Smart Self-Checkout Carts Based on Deep Learning for Shopping Activity Recognition

Hong-Chuan Chi†
Muhammad Atif Sarwar† Yousef-Awwad Daraghmi‡ Kuan-Wen Liu† Tsì-Uí İk†* Yih-Lang Li†
Department of Computer Science, College of Computer Science
National Chiao Tung University
1001 University Road, Hsinchu City 30010, Taiwan
*Email: cwyi@nctu.edu.tw

***Abstract*—Fast and reliable communication plays a major role in the success of smart shopping applications. In a "Just Walk Out" shopping scenario, a video camera is installed on the cart to monitor shopping activities and transmit images to the cloud for processing so that items in the cart can be tracked and checked out. This paper proposes a prototype of a smart shopping cart based on image-based action recognition. Firstly, deep learning networks such as Faster R-CNN, YOLOv2, and YOLOv2-Tiny are utilized to analyze the content of each video frame. Frames are classified into three classes: No Hand, Empty Hand, and Holding Items. The classification accuracy based on Faster R-CNN, YOLOv2, or YOLOv2-Tiny is between 93.0% and 90.3%, and the processing speed of the three networks can be up to 5 fps, 39 fps, and 50 fps, respectively. Secondly, based on the sequence of frame classes, the timeline is divided into No Hand intervals, Empty Hand intervals, and Holding Items intervals. The accuracy of action recognition is 96%, and the time error is 0.119s on average. Finally, we categorize the events into four cases: No Change, placing, Removing, and Swapping. Even including the correctness of the item recognition, the accuracy of shopping event detection is 97.9%, which is higher than the minimal requirement to deploy such a system in a smart shopping environment. A demo of the system and a link to download the data set used in the paper are in Smart Shopping Cart Prototype or found at this URL: https://hackmd.io/abEiC83rQoqxz7zpL4Kh2w.**

***Index Terms*—Smart shopping cart, frame classification, action recognition, Faster R-CNN, YOLOv2, YOLOv2-Tiny.**

I. Introduction

Future networks assist in developing smart shopping systems and the "Just Walk Out" concept in the retail business for speeding up the checkout process and increasing customer convenience. For example, Amazon released Amazon Go's intelligent concept store in Seattle at the end of 2016 [1], and Alibaba Group opened its intelligent store "Taocafe" in Hangzhou city in July 2017 [2]. Customers only need to be authenticated through their smartphones at the entrance and then no explicit checkout procedures are needed before leaving. To realize "Just Walk Out" shopping scenarios, besides authentication and online payment, tracking of items in shopping lists is a major issue that requires integrating advanced technologies, such as computer vision, deep learning, and action recognition.

Recent smart checkout solutions included facial recognition and image-based identification for tracking customers by cameras installed in stores [3]. Also, image-based action recognition was developed to detect taking and placing items from shelves into shopping carts [4]. These services are combined with RFID sensors so that purchased items can be identified [5]. However, these solutions have drawbacks as images can be unclear, or targets are obstructed by objects causing inaccurate item recognition or event detection. Also, adding sensors to retails, such as RFIDs, is expensive. Alternatively, tracking items placed in shopping carts by cameras mounted on the carts would be more practical [2], [6].

This paper proposes an accurate and efficient smart checkout system. A camera is installed on the shopping cart to monitor shopping activities within the cart. The video is uploaded to the cloud, and deep learning networks are used to analyze the context of each frame. The frames are classified into three classes depending on whether hands and items can be found in the image or not. If there is no hand, the frame will be classified as a No Hand (NH). While if a hand is found, the frame will be classified as either Empty Hand (EH) or Holding Item (HI) classes. Then, the timeline is segmented into three types of shopping action intervals: no-hand intervals, empty-hand intervals, and holding-item intervals. Then, a shopping event is defined as No Change, Placing, Removing, and Swapping.

To increase the accuracy, the Dynamic Time Warping (DTW) is used for matching the shopping action series between ground truth and prediction. Without considering the type of items, the accuracy of shopping action recognition is 96%, and the average time error is 0.119 seconds. For shopping event detection, if DTW is applied to compare the shopping event series between ground truth and prediction, the accuracy can reach up to 97.9%. The proposed approach is compared with the Faster Region Convolution Neural Network (Faster R-CNN), You Only Look Once (YOLOv2), and YOLOv2-Tiny. In total, ten different items are used for performance evaluation. The results show that the proposed system outperforms the other methods.

† EECS International Graduate, National Chiao Tung University, 1001 University Road, Hsinchu City 30010, Taiwan
‡ Depratment of Computer Systems Engineering, Palestine Technical University - Kadoorie, Tulkarem, Palestine

II. Related Work

The application level of the related work includes several systems from the retail business. The Amazon Go stores and the Taocafe are two famous systems that focus on smart check-out. The Amazon Go stores have shown the significance of linking "shelf sensing" and "computer vision" simultaneously to track customer's activities and detect selected items from the shelf [7]. Load cells can sense removing and placing events and recognize which items are involved in the events. If customers pick or place items at similar places or replace items of similar weights, mistakes might occur. If several load cells are installed under the shelf platform, the position of the picked up or placed item can be estimated based on the change in weights of the load cells. The Taocafe revealed that techniques including sensors, RFID, and image processing are integrated to develop their smart shopping environment [3]. Other smart self-checkout systems and technologies can be found in [8].

The methodological level of the related work includes the methods used for objects and event recognition. Traditional image recognition often utilizes the histogram of oriented gradients (HOG) [9], scale-invariant feature transform (SIFT) [10], speeded up robust features (SURF) [11] and other statistic functions to determine local image features. But, there are a few drawbacks such as image quality, camera angle, and object size. The sliding window is a widely adopted technique to search and detect target objects over the image. The performance often depends on whether the selected parameters are appropriate or not. Recently, based on the advancement of GPU technology, deep learning networks that structurally extract features and detect objects have become the mainstream of image processing [12]. Other studies have focused on spatial and temporal action segmentation, such as Action Tube [13], which can be used to analyze customer's shopping activities.

Further, the R-CNN family, including R-CNN [14], Fast R-CNN [15], and Faster R-CNN [16], is a representation of multi-stage object detection. The searching space is scanned to select candidate regions, and individual inspection is executed on each candidate region. Although this approach can achieve high accuracy, computation speed is a concern. Faster R-CNN is the widely used design among the R-CNN family. It adopts CNN to calculate a feature map. Then, RPN (Region Proposal Network) that has the properties of sliding windows [17] is applied to find ROIs (Region of Interest). For each RoI, its corresponding feature map block will be used to calculate the bounding box and class of the object on the RoI.

Responding to the need for real-time applications, the YOLO, YOLOv2, and YOLOv3 are an end-to-end design that achieves high processing speed [18]–[20]. YOLO splits the input image into $S \times S$ grid cells and extracts feature using the CNN. It predicts the boundary boxes of objects and each boundary box has conditional probability and confidence score for classes. The confidence score tells that the boundary box has some objects, but this score does not tell what kind of objects is in the box. Each boundary box also predicts the class. It works like a classifier that determines a probability distribution over all the classes. The class probability and confidence score are combined into one score that shows the specific object contained the boundary box.

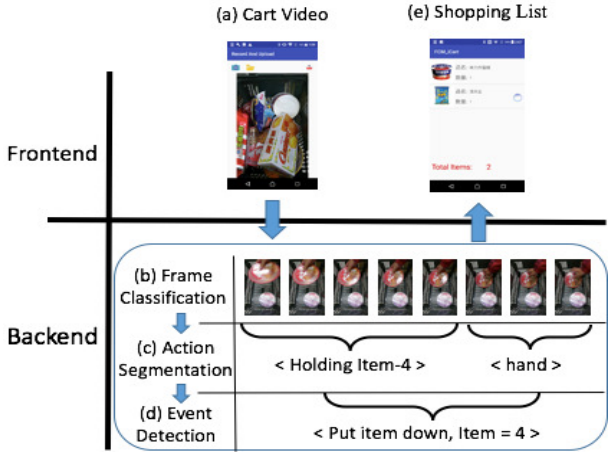

Fig. 1. iCart System Architecture

In conclusion, the accuracy and efficiency of object recognition still require improvement to enable real-time smart and Just Walk Out shopping.

III. System Architecture and Prototype Design

The system architecture consists of five main modules that are: the cart video module Fig. 1(a), frame classification Fig. 1(b), shopping action recognition Fig. 1(c), shopping event detection Fig. 1(d), and virtual shopping cart module Fig. 1(e). In this architecture, Fig. 1(a) and 1(e) are front-end user app interfaces. They are developed on Android Studio for Android platform. While, Fig. 1(b), 1(c), and 1(d) are back-end services developed on Linux platform using Python and C++. A NVIDIA 1080Ti GPU is used to execute deep learning networks, and the FFmpeg package is for the basic video processing, e.g. frame capture.

A. Cart Video Module

The cart video module monitors and records shopping activities of the cart. A Customer starts a video of shopping activities, particularly items added or removed from the cart, once enters the store using the app installed on the smartphone. The smartphone supports the cart video module. After finishing the shopping, the Customer transfers the videos to back-end servers for further processing.

B. Frame Classification Module

This classifies frames by detecting the object of interest in the images via deep learning algorithms. According to the presence or absence of either an empty hand image or a hand holding item image, frames are classified into No Hand (NH), Empty Hand (EH), or Holding Item (HI). For example, Fig. 2(a) is a No hand case in which no objects are labeled, Fig. 2(b) is an Empty Hand case, and Fig. 2(c) is a Holding Item

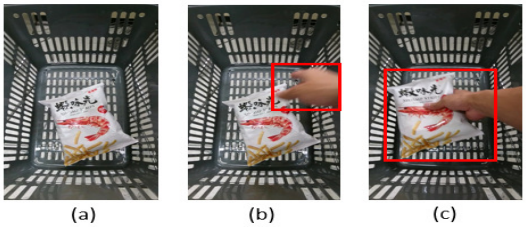

Fig. 2. Object labeling and Frame Classification: (a) No-Hand, (b) Empty-Hand, (c) Holding-Item

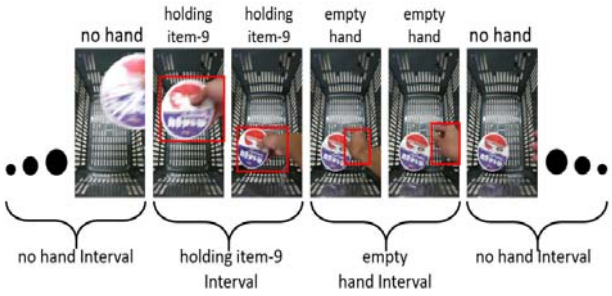

Fig. 3. Shopping Action Decomposition

case. The HI can be subdivided into 10 sub-classes: HI-1, HI-2, ..., HI-10 if the item in hand is considered. To label objects, the existence of a hand image is determined by the appearance of part of the palm. In case fingers are shown but not the palm, it is considered as a NH case. In the testing phase, if more than one object is detected, the object with the highest confidence score is used to classify the frame. In the end, frames are classified into 12 classes.

C. Shopping Action Recognition Module

The shopping action is used to analyze the process in which the customer places the item into the shopping cart or take it out. An action refers to the same state that lasts for a period of time. The action recognition module partitions the timeline into action intervals by aggregating consecutive frames with the same frame class. The corresponding action intervals of NH, EH, and HI frame classes are called No-Hand periods, Empty-Hand periods, and Holding-Item periods. Each action period or interval is a shopping action. In addition to the class attribute, the item name, start time, and end time are also important parameters in action recognition. A shopping action analysis is based on the time sequence of the frame classes, and each action maps to an exclusive time interval with a NH, EH, or HI label as shown in Figure 3. After an action is determined, the list of items in the virtual shopping cart can be added or deleted according to the shopping event detection module.

Misclassification of frames may occur, and this cause misses in action recognition and event detection. Based on the observation that each action should last for a period of time, we conjecture that smoothing on the time series of frame classes which can assist in correcting frame misclassification. Simultaneously, smoothing removes fragmented intervals caused by instances of misclassification. For each frame, the smoothing algorithm refers to k frames in front and k frames behind, and in total $2k + 1$ frames including the frame itself do a major vote to decide the class of the frame. In the final performance evaluation, $k = 2$ was chosen. The major vote based smoothing algorithm is described below.

Smoothing Algorithm for Frame Classification

```
Stage 1: Consider all HI sub-classes as the same class.
  Mark all HI sub-classes as one HI class (o);
if a single major class exists then
    if the major class is HI then
        goto Stage 2;
    else
        mark the frame as the major class;
        return;
    end
else
    mark the frame as the noise (n);
    return;
end

Stage 2: HI is the major class.
if a single major sub-class exist then
    mark the frame as the major sub-class;
    return;
else
    goto Stage 3;
end

Stage 3: Resolve the rest (o) mark
count frames in front or behind having the same class;
if one is more than the other then
    mark the frame as the class with larger group;
    return;
else
    mark the frame as the class in front;
    return;
end
```

Fig. 4 shows an example of frames classification. The first row, labeled by Step 1, is the series of original frame classification. In the second row, labeled by Step2, all holding item sub-classes are considered as the same class and denoted as 'o'. The third row, labeled by Step 3, is the outcome of Stage 2. The last row, labeled by Result, is the final outcome.

D. Shopping Event Detection Module

The shopping event refers to the change made by a customer on the item within the time interval between the appearance and disappearance of hand images, including no change, placing, removing, and swapping. The first action after the NH period and the last action before the NH incorporating the

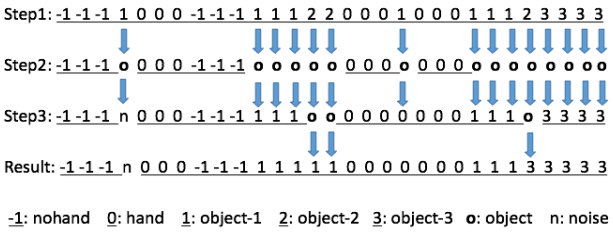

Fig. 4. Smoothing of frame classification series

items involved in the HI periods are key information to decide what kind of the event and which item is removed and/or placed. "No change" means that the items within the cart are not changed. The possible situations are (1) empty-hand-in and empty-hand-out, and (2) holding-item-in and holding-item-out with the same item. "Placing" means an item is placed into the cart when holding-item-in and empty-hand-out. "Removing" means an item is removed from the cart when empty-hand-in and holding-item-out. "Swapping" refers to placing an item in and then removing another item out when holding-item-in and holding-item-out but with a different item. By combining the products involved in the action, the items related to a shopping event are known, and the items list can be maintained. For example, if an event is to place item 'A' into the cart, the amount of item 'A' in the shopping list will be increased by one.

In this work, the primary task is to develop the capability of action analysis, and item recognition is a secondary issue. So, item recognition is simply implemented in the frame recognition module. We assume each shopping action should span at least two frames. The smoothing process will reduce the occurrence of action of fewer than 2 frames. If the frame rate is 30 fps, in terms of time, 0.07 sec is the minimum duration of action.

E. Virtual Shopping Cart Module

Finally, the virtual shopping cart module calculates the item list according to the detected shopping events. The item list is displayed on the smartphone app via the Google FCM service for reference. The virtual shopping cart module displayed the items as it received the tokens from the server.

IV. Performance Evaluation

A. Data Set

The data set contains 36 videos. The possible scenarios include placing, picking, viewing, and replacing items within the shopping cart. The videos were recorded by a Sony F5121 Android phone with a resolution of 756×1344 at a frame rate of 30 fps. The total length of the videos is about 78 minutes. The data set is separated into three folders: "data_0", "data_1", and "data_2". "data_0" consists of 28 fully labeled videos. "data_1" has 2 partially labeled videos which are mainly used to supplement "data_0" such that every class has roughly the same number of frames. "data_2" contains 6 videos with only labelled files of shopping actions and events for the final performance evaluation. In these three folders, each video along with its label files is stored in a sub-folder named video0, video1, etc. The video files are named video0.mp4, video1.mp4, etc. The corresponding frame label file, shopping action file, and shopping event file of video0.mp4 considered as an example are named as video0.csv, video0_action.csv, and video0_event.csv, respectively. The image files are sequentially named as 000001.jpg, 000002.jpg, etc. The labeled images are stored in a folder named "jpg". In total, there are $81,374$ frames labeled. The unlabeled images are stored in a folder named "nolabel".

In the frame label files, each line represents a labeled frame with at most one-labeled object. The attributes including "file name", "class", "x coordinate (xmin)" and "y coordinate (ymin)" are in the upper left corner of the object, x coordinate (xmax) and y coordinate (ymax) are in the lower right corner of the object. "file name" and "class" are the file names of the labelled frame and the class of the frame. If an object is labelled in the frame, "xmin" and "ymin" are the coordinates in pixel of the upper left corner of the labelled object and "xmax" and "ymax" are the coordinates of the lower right corner of the object. If frames belong to the NH class, no object will be labelled, and "xmin", "ymin", "xmax", and "ymax" will be blank.

In the shopping action labeled files, each line annotates one action interval which composes the start frame (start), the end frame (end), and the action class (action). For the HI intervals, the item information is also recorded. The shopping event label file records changes of items in the shopping cart between two NH periods. The attributes include the start frame, end frame, event class, placed item and picked item of the event.

B. Evaluation of Frame Classification

In this section, the pros and cons of adopting Faster $R-CNN$, $YOLOv2$, and $YOLOv2-Tiny$ for frame classification are evaluated. The 3-fold cross-validation is applied for this purpose. In the evaluation, about 38-minute of video with 68,754 frames are used. The video was divided into three sets of equal size. To keep the integrity of each shopping event, the cutting points are located in the NH interval. In the training phase, the number of NH and EH frames are 43084 and 9005, respectively; and the number of HI-1, HI-2, ..., HI-10 frames are 1669, 1,635, 1,712, 1,580, 1,573, 1,736, 1615, 1688, 1756, and 1521, respectively. The accuracy of the 3-class models of Faster R-CNN, YOLOv2, and YOLOv2-Tiny is 93.0%, 91.3%, and 90.3%, respectively. The accuracy of the 12-class model for the three networks is 92.0%, 90.8%, and 88.7%, respectively. The accuracy of the fused 3-class models is 92.2%, 91.3%, and 89.4%, respectively. The accuracy is calculated as:

$$Accuracy = \frac{\text{\# of hits}}{\text{\# of frames}} \tag{1}$$

The average processing speed of YOLOv2, YOLOv2-Tiny, and Faster R-CNN, running on PCs are equipped with an Intel i5-7500 CPU, an NVIDIA GTX-1080 Ti GPU, and 32GB RAM are 5 fps, 39 fps, and 50 fps, respectively. Although the

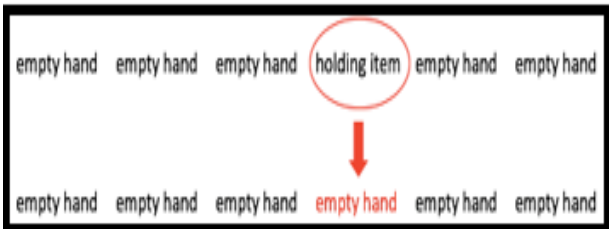

Fig. 5. Smoothing can correct the misjudgment of frame classes

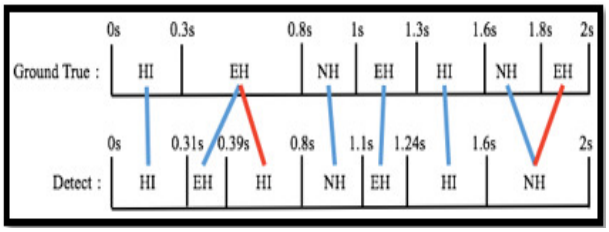

Fig. 6. Perform alignment of action time series using DTW

TABLE I
FRAME CLASSIFICATION ACCURACY AFTER SMOOTHING

Model	# of Classes	Single Frame	Smoothing (k=2)	Smoothing (k=3)
Faster R-CNN	3-class	0.930	0.930	0.929
	12-class	0.920	0.920	0.919
	fused 3-class	0.922	0.922	0.921
YOLOv2	3-class	0.913	0.915	0.914
	12-class	0.908	0.911	0.909
	fused 3-class	0.913	0.915	0.914
YOLOv2 Tiny	3-class	0.903	0.905	0.905
	12-class	0.887	0.891	0.890
	fused 3-class	0.894	0.896	0.894

3-class model of Faster R-CNN has the highest accuracy, we chose YOLOv2 due to the consideration of the speed factor. The 12-class model of YOLOv2 is going to be adopted in the following performance analysis since it can provide the classification of items.

C. Improvement by Smoothing

Frame classification has a certain miss rate. Based on the observation of frame classification, action should last for a reasonable period. Smoothing removes the miss rate among the sequence of hits of the same categories in a significant way. This assists in identifying the direction of the beginning and end of an action interval. Also, smoothing improves the correctness of the subsequent shopping action bias. An example is shown in Fig. 5 where the sequence is a part of an EH interval. If there is a classification miss rate, the EH interval will be separated into two.

To evaluate the effect of smoothing, the accuracy of every combination of networks and models which cooperates with various smoothing parameters is given in Table I. In this table, a "single frame" is meant to be without smoothing. So, comparing to the accuracy rate of "single frame", smoothing improves the accuracy of frame classification. After smoothing, the ranking of three algorithms in terms of accuracy is similar to the original one.

The effectiveness of smoothing with $k = 2$ is better than $k = 3$. Therefore, the 12-class YOLOv2 model with smoothing parameter $k = 2$ should be applied in subsequent tests of action recognition and event detection.

D. Evaluation of Shopping Action Recognition

To evaluate the action recognition, the Dynamic Time Warping (DTW) algorithm [21] is adopted to compare the two action time series of ground truth and prediction. Since the correspondence may be one-to-many or many-to-one in the evaluation, the pairs of same action classes are considered as hits, and different action classes are considered as miss rate. Fig. 6 shows an example. This line indicates the pairing result of DTW. The blue lines denote accurate pairing, and the red lines denote inaccurate pairing.

The performance of shopping action recognition will be evaluated by two indices, including the ratio of correct pairs, and the average time error. The ratio of action pairs, denoted by ActAcc, is given by

$$ActAcc = \frac{\text{\# of Correct Action Pairs}}{\text{Total \# of Action Pairs}} \tag{2}$$

The average time error, denoted as $TimeErr$, only considers the correct action pairs. The average time error on the start time and end time of every correct action pairs is calculated. In terms of the number of pairs, there are six accurate action pairs, and two inaccurate action pairs, Fig. 6. So, the $ActAcc = 6/8 = 0.75$. The start time difference of the six accurate action pairs is 0s, 0.01s, 0.1s, 0s, 0.06s, and 0s, and the end time difference is 0.01s, 0.41s, 0.1s, 0.06s, 0s, and 0.2s. So, $TimeErr = 0.079s$.

For evaluating the performance, we recorded another six videos. The total length of the videos is about six minutes. The total number of shopping actions is 395, including 121 NH action periods, 139 EH action periods, and 10 HI-1, 11 HI-2, 14 HI-3, 15 HI-4, 15 HI-5, 13 HI-6, 11 HI-7, 18 HI-8, 16 HI-9, and 11 HI-10 action periods. There were 117 shopping events, including 10 no-change, 59 placings, 44 removings, and 4 swappings. The video is placed in the data_2 folder, and the ground truth of the shopping action is stored in the video_action.csv file. Table II is the result of the above-mentioned action evaluation of before and after smoothing. $ActAcc$ is the ratio of correct action pairs, and $TimeErr$ is time error.

E. Evaluation of Shopping Event Detection

Similar to the evaluation of shopping action recognition, DTW is adopted to compare the two events time series of ground truth and prediction for the evaluation of shopping event detection. Since the shopping events are used for tracking the list of items in carts, both the event class and the item type should be validated. In other words, the accurate event pair refers to both the event class and the item type to be

TABLE II
ACCURACY OF ACTION RECOGNITION

Video	Before Smoothing		After Smoothing	
	$ActAcc$	$TimeErr$	$ActAcc$	$TimeErr$
test001	0.914	0.348	0.972	0.195
test002	0.779	0.312	0.927	0.134
test003	0.858	0.124	0.981	0.069
test004	0.830	0.275	0.974	0.083
test005	0.893	0.263	0.958	0.076
test006	0.832	0.274	0.949	0.158
Average	**0.851**	**0.266**	**0.960**	**0.119**

TABLE III
ACCURACY OF SHOPPING EVENT DETECTION

Video	Before Smoothing	After Smoothing
test001	1.000	1.000
test002	1.000	1.000
test003	0.848	0.970
test004	0.524	0.905
test005	1.000	1.000
test006	0.810	1.000
Average	**0.864**	**0.979**

consistent. Table III shows the accuracy of event detection. The accuracy of event detection before and after smoothing is 86.4% and 97.9%, respectively. Thus, smoothing is a key technique to improve the accuracy of shopping event detection.

In the application of smart stores, the list of purchased items is significant information. Each mistake can have a negative impact on the customer's shopping experience and also affect the store's income. 95% accuracy rate is the benchmark value of such systems. Finally, if a picked item cannot be found in the item list of the shopping cart, it implies some errors and the customer can ask for the assistance.

V. CONCLUSION

This paper proposes a smart and accurate shopping cart system equipped with a camera to monitor the shopping activities in the cart. The deep learning networks are adopted to classify frames according to object of interest in each frame. The interval of each shopping action is decided by aggregating consecutive frames of the same class. Finally, shopping events that are placing, removing or swapping are of items in the shopping cart are identified so that the shopping list is updated. The implemented prototype shows that efficient deep learning methods, cloud computing and fast network system are key elements in developing successful smart shopping. In future, more sophisticated approaches can be introduced to further improve the accuracy like Hidden Markov Models.

ACKNOWLEDGEMENT

This work of T.-U. İk was supported in part by the Ministry of Science and Technology, Taiwan under grant MOST 108-2627-H-009-001. This work was financially supported by the Center for Open Intelligent Connectivity from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan.

REFERENCES

[1] Amazon, "Amazon go," https://www.amazon.com, 2016, [Online; accessed 23-January-2017].

[2] H. Yourong, "Early amazon step! alibaba announced that no one retail store is coming," https://tw.news.yahoo.com/, 2017, [Online; accessed 3 July 2017].

[3] R. Bobbit, J. Connell, N. Haas, C. Otto, S. Pankanti, and J. Payne, "Visual item verification for fraud prevention in retail self-checkout," in *2011 IEEE Workshop on Applications of Computer Vision (WACV)*. IEEE, 2011, pp. 585–590.

[4] 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.

[5] Y.-C. Wang and C.-C. Yang, "3s-cart: a lightweight, interactive sensor-based cart for smart shopping in supermarkets," *IEEE Sensors Journal*, vol. 16, no. 17, pp. 6774–6781, 2016.

[6] M. A. Sarwar, Y.-A. Daraghmi, K.-W. Liu, H.-C. Chi, T.-U. İk, and Y.-L. Li, "Smart shopping carts based on mobile computing and deep learning cloud services," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2020, pp. 1–6.

[7] K. Wankhede, B. Wukkadada, and V. Nadar, "Just walk-out technology and its challenges: A case of amazon go," in *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*. IEEE, 2018, pp. 254–257.

[8] B. Nallapureddy, P. Das, N. Nagaraj, S. Parameswaran, J. Zanin-novich, and P. S, "Future of self checkout a landscape study," https://scet.berkeley.edu/wp-content/uploads/Future.pdf, [Online; accessed 16 November 2019].

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[10] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

[11] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[12] Y. Gu, X. Ye, W. Sheng, Y. Ou, and Y. Li, "Multiple stream deep learning model for human action recognition," *Image and Vision Computing*, vol. 93, p. 103818, 2020.

[13] V. Kalogeiton, P. Weinzaepfel, V. Ferrari, and C. Schmid, "Action tubelet detector for spatio-temporal action localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 4415–4423.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 1, pp. 142–158, 2016.

[15] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[16] 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.

[17] R.-C. Chen *et al.*, "Automatic license plate recognition via sliding-window darknet-yolo deep learning," *Image and Vision Computing*, vol. 87, pp. 47–56, 2019.

[18] 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.

[19] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint*, 2017.

[20] ——, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[21] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series." in *KDD workshop*, vol. 10. Seattle, WA, 1994, pp. 359–370.