IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Volume 14, PP 19-23

Self Checkout System Using Image Detection

Tanmay Singh[1], Keshav Yadav[2], Ketan Kandalkar[3], Priya Porwal[4]
[1](Computer Science Department, Atharva College Of Engineerning/ Mumbai University, India)
[2](Computer Science Department, Atharva College Of Engineerning/ Mumbai University, India)
[3](Computer Science Department, Atharva College Of Engineerning/ Mumbai University, India)
[4](Assistant Professor, Atharva College Of Engineerning/ Mumbai University, India)

Abstract: *Most of the current self-checkout systems rely on barcodes, RFID tags, or QR codes attached to items to distinguish products. This paper proposes an Intelligent Self Checkout System (ISCOS) embedded with a single camera to detect multiple products without any labels in real-time performance. The bounding boxes are annotated by background subtraction with a fixed camera to avoid time-consuming process for each image. The Database for storing the products is used to generate the bill. The contribution of this work is to combine deep learning and data mining approaches to real-time multi-object detection in image-based checkout system. [2]*

Key Words: *Data mining, Deep learning, Multi-object detection, Self-checkout, Smart retail.*

I. Introduction

The global self-checkout (SCO) shipments steadily grow in 2010 to 2019, according to Retail Banking Research (RBR) [1], which is a consulting firm provides services to organizations active in retail automation, banking and payments. In 2014, Sumac set a self-checkout system in the physical store in Slovenia for checkout effects. Table 1 is the statistical results of customer acceptance to SCOs by using barcode scanning to make purchases. Consumers prefer SCOs to current checkout while buying more than five items and 62.4% of them had used other self-service terminals.

Demographic characteristics	Freq.	%
Using SSCT when buying up to		
3 items	7	4.1
5 items	26	15.3
8 items	35	20.6
10 items	40	23.5
Number of items is not important	62	36.5
Using also other self-service terminals (self-service check-in at airport, etc.)		
Yes	106	62.4
No	64	37.6

Table 1. *Customer Acceptance to SCOs*

Rossetti proposed the simulation modeling of customer checkout configurations and recorded its impact on checkout efficiency [9]. Checkout cases are divided into five scenarios. Scenario 1 is the traditional checkout system served by cashier scanning barcodes. Scenario 3 presents that customers choose SCO lanes by a number of items. As a result, SCOs save one minute of total average waiting time for users compared to Scenario 1, see SCO database.

	Scenario 1 Avg. (s.d.)	Scenario 2 Avg. (s.d.)	Scenario 3 Avg. (s.d.)	Scenario 4 Avg. (s.d.)	Scenario 5 Avg. (s.d.)
#Shoppers in Store	106.85 (4.012)	106.15 (2.887)	106.15 (2.887)	106.15 (2.887)	105.86 (3.487)
Time Shopping	53.63 (.628)	53.48 (0.639)	53.48 (0.639)	53.48 (0.639)	53.55 (0.541)
Items Per Customer	89.29 (1.069)	88.99 (1.198)	89.00 (1.201)	88.96 (1.213)	89.13 (.853)
Payment Time	1.38 (.016)	1.39 (0.015)	1.39 (0.016)	1.39 (0.014)	1.39 (0.015)
Checkout Time	7.04 (.144)	7.00 (0.162)	7.00 (0.167)	6.99 (0.164)	6.99 (0.125)
Bagging Time	1.86 (.022)	1.85 (0.025)	1.85 (0.025)	1.85 (0.025)	1.86 (0.018)
#In Store	140.10 (10.53)	135.85 (8.69)	134.71 (8.074)	139.20 (7.682)	128.54 (4.785)
Time In Store	70.30 (3.164)	68.27 (3.421)	67.68 (2.878)	69.77 (2.953)	64.95 (0.94)
Total Wait Time	6.37 (2.858)	5.92 (3.347)	5.33 (2.863)	7.46 (2.788)	2.49 (0.473)

Table 2. *SCO Database*

Rather than dealing with the 1 Waiting Time In this paper, an Intelligent Self-Checkout System (ISCOS) is proposed to address the non-barcode and camera reduction solution to current image-based checkout system. The system sums up the products overlooked by single camera set above the counter. Also, product detection prevents from fraud that users scan the cheaper items- labels rather than the more expensive ones [1]. We construct market database as training set to facilitate the research of shape matching and product recognition with emerging deep learning techniques, You Only Look Once (YOLO) [1] and ImageNet-trained CaffeNet1 [1]. Each product is classified into one of shape categories by YOLO, using convolutional features to detect potential bounding boxes. After system has been processed by the training set, it deals with multiple instances of the same product and those with different brands in the image.

II. Related Work

Self-checkout (also known as self-service checkout and as semi-attended customer-activated terminal, SACAT) machines provide a mechanism for customers to process their own purchases from a retailer. They are an alternative to the traditional cashier-staffed checkout. The customer performs the job of the cashier themselves, by scanning and applying payment for the items. In this type of system, the customer has to first scan the barcode of all the products he or she intends to purchase manually. Once products are scanned individually customers are required to input the type of product into the system then the system calculates the bill. As the name suggests the system is semi-automated and also the time required to scan the barcode for each product is high in case of a huge number of products to be purchased.

In order to reduce the amount of time required to checkout, the process of scanning the barcode of each object individually can be changed to scanning all the objects at once using the object recognition methods. There are many different ways in the which object recognition can be done.

Object recognition using the appearance-based method: - In this method, the object within an image is recognized by first taking the picture of an image and then comparing it with a database containing multiple images of the same object as the original image under different conditions such as a change in angle, change in color, change in light etc. The set containing the image which is closest to the input image is given as the result. [1] Object recognition using the feature-based method: -This method makes use of image contour (boundaries of objects within an image) The approach can be broken down into three main sections. a) The image is translated into a set of contours. b) The contours are reduced into a set of simple geometric features that describe the most significant parts of the original image's) The features are processed to form a search word, which is compared to a database to find which objects are likely to be in the image. [3]

Object recognition using machine learning. Machine learning is a data analysis technology that teaches computers to do what comes naturally to humans and animals: learn from experience. Machine learning algorithms use computational methods to "learn" information directly from data without relying on a predetermined equation as a model. The algorithms adaptively improve their performance as the number of samples available for learning increases. Basically, you make the machine learn by teaching it using samples. There are many different open source libraries available from where images can be accessed to train the machine such as ImageNet, OpenCV etc. In machine learning, the concept of convolutional neural networks is used. Image Recognition using Genetic Algorithm:-In Genetic algorithm the concept of genes, chromosomes, and population are used. In this type of technique, the machine tries to improve the accuracy with which the object is recognized each time runs, it then keeps the genes that provide the best result and discard the genes that are not accurate. [6] There are several instances of machine vision checkout system for retail. Kroger was the first one to install a fully automated scanner tunnel (FAST) in July 2010. The system is equipped with multiple image scanners that not only identify barcodes but also analyze shapes, sizes, and colors of the package by using image recognition. Non-barcode items such as fruit and vegetable, need to be labeled in self-service before going through the conveyer belt. [9]

III. Proposed System

3.1 YOLO image detection algorithm

The image that is taken from the camera will be provided as an input to the YOLO algorithm. This algorithm will then make use of trained datasets in order to recognize the different objects that are present in the system. The output of this algorithm will be an image in which all the different objects present within it are labeled with their name and are within a colored boundary specifying the exact location of the object within the image.

3.2BBox label tool

We will use this tool in order to train the machine for recognizing the objects that are present within the images given as input to the YOLO. The bounding boxes have to be done manually for a certain no of images before the YOLO becomes capable of recognizing the new images given to it.

3.3Image datasets

In this stage, the datasets which are required in order to train the machine to recognize a specific object are either created or accessed from open source libraries. We can use one of the many available open source datasets libraries such as OpenCV or ImageNet.

3.4Database System

A database system will be required in order to store various attributes of the objects that the customers can buy within the store such as price, name, quantity, company name etc. We can use one of many databases such as MySQL Microsoft access or Oracle 9i.

The proposed system will use all the above-mentioned components in order to help the customers to self-checkout without needing to scan the barcode.

IV. Architecture

YOLO (You Only Look Once)

YOLO is a neural network for doing object detection in real-time. You can take a classifier like VGGNet or Inception and turn it into an object detector by sliding a small window across the image. At each step, you run the classifier to get a prediction of what sort of object is inside the current window. Using a sliding window gives several hundred or thousand predictions for that image, but you only keep the ones the classifier is the most certain about. This approach works but it's obviously going to be very slow since you need to run the classifier many times. A slightly more efficient approach is to first predict which parts of the image contain interesting information so called region proposals and then run the classifier only on these regions. The classifier has to do less work than with the sliding windows but still gets run many times over. [7]

YOLO takes a completely different approach. It's not a traditional classifier that is repurposed to be an object detector. YOLO actually looks at the image just once (hence its name: You Only Look Once) but in a clever way. YOLO divides up the image into a grid of 13 by 13 cells:

Fig. 1 *TEST IMAGE 1*

Each of these cells is responsible for predicting 5 bounding boxes. A bounding box describes the rectangle that encloses an object. YOLO also outputs a confidence score that tells us how certain it is that the predicted bounding box actually encloses some object. This score doesn't say anything about what kind of object is in the box, just if the shape of the box is any good. The predicted bounding boxes may look something like the following (the higher the confidence score, the fatter the box is drawn):

Fig 2 *TEST IMAGE 2*

For each bounding box, the cell also predicts a class. This works just like a classifier: it gives a probability distribution over all the possible classes such as bicycle, boat, car, cat, dog, person, and so on. The confidence score for the bounding box and the class prediction are combined into one final score that tells us the probability that this bounding box contains a specific type of object. For example, the big fat yellow box on the left is 85% sure it contains the object "dog"

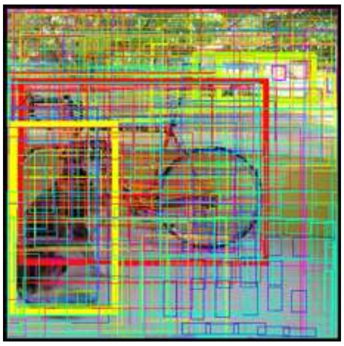

Fig. 3 *TEST IMAGE 3*

Since there are 13×13 = 169 grid cells and each cell predicts 5 bounding boxes, we end up with 845 bounding boxes in total. It turns out that most of these boxes will have very low confidence scores, so we only keep the boxes whose final score is 30% or more (you can change this threshold depending on how accurate you want the detector to be).The final prediction is then:

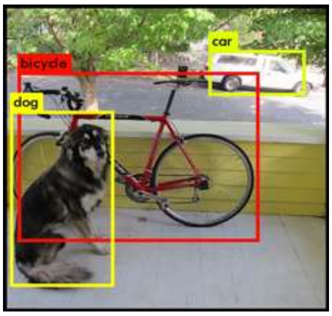

Fig. 4 *TEST IMAGE 4*

From the 845 total bounding boxes, we only kept these three because they gave the best results. But note that even though there were 845 separate predictions, they were all made at the same time the neural network just ran once. And that's why YOLO is so powerful and fast.

V. Implementation

For the detection of all the products, we'll use the YOLO model. We can either feed it a live stream of the products placed under the camera or simply a photo. YOLO will then generate the bounding boxes around the objects detected. For demonstration purposes here's an example where the image is passed to the model through the terminal but it can be easily automated using PHP and a bash script. This will allow the user to use the web browser as an interface by simply clicking on a button the predictions.png file will be created in the working directory.

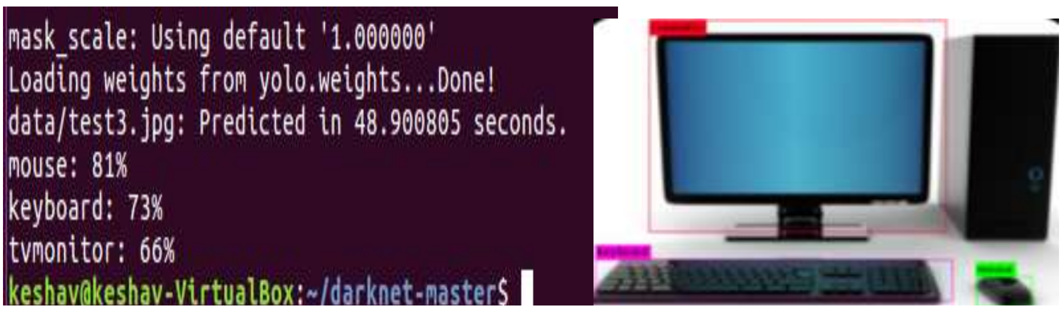

Fig. 5 *Products Detection multiple items*

The prediction file can be generated either as a .png or a .txt file. We'll use the .txt file so as to create a database of the items detected. Once all this is done we can query the existing database which contains the names and prices of all the products that are available to generate the bill.

VI. Conclusion

Compared with the current system, our system replaces the RFID and barcodes during checkout and scanning all the products in one go thereby reducing the checkout time. In this paper, I have explored redesigning a self-checkout system with a universal design, and I believe my design addresses the most egregious anthropometric human factors issues present in current designs. However, a major tradeoff of my design (and most likely of any design which decreases functional reach requirements for some of its components).

Although it's being implemented in more and more stores, the system comes with certain security risks and reliability issues. Due to its speedy performance, respecting customers' privacy and giving them control it is sure to increase the profit of the retail shop.

Whether or not self-service checkout is right for business depends on your customer base and business model, but be aware this disruptive technology enables a level of automation that you'll need to compete with.

References

[1] Appearance-based Image recognition-https://classes.soe.ucsc.edu/cmpe264/Fall06/Lec20.pdf.

[2] Factors affecting the adoption of self-service checkout terminals in Slovenia." Information and Communication Technology, Electronics and Microelectronics(MIPRO), 2014 37th International Convention on. IEEE,2014.

[3] Feature-based Image recognition-http://www.massey.ac.nz/~rcflemme/feature-based%20object%20recognition.pdf.

[4] How to train YOLO-https://timebutt.github.io/static/how-to-train-yolov2-to-detect-custom-objects.

[5] Khurana P, Sharma A, Singh SN, Singh PK. A survey on object recognition and segmentation techniques. In Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on 2016 Mar 16 (pp. 3822-3826) IEEE.

[6] Man KF, Tang KS, Kwong S. Genetic algorithms: concepts and applications [in engineering design]. IEEE transactions on Industrial Electronics. *1996 Oct;43(5):519-34*.

[7] 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 *2016 (pp. 779-788*).

[8] Rossetti MD, Pham AT. Simulation modeling of customer checkout configurations. In Proceedings of the 2015 Winter Simulation Conference 2015 *Dec 6 (pp. 1151-1162). IEEE Press*.

[9] Wu BF, Tseng WJ, Chen YS, Yao SJ, Chang PJ. An intelligent self-checkout system for smart retail. In System Science and Engineering (ICSSE), *2016 International Conference on 2016 Jul 7(pp. 1-4). IEEE.*