, last name, id, facial feature, email and balance. In fact, the facial feature attribute corresponds to the facial feature vector stored as a Pickle object. Whereas, the table Product accommodates 4 attributes: name, unit price, discount and id. To illustrate, the selling price for the product can be calculated as the product of the unit price and discount. Due to the output from the object detection model items being numerically represented, the attribute id helps the model determine the specific product. Fig 4. displays the entity relationship diagram of our database.

Furthermore, the face recognition and object detection model are connected with help from the databases in order to make the automatic checkout process feasible. First and foremost, facial features from the customer are extracted by our system and compared with the records stored in the Customer table one by one. Next, find the customer with the closest facial features, in addition, if the distance from the two vector features is lower than the given threshold, the customer's identity has been determined. Then the photo containing the products needed for checkout, is fed onto the object detection model. This model then outputs the ids, bounding boxes, and the confidence score for each item. In general, the ids can be used to determine the name and the price of items by querying the Product table. Afterwards, the subtotal can be calculated automatically. In fact, since we know the identity of the customer, the checkout process can be implemented by subtracting the amount from the user's balance. In addition, we can visualize the result by marking the bounding boxes, item names, and price information onto the input photo, and then display it for the customer. To summarize, the overall checkout process can be depicted by the flowchart of Fig. 5.

V. EXPERIMENT AND RESULT

For the next part of our research proposal, we are going to provide information on the training model, including the platform, the selection of hyperparameters

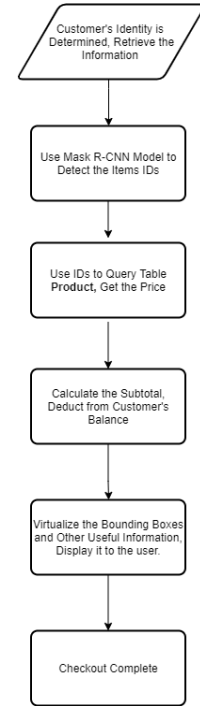


Fig. 5: Checkout Process Flowchart

TABLE II: Experiment Platform Configuration

Operating System	Ubuntu 20.04.1 LTS
CPU	Intel(R) Xeon(R) CPU E5-2670 v3 @ 2.30GHz x 48
RAM	512GB
GPU	NVIDIA Tesla M40 x 4
Libraries	Keras 2.31, TensorFlow 2.1.0, PyTorch 1.6.0, CUDA 10.1

and the training process. In addition to that, we will evaluate our model that is based on several metrics.

Table II shows the platform configuration used to train our models. Including the hardwares, operating system and libraries we use to train our models. In the experiment, the model is trained on 13 out of 60 categories of D2S dataset for testing. The dataset configuration shows in Table III.

In fact, for the Mask R-CNN model, we do not need to train it from scratch, but will instead train it based on the pre-trained model. In general, the pre-trained model is trained on the COCO dataset [46], which is a large-scale object detection, segmentation, and captioning dataset that contains more than 200,000 labeled images with 80 object categories. Therefore, the pre-trained model

TABLE III: Dataset Configuration

Number of Images of Training Dataset	1500
Number of Images of Validation Dataset	640
Image Scale	1920 × 1440

TABLE IV: Critical Parameters of Mask R-CNN model for Training

Backbone	ResNet101
Image Shape	1024 × 1024 × 3
RPN Anchor Scales	32, 64, 128, 256, 512
Detection Minimum Confidence	0.7
Number of Classes	13
Learning Momentum	0.9
Learning Rate	0.001
Steps per Epoch	1000
Weight Decay	0.0001

already contains a strong generalization ability for a large scale dataset, which generally means that it can extract deep and abstract core features. To train from pre-trained model, we first load the weights of Mask R-CNN model trained on COCO dataset [47]. During the training, we can freeze some layers that might extract features, but instead train the layers that do the task of classification.

Consequently, the training process can be divided into 3 stages. The first stage will only train the head of the Mask R-CNN that includes the Region Proposal Network, classifier and the mask heads of the network. We are also going to train the model for 40 epochs. Meanwhile in the second stage, we start to train the backbone of the Mask R-CNN model. In other words, for our model, we will use ResNet101 as our backbone network. The second stage will train up to 10 epochs, whereas the overall model will be trained for 160 epochs in the final stage. By training on different layers for different epochs, the model will concentrate on learning new features, but will not discard older experiments. Lastly, in stage 3, after the model is trained, we then evaluate the performance on the accuracy obtained by each class object. Table 3 shows the parameter we use for training the model.

The output of our model is shown in Figure 6. Clearly, we can see that our model can precisely predict the locations and categories of different objects. Nevertheless, the bounding boxes are marked along with the name of the items and the confidence score. In addition, by querying from the database, the subtotal of the items shown in the graph is automatically calculated and displayed.

In order to evaluate the performance of the trained model, we test our model on 200 out of 640 images of D2S validation dataset, and calculate the average precision of each class for Intersection over Union (IoU) [48] equals 0.5. IoU is defined by the area of overlap divided by the area of union, a larger IoU means the



Fig. 6: Object Detection Prediction Model. The Name and Confidence Scores of each Item are Marked along with the Bounding Boxes. The Subtotal Information is Displayed at the Top Left Part of the Image.

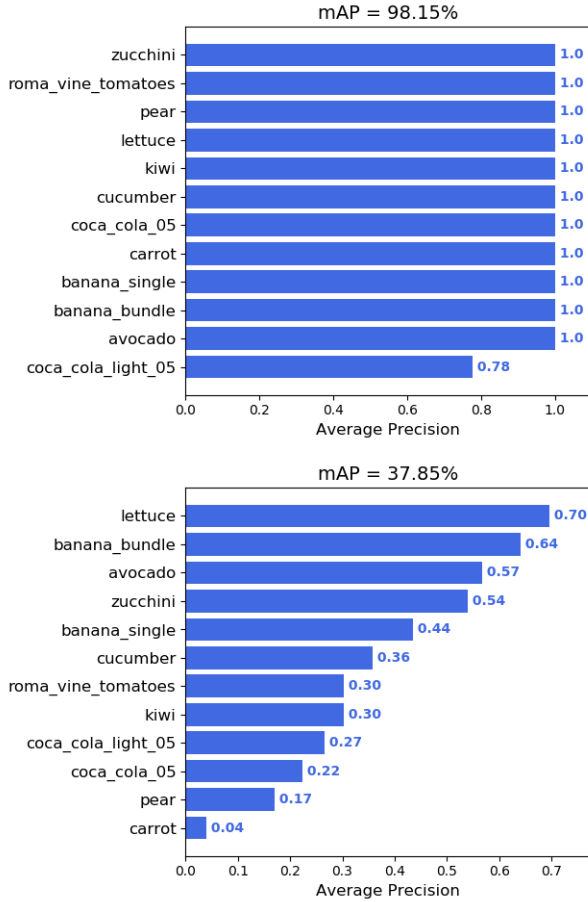


Fig. 7: Average Precision for Each Category (UoI=0.5). Up: mAP on Training Dataset. Down: mAP on Validation Dataset

predicted boundary overlaps more area with the ground truth.

$$IoU = \text{Area of Overlap} / \text{Area of Union}$$

The Average Precision (AP) is calculated with PASCAL VOC metric [49]. In contrast, the mean Average Precision can be computed to represent the overall performance of our model. Fig.7 shows the mean average precision of each category for training (up) and validation (down) dataset.

We are also testing our face recognition and anti-spoofing model. As we discussed earlier, the model will check the liveness first, then identify the face only if it passes the anti-spoofing check. Fig 8 shows that the face doesn't pass the anti-spoofing check and the face is recognized.

At the same time, we need to point out that our object detection model does not perform well in the test

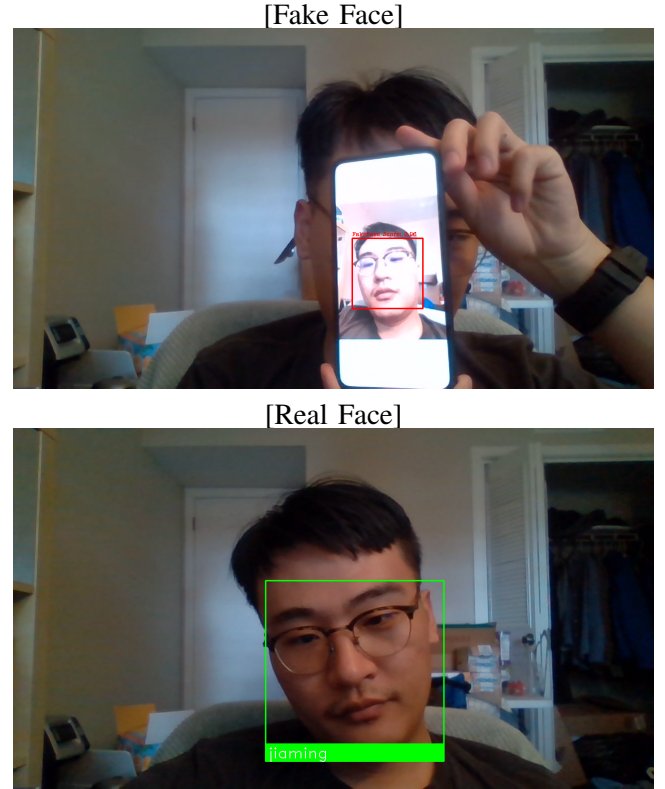


Fig. 8: Output of User Identity Verification System

data as expected. The main reason is due to the change in the object background, as comparatively the Mask R-CNN classifies the objects background into separate specified categories. However, the data set in our model, D2S, contains a single tonal texture for the background. Once the detected objects are placed against different backgrounds, the accuracy of prediction decreases. In addition, facial recognition with applied 2D images is certainly not the most reliable method of authentication. Even though we have considered putting fraud detection on the image before authenticating it, the limitations on 2D images precisely allow some fraud methods to bypass the authentication process.

In the next step, we are planning on constructing our own dataset in order for our model to have a stronger ability in distinguishing backgrounds. To illustrate, our dataset should be more challenging, with more occlusion and overlapping between certain objects. This would make our models more robust that will ultimately be placed onto a real commercial usage. In fact, we will also consider using dual-camera systems in future studies for authentication. The dual-camera system can concisely help obtain the depth information of the space, which can eventually help determine the liveness of customers

more precisely.

VI. CONCLUSION

In this paper, we have proposed an architecture of intelligent self-checkout systems. To demonstrate, this system uses state-of-the-art object detection deep learning model with Mask R-CNN, that is able to detect multiple items purchased at once by customers. Moreover, by querying from the database, we can get the unit price and discount information of the items with the final justified quantity each customer will need to pay, that is generated from our system. In addition, we are proposing a customer identity recognition system that is based on facial recognition techniques. In general, the system includes facial feature encoding, matching, anti-spoofing along with other functionalities. Under the premise of ensuring each customer's privacy, we initiate computer vision technology to better reduce the cumbersome scanning process of cashiers, and customers that do not wish to repeat the current, ongoing complicated payment process.

At the same time, we need to point out that our object detection model does not perform well in the validation data as expected. The main reason is due to the limited images in the training dataset. However, that is mainly because we only test part of categories of the D2S dataset. When the model is tested in the validation dataset, unknown items will confuse our model thus yield accuracy prediction. The performance of our model will be better if we train it on the overall dataset. In addition, facial recognition with applied 2D images is certainly not the most reliable method of authentication. Even though we have considered putting fraud detection on the image before authenticating it, the limitations on 2D images precisely allow some fraud methods to bypass the authentication process.

In the next step, we are planning on constructing our own dataset in order for our model to have a stronger ability in distinguishing backgrounds. To illustrate, our dataset should be more challenging, with more occlusion and overlapping between certain objects. This would make our models more robust that will ultimately be placed onto a real commercial usage. In fact, we will also consider using dual-camera systems in future studies for authentication. The dual-camera system can concisely help obtain the depth information of the space, which can eventually help determine the liveness of customers more precisely.

REFERENCES

- [1] "How to implement self-checkout system in your retail business?" Feb 2020. [Online]. Available: <https://solumesl.com/how-to-implement-self-checkout-system-in-your-retail-business/>
- [2] M. Browne, "For customers, the waiting is the hardest part," Aug 2018. [Online]. Available: <https://www.supermarketnews.com/retail-financial/customers-waiting-hardest-part>
- [3] Blog, "The evolution of self-service technology," Aug. 2018, section: Blog. [Online]. Available: <https://www.qikserve.com/self-service-tech-a-history/>
- [4] N. C. Craig Guillot, "Covid-19 boosts the prospect of contactless commerce," May 2020. [Online]. Available: <https://nrf.com/blog/covid-19-boosts-prospect-contactless-commerce>
- [5] T. Rueter, "What's next for food retail self-checkout?" [Online]. Available: <https://progressivegrocer.com/whats-next-food-retail-self-checkout>
- [6] V. Saraogi, "Contactless payments will be the new normal for shoppers in the post covid-19 world," May 2020. [Online]. Available: <https://newsroom.mastercard.com/asia-pacific/2020/05/20/contactless-payments-will-be-the-new-normal-for-shoppers-in-the-post-covid-19-world/>
- [7] Brightpearl, "Barcodes vs rfid: Why barcodes still win," Aug 2020. [Online]. Available: <https://blog.brightpearl.com/barcodes-vs-rfid>
- [8] "The history of the bar code," Sep 2015. [Online]. Available: <https://www.smithsonianmag.com/innovation/history-bar-code-180956704/>
- [9] M. Bhuptani and S. Moradpour, *RFID field guide: deploying radio frequency identification systems*. Prentice Hall PTR, 2005.
- [10] C. Bocanegra, M. A. Khojastepour, M. Y. Arslan, E. Chai, S. Rangarajan, and K. R. Chowdhury, "Rfgo: a seamless self-checkout system for apparel stores using rfid," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 2020, pp. 1–14.
- [11] S. Tseng, "Comparison of holistic and feature based approaches to face recognition," 2003.
- [12] M. Jones and P. Viola, "Fast multi-view face detection," *Mitsubishi Electric Research Lab TR-20003-96*, vol. 3, no. 14, p. 2, 2003.
- [13] Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting," *Journal-Japanese Society For Artificial Intelligence*, vol. 14, no. 771-780, p. 1612, 1999.
- [14] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, vol. 1. IEEE, 2001, pp. 1–1.
- [15] L. Zhang and Y. Liang, "A fast method of face detection in video images," in *2010 2nd International Conference on Advanced Computer Control*, vol. 4. IEEE, 2010, pp. 490–494.
- [16] B. Fröba and C. Külbeck, "Real-time face detection using edge-orientation matching," in *International Conference on Audio- and Video-Based Biometric Person Authentication*. Springer, 2001, pp. 78–83.
- [17] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Z. Li, "Face detection based on multi-block lbp representation," in *International conference on biometrics*. Springer, 2007, pp. 11–18.
- [18] C. Zhu, Y. Zheng, K. Luu, and M. Savvides, "Cms-rcnn: contextual multi-scale region-based cnn for unconstrained face

- detection,” in *Deep learning for biometrics*. Springer, 2017, pp. 57–79.
- [19] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.
 - [20] R. Girshick, “Fast r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.
 - [21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
 - [22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *European conference on computer vision*. Springer, 2016, pp. 21–37.
 - [23] V. Bruce and A. Young, “Understanding face recognition,” *British journal of psychology*, vol. 77, no. 3, pp. 305–327, 1986.
 - [24] T. Ahonen, A. Hadid, and M. Pietikainen, “Face description with local binary patterns: Application to face recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.
 - [25] Y. Sun, D. Liang, X. Wang, and X. Tang, “Deepid3: Face recognition with very deep neural networks,” *arXiv preprint arXiv:1502.00873*, 2015.
 - [26] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, “A survey of deep learning-based object detection,” *IEEE Access*, vol. 7, pp. 128 837–128 868, 2019.
 - [27] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
 - [28] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, vol. 1. IEEE, 2005, pp. 886–893.
 - [29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
 - [30] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, “Support vector machines,” *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.
 - [31] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *Advances in neural information processing systems*, 2015, pp. 91–99.
 - [32] J. Fingas, “Amazon’s checkout-free store opens to the public january 22nd,” Feb 2020. [Online]. Available: <https://www.engadget.com/2018-01-21-amazon-go-opens-to-public-january-22.html>
 - [33] P. lint, “What is amazon go, where is it, and how does it work?” Feb 2020. [Online]. Available: <https://www.pocket-lint.com/phones/news/amazon/139650-what-is-amazon-go-where-is-it-and-how-does-it-work>
 - [34] A. Cheng, “Amazon go looks to expand as checkout-free shopping starts to catch on across the retail landscape,” Nov 2019. [Online]. Available: <https://www.forbes.com/sites/andriacheng/2019/11/21/thanks-to-amazon-go-checkout-free-shopping-may-become-a-real-trend/3f947bc0792b>
 - [35] F. Femling, A. Olsson, and F. 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)*. IEEE, 2018, pp. 9–15.
 - [36] B.-F. Wu, W.-J. Tseng, Y.-S. Chen, S.-J. Yao, and P.-J. Chang, “An intelligent self-checkout system for smart retail,” in *2016 International Conference on System Science and Engineering (ICSSE)*. IEEE, 2016, pp. 1–4.
 - [37] Ulisticadmin, “Common authentication methods used for network security,” Aug 2019. [Online]. Available: [https://www.alliancetechnpartners.com/common-authentication-methods-used-network-security/:text=Common biometric authentication methods include,not ideal for some industries](https://www.alliancetechnpartners.com/common-authentication-methods-used-network-security/:text=Common%20biometric%20authentication%20methods%20include,not%20ideal%20for%20some%20industries).
 - [38] Ageitgey, “facerecognition.” [Online]. Available: https://github.com/ageitgey/face_recognition
 - [39] “dlib_c_i_b_r_a_r_y.” [Online]. Available: <http://dlib.net/>
 - [40] Minivision-Ai, “minivision-ai/silent-face-anti-spoofing.” [Online]. Available: <https://github.com/minivision-ai/Silent-Face-Anti-Spoofing>
 - [41] A. Rigner, “Ai-based machine vision for retail self-checkout system,” *Master’s Theses in Mathematical Sciences*, 2019.
 - [42] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.
 - [43] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.
 - [44] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.
 - [45] P. Follmann, T. Bottger, P. Hartinger, R. König, and M. Ulrich, “Mvtc d2s: Densely segmented supermarket dataset,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 569–585.
 - [46] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *European conference on computer vision*. Springer, 2014, pp. 740–755.
 - [47] W. Abdulla, “Mask r-cnn for object detection and instance segmentation on keras and tensorflow,” <https://github.com/matterport/MaskRCNN>, 2017.
 - [48] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, “Generalized intersection over union,” June 2019.
 - [49] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results,” <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>.