

Cashierless Checkout Vision System for Smart Retail using Deep Learning

Ren-Yi Lee¹, Tong-Yuen Chai², Sing-Yee Chua¹, Yen-Lung Lai¹, Sim Yee
Wai², Su-Cheng Haw³

¹ Lee Kong Chian Faculty of Engineering and Science, Universiti Tunku Abdul
Rahman, 43000 Kajang, Selangor, Malaysia

² Faculty of Computing and Engineering, Quest International University, 31350
Ipoh, Perak, Malaysia

² Faculty of Computing and Informatics, Multimedia University, 63100 Cyberjaya,
Selangor, Malaysia

chaitongyuen@gmail.com (corresponding author)

Abstract. As Corona Virus Disease (COVID-19) pandemic strikes the world, retail industry has been severely impacted by staff shortage and high risk of virus outbreak. However, most of existing smart retail solutions is associated with high deployment and maintenance cost that are infeasible for small retail stores. As an effort to mitigate the issue, a computer vision-powered smart cashierless checkout system is proposed based on You Only Look Once (YOLO) v5 and MobileNet V3 for product recognition along with 3-stage image synthesis framework that includes crop and paste algorithm, GAN-based shadow synthesis and light variation algorithm. By using 3000 images generated from the framework, proposed model was trained and optimized with TensorRT. Experimental result shows that the lightweight model can be deployed on affordable edge devices like Jetson Nano while achieving high Mean Average Precision (mAP) of 98.2%, Checkout Accuracy (cAcc) of 89.17% with only 0.142s of inference time.

Keywords: computer vision, object detection, deep learning, retail stores, smart retail.

1. Introduction

As the last stage of goods distribution channel, retail can be defined as commercial activities that involve direct selling of merchandise to consumers at a specific point of purchase (Bankim, 2015). Among the merchandise available in the market today, most of them are generally categorized under Fast-Moving Consumer Goods (FMCG) which possess several characteristics such as high consumer demand, common availability and associated with wide variations. With that, retail stores will require high capacity of manpower especially in checkout process to accommodate high requirement of FMCG products. However, COVID-19 pandemic has caused most of the retail stores to suffer from concern of staff shortages according to Kumar et al. (2020). Concurrently, traditional face-to-face checkout process is associated with high risk of virus spread chain because close contact within 1 meter will inflict higher risk of being infected by COVID-19 virus as stated in Ministry of Health Malaysia (2020).

There are multiple solutions available in the market to help retail stores in shifting their operation to cashierless and contactless operation. For instance, Regi-Robo™ by Panasonic (2018) has adopted Radio Frequency Identification (RFID) in their checkout process where all products in store will be labelled with unique RFID tag. Thus, customers can simply pick their desired product and placed them in the basket before proceeding to a special checkout counter that is equipped with RFID reader. The reader will emit signals to the RFID tags. Once the tags are activated, they will send their embedded information in wave form to the reader for interpretation, allowing an automated checkout process. However, this implementation may incur additional costs since every product needs to be manually labelled with RFID tags by employee. Additionally, according to Periyasamy & Dhanasekaran (2014), RFID technology has degraded performance when dealing with metal or liquids, which can be unsuitable for FMCG.

Besides, Amazon Go by Amazon.Com, (n.d.) has combined deep learning, computer vision and different types of sensors in their cashierless store. In the store, users can pick their desired item from the shelf and their action will be tracked by the cameras mounted in store. Multiple sensors placed on the shelf will also be used to increase reliability of product recognition. This implementation simplifies the overall shopping experience since user can just enter the store by scanning their Quick Response (QR) code at entrance, grab in-store products and leave the store directly. However, as stated in Polacco & Backes (2018), the technology can only handle low capacity of customer and may fail in recognizing items with similar shape. Additionally, the implementation may require complete overhaul of store, which can be unfeasible for small-scale retail stores.

Thus, this paper proposes a software prototype of cashierless checkout intelligent vision system that can be deployed on low-cost edge devices. It utilizes state-of-the-

art deep learning models at its core so that small-scale retail can easily install it at their checkout counters while keeping the implementation cost at its minimal. Our contribution also includes a novel 3-stage image synthesis framework to effectively simulate the actual checkout scenarios with lesser human intervention.

The remainder of the paper is organized to several sections. Section 2 describes the techniques and works related to product recognition. Section 3 and 4 will introduce the methods for software prototype development and experimental results. Lastly, Section 5 will include conclusions and acknowledgements.

2. Related Works

2.1. Retail product datasets

In the field of product recognition, several datasets were released publicly and two of them are commonly adopted in recent research in the field. One of datasets is known as MVTec D2S Dataset by Follmann et al. (2018) that was prepared to address multiple actual checkout counter scenarios such as lighting variation, product occlusion, and intraclass variation. The dataset can be found in Ning et al. (2019) and Bi & Wang (2021). As shown in Figure 1, It contains 60 classes of Germany groceries and made up of 14380 training and 13020 validation images that were captured in a resolution of 1920 x 1440. Each of the images was annotated with bounding boxes and class name in COCO JSON format.



Fig. 1: Train and validation split from D2S dataset.

Meanwhile, another dataset known as RPC Dataset introduced by Wei et al. (2019) is also widely adopted in recent works such as C. Li et al. (2019) and Xiao et al. (2020) as it represents a large-scale dataset with 200 classes of China groceries, loaded with 53739 training and 30000 validation images that were captured at a resolution of 1592 x 1440 and 1800 x 1800 respectively with their annotations in COCO JSON format. The training images were designed to ease image segmentation and synthesis while

the test images can be further split into 3 clutter levels. However, lighting variation is not considered in the dataset as all the test images were captured under constant lighting. Some samples from the dataset are shown in Figure 2.



Fig. 2: Train and test split from RPC dataset.

2.2. Image augmentation

Generally, vast amount of training samples will be required to achieve a well-performed deep learning model. However, F.-F. Li et al. (2006) stated that large dataset can be hard to acquire. Thus, image augmentation provides an inexpensive approach in expanding the dataset diversity through generation of new images by introducing operations to the existing images. Several image augmentation techniques that were used in product recognition encompassed mask-based synthesis, GAN rendering and conventional method.

Conventional data augmentation will typically involve fundamental manipulation of images such as geometric transformation, flipping and colour space transformation, cropping, rotation, translation, and noise injection. Such approach can be seen in works by Rathnayake & Nawarathna (2020) and Rigner (2019).

As for mask-based synthesis, the algorithm will commonly involve extraction and segmentation of object mask and apply different operations based on the extracted mask. In Yi et al. (2019), each product was cropped based on the bounding box generated through Selective Search. The cropped products were then used as small patches to cover other products to simulate product occlusion. Additionally, in Koturwar et al. (2019), product mask was extracted by calculating pixel-wise standard deviation across Red, Green and Blue (RGB) colored background. Accurate mask can be extracted since products will have lower standard deviation because their pixel values will not be affected by changes of background colour. After that, the

images were crop and paste on an empty background in random orientation to simulate random product placement during actual checkout process.

Meanwhile, Generative Adversarial Networks (GAN) can also be implemented in image synthesis. By providing two different datasets that represent the no-shadow and shadow domain, GAN can be trained to learn the difference between the two domains which eventually allows conversion of images with no-shadow to images with shadow. Such approach can be seen in papers by Li et al. (2019), Wei et al. (2019) as well as Xiao et al. (2020) where a variant of GAN known as CycleGAN released by Zhu et al. (2020) was applied on synthetic images with multiple products generated from RPC dataset to achieve higher reliability of training data with realistic shadows.

2.3. State-of-the-arts object detection models

2.3.1. YOLOv3

YOLOv3 by Redmon & Farhadi (2018) represents the third model in YOLO family and it was developed based on YOLOv2 and introduced several architectural improvement. Firstly, as shown in Figure 3, a larger backbone was used, namely Darknet-53 that incorporated the concept of skip connections in Residual Network (ResNet) that was proposed in He et al. (2015), which will prevent gradient from diminishing during model training. In addition, the new architecture allows detection at three different scales by first downsampling feature maps by 3 different ratios which are 8, 16, 32 for detection of small, medium, and large object respectively before each of them is passed for detection at respective convolutional layers. Thus, the detection performance of small-sized objects can be improved.

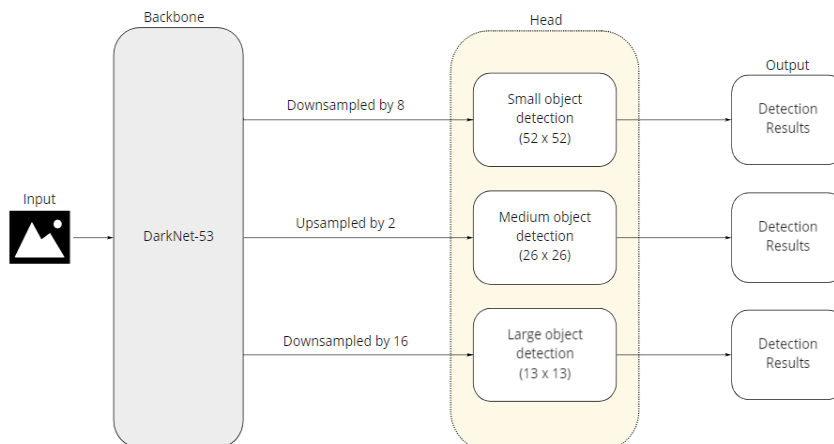


Fig. 3: YOLOv3 model architecture.

2.3.2. YOLOv5

YOLOv5 proposed by Jocher et al. (2021) is a variant of YOLO family that was written in python language rather than C language adopted in previous versions of YOLO. As shown in Figure 4, YOLOv5 can be decomposed into 3 modules which are referred to the backbone for feature extraction, detection neck that is used to generate feature pyramids for object scale generalization and detection head that is responsible for the bounding box regression and object class prediction. In contrast to YOLOv3, YOLOv5 adopts Cross Stage Partial Network (CSPNet) by Wang et al. (2019) and Spatial Pyramid Pooling (SPP) by He et al. (2014) as their model backbone. The architecture can efficiently reduce the repeated gradient by propagating feature maps in two paths and combine after dense and transition layer while adapting to variable input image size. As for the neck, YOLOv5 endorsed Path Aggregation Network (PANet) by Liu et al. (2018) that will improve performance through shorter information path lower and upper feature layers. As for the head, YOLOv5 inherits the architecture of YOLOv3 that predicts at 3 different stages.

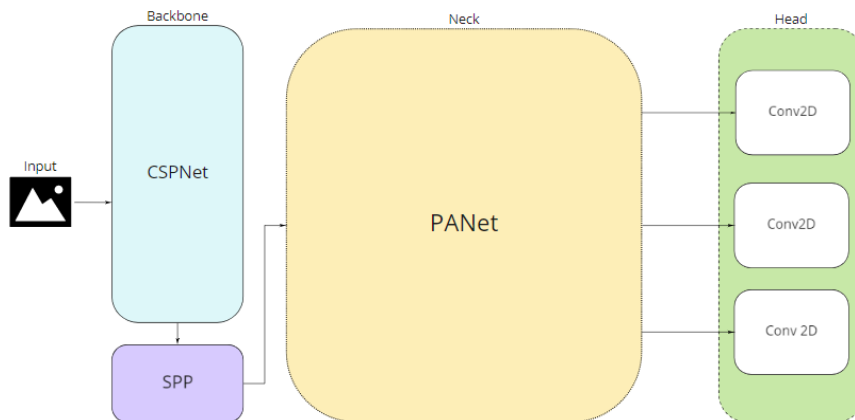


Fig. 4: YOLOv5 Model Architecture.

2.3.3. RetinaNet

RetinaNet released by Lin et al. (2018) is a single-stage object detection model designed to tackle the issue of low foreground-background ratio which is commonly found in single-stage object detectors by introducing a new loss function known as Focal Loss that can emphasize on foreground objects that are hard to detect through larger weights while reducing the importance of easy examples like background, thus elevating the accuracy. In terms of architecture, RetinaNet uses a bottom-up pathway and Feature Pyramid Network (FPN) that was introduced by Lin et al. (2017). It acts as a top-down pathway to allow scale-invariant feature extraction. The backbone is then connected to 2 parallel subnetworks for object classification or bounding box regression task respectively as shown in Figure 5.

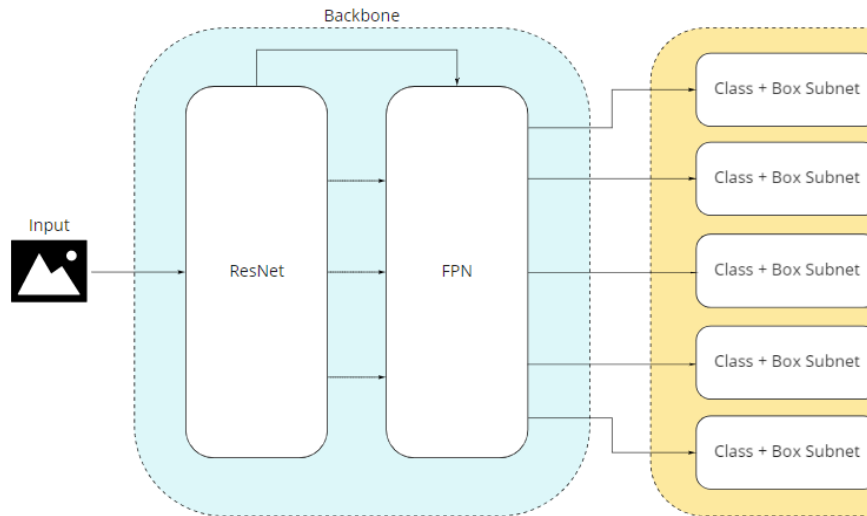


Fig. 5: RetinaNet model architecture.

2.4. Lightweight CNN

2.4.1. MobileNet V3

As the third version of MobileNet series, MobileNet V3 by Howard et al. (2019) inherits the concept of MobileNet family by replacing the traditional convolutional operation with Depthwise Convolutional Filters and Pointwise Convolution that can reduce computational complexity since less multiplications are involved during the process. Additionally, MobileNet V3 also implements Squeeze-and-Excitation (SE) layers which will emphasize important features through compression and restoration. Besides, it adopts a new activation function that is faster to compute since no exponential function is involved compared to sigmoid loss function.

2.4.2. ShuffleNet V2

ShuffleNet V2 by Ma et al. (2018) was constructed based on ShuffleNet V1 and adheres 4 rules of efficient CNN architecture. In contrast to ShuffleNet V1, it introduces a two-channel pathway to prevent computationally expensive group convolutions. Additionally, subsequent layers were removed while preserving channel shuffle, allowing information sharing between channel groups to reduce computational load while improving accuracy.

2.4.3. 4GhostNet

As a state-of-the-art lightweight CNN, GhostNet is proposed by Han et al. (2020). It introduces plug-and-play ghost module that is able to extract equivalent amount of feature maps during convolutional operations with lower Floating-Point Operations (FLOPs) as some of the feature maps will be similar and can be generated from other

essential feature maps using linear operations instead of using convolution that is computationally expensive.

The module will involve traditional convolution to generate feature map with less channel before applying linear operations to generate the feature maps in the remaining channels, thus reducing the overall computational complexity compared to the traditional convolution operation.

3. Methodology

3.1. Dataset preparation and synthesis

In this paper, a small-scale dataset that consists of 20 classes of retail products was acquired through a Huawei P20 camera under environment with uniform lighting and background for easy pre-processing. By using 37 human-captured, single-product images (size of 2976 x 2976) as inputs, multiple training images were generated using a novel image synthesis framework that involved combination of 3 different algorithms, namely crop and paste algorithm, GAN-based shadow synthesis algorithm and light variation algorithm.

First, crop and paste algorithm will randomly select 3 classes of product out of 20 classes of products and raw images of corresponding classes will be cropped and spontaneously placed onto a plain checkout counter image to simulate the random placement of products at the checkout counter. Subsequently, a best performing GAN was utilized to render shadows in the synthesized images after comparison between CycleGAN (Zhu et al., 2020) and AttentionGAN (Tang et al., 2021). The comparison was done based on training of 200 epochs at resolution of 800 x 800 and identity loss of 0.4 for background preservation. Subsequently, lighting variation was incorporated to the rendered images through conventional data augmentation. All the algorithms were associated with automatic annotation as manual image acquisition can be time consuming especially when involving wide range of product classes. Ultimately, the framework formed a dataset with 3000 synthesized training images, 300 validation images and 1200 test images that were captured in the real checkout counter scenario, which is as shown in Figure 6.



Fig. 6: Types of images involved in the proposed checkout system.

3.2. Model training

Prior to improvement and optimization of model, a baseline model was selected from several representative single-stage deep learning models available in the field, consisting of YOLOv3, YOLOv5L, and RetinaNet. By utilizing subset of MVTEC D2S Dataset from Ning et al. (2019), the training of each model was done for 150 epochs at a learning rate of 0.001 with Adam optimizer, weight decay of 0.0005, image size of 512 x 512, and batch size of 16. YOLOv5L was then used for further improvement due to higher performance in the benchmark.

Subsequently, several lightweight experimental models were constructed by replacing the original CSPNet backbone in the YOLOv5 with state-of-the-arts lightweight CNNs, including MobileNet V3, ShuffleNet V2, and GhostNet, which is as shown in Figure 7. Each of the models was then trained for the same hyperparameters for fair comparison with the baseline model.

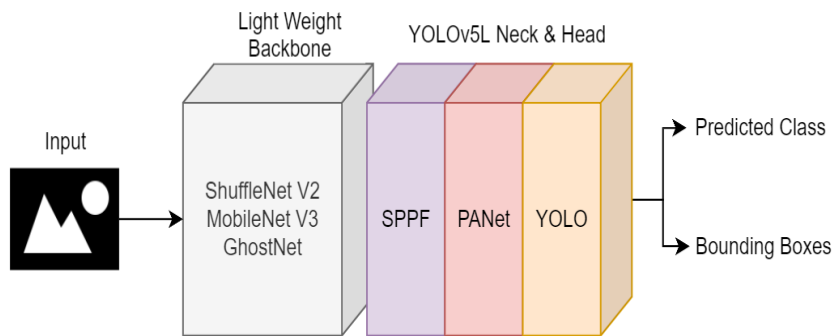


Fig. 7: Experimental YOLOv5 models.

3.3. TensorRT optimization

To further optimize the performance of experimental models on Jetson Nano, NVIDIA TensorRT by NVIDIA (2016) which is a deep learning model optimization runtime was used. The runtime provides a 5-step optimization in maximizing the throughput of deep learning models on embedded systems with NVIDIA GPU and they are referred to the calibration of weight and precision, layers and tensor fusion, auto-tuning of kernel, dynamic tensor memory and CUDA multiple stream execution.

3.4. Evaluation metrics

To evaluate the performance of multiple GANs and product recognition models in the actual in-store checkout, all models in this paper were benchmarked through multiple evaluation metrics. For GANs in our image synthesis framework, Fréchet Inception Distance in Heusel et al. (2018) was used along with qualitative analysis. As for product recognition models, besides involving common metrics like mean Average Precision (mAP) and confusion matrix for accuracy, all models were also evaluated using Checkout Accuracy (cAcc) proposed by Wei et al. (2019) because it

reflects a model's practicality in providing exact quantity and classes of products during actual checkout process.

Concurrently, efficiency of models was also measured through training time and average inference time. For fair comparison, all training time was obtained on Google Colaboratory powered by NVIDIA Tesla P100 Graphical Processing Unit (GPU) while all average inference time was measured on Jetson Nano with NVIDIA Tegra X1 GPU.

3.5. Software prototype

In this paper, the product recognition model was incorporated into a software prototype built using Tkinter library (Python Software Foundation, 2022) and deployed on Jetson Nano equipped with Spedal AF926H Camera. The prototype was also capable to update the model weights automatically and compute using latest product prices through connection to MongoDB database (MongoDB, n.d.).

4. Results and Discussion

Before selecting YOLOv5L as the baseline model for the product recognition task, preliminary benchmark was carried out for multiple single-stage object detection models used in previous works such as YOLOv3, YOLOv5L and RetinaNet. The dataset used at this stage was a subset of MVTec D2S Dataset from Follmann et al. (2018). The results obtained is as tabulated in Table 1.










Table 1: Preliminary benchmark results.

Model	Backbone	mAP (%)	Inference Time (ms)	Training Time (hrs)
Mask R-CNN (Ning et al., 2019)	ResNet- 9	96.20	75.00	-
YOLOv3	Darknet-53	99.79	33.33	10.50
YOLOv5L	CSPNet	99.50	12.20	5.99
RetinaNet	ResNet-50	99.49	35.32	14.26

It can be noticed that all single-stage models outperformed Mask R-CNN as two-stage models especially in inference time and training time. Concurrently, YOLOv5 achieved a relatively short inference time and training time of 12.2ms and 5.69 hours respectively despite having a slightly lower mAP of 99.50 compared to YOLOv3 with mAP of 99.79%. As for RetinaNet, it achieved similar mAP score with YOLOv5 in exchange of high inference time and training time.

On the other hand, to select for the best performing model for shadow synthesis task, benchmark was done between CycleGAN and AttentionGAN, the results can be seen in Table 2.

Table 2: Results of shadow synthesis

	Generated Samples			FID
Input				-
Attention GAN				46.82
Cycle GAN				40.99

From Table 2, CycleGAN offers a more reliable training image with smoother and realistic shadow compared to AttentionGAN under the same hyperparameters especially for products with irregular packaging. Additionally, less degradation in colours and product details can be spotted for images generated by CycleGAN compared to AttentionGAN. As for the background preservation, AttentionGAN can provide a more realistic background since the translation is focused on products through attention mask in its architecture. For quantitative result, FID score of CycleGAN with a value of 40.99 indicates that its generated images are highly correlated with the actual checkout counter scenario. By using CycleGAN in the proposed image synthesis framework, the dataset can be easily constructed while having high correlation with the actual checkout situation without requiring additional effort in image acquisition process.

Additionally, the effect of image synthesis framework on model performance was measured through training and evaluation of YOLOv5L using 3 datasets that represent different levels of image synthesis (Single, Synthesized, GAN-rendered). The results are also compared with re-trained DPNet in Li et al. (2019) which is a

Faster R-CNN that also adopted GAN-rendered images in their approach. The results are as tabulated in Table 3.

Table 3: Results of shadow synthesis.

Model		Non-overlapping		Overlapping		Avg Inference Time (s)
		mAP (%)	cAcc (%)	mAP (%)	cAcc (%)	
DPNet (C. Li et al., 2019)	Single	98.4	56.17	65.4	0.50	1.765
	Syn	97.9	81.83	97.5	76.50	1.768
	Ren	98.8	83.83	97.8	79.67	1.774
YOLOv5L	Single	99.5	94.50	89.8	39.00	0.673
	Syn	99.5	99.33	98.5	96.33	0.689
	Ren	99.5	99.83	98.5	97.33	0.682

It can be noted that YOLOv5L surpassed DPNet at all 3 levels of image synthesis with an average inference time of 0.673s on Jetson Nano. Concurrently, YOLOv5L shows an increment in cAcc when trained with 3 different datasets, especially when overlapped product is present. The cAcc and mAP increased to 96.33%, 98.5% respectively with images generated through crop and paste algorithm. The cAcc elevates further to 97.33% when rendered images are used for training while maintaining mAP and inference time at 98.5% and 0.682s.

Moreover, the CSPNet backbone of YOLOv5L model was replaced with other lightweight CNNs to allow efficient deployment of Jetson Nano. The results of each variant are recorded in Table 4.

Table 4: Performance of experimental models.

Backbone	GFLOPs	mAP (%)	cAcc (%)	Training Time (hrs)	Inference Time (s)
CSPNet (Baseline)	108.1	98.5	97.33	6.470	0.505
GhostNet	42.5	98.2	89.33	3.852	0.244
ShuffleNet V2	40.7	98.2	87.83	2.891	0.217
MobileNet V3	38.5	98.2	89.17	2.019	0.200

From Table 4, it can be noticed that all lightweight models have successfully reduced the FLOPs of YOLOv5L baseline model to low values in exchange of slight degradation of mAP and cAcc. Among all lightweight models, GhostNet with 42.5 GFLOPs the smallest mAP, cAcc degradation of 0.3% and 8% from the baseline

performance respectively. Simultaneously, MobileNet V3 variant has achieved similar performance with lower GFLOPs, whereas ShuffleNet V2 has largest drop in cAcc, which is 9.5% from the baseline model. As for the training time and inference time MobileNet V3 outperformed other models with shortest training time of 2.019 hours and 0.2s of inference time.

On the other hand, qualitative analysis was performed for each model to justify their adaptability to lighting variation and extreme condition with heavily overlapped products. Their predictions can be seen in Table 5, Table 6, Figure 8, and Figure 9.

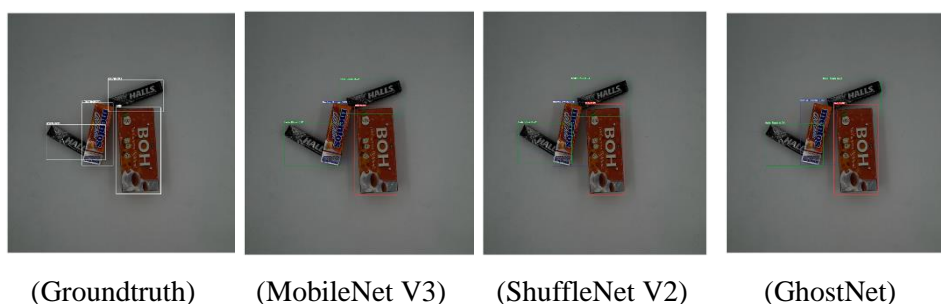


Fig. 8: Models' predictions in low-light conditions.

Table 5: Confidence score in low-light condition

Items	Confidence Score		
	MobileNet V3 + YOLOv5L	ShuffleNet V2 + YOLOv5L	GhostNet + YOLOv5L
Boh	0.95	0.95	0.88
Halls Black	0.93	0.94	0.95
	0.91	0.93	0.96
Mentos Orange	0.88	0.91	0.90



(Groundtruth) (MobileNet V3) (ShuffleNet V2) (GhostNet)

Fig. 9: Models' predictions in extreme condition.

Table 6: Confidence score in extreme condition.

Items	Confidence Score		
	MobileNet V3 + YOLOv5L	ShuffleNet V2 + YOLOv5L	GhostNet + YOLOv5L
Dequadin	0.96	0.95	0.98
Full Cream Milk	0.92	0.92	0.96
Low Cream Milk	0.97	0.96	0.95
Halls Black	0.95	0.95	0.99
HFT Black Soya	0.95	0.95	0.96
KitKat	0.97	0.90	0.98
Mentos Orange	0.91	0.89	0.95
NutriOne Nuts	0.95	0.96	0.96
Ricola Lemon	0.93	0.95	0.97

In low-light condition, all models can locate the products accurately using bounding boxes and predict each product at high confidences. Whereas for extreme condition, all product classes are accurately detected. However, MobileNet V3 and GhostNet variant outperform ShuffleNet V2 with relatively high and stable confidence score. However, it can be noticed that the bounding boxes generated by MobileNet V3 variant are less converged to the ground truth compared to other models.

Furthermore, to pursue for faster inference time on Jetson Nano, TensorRT was applied to all models and their optimized inference time are shown in Figure 10. All models' inference time on Jetson Nano are successful reduced with a minimal inference time of 0.142s with MobileNet V3 variant.

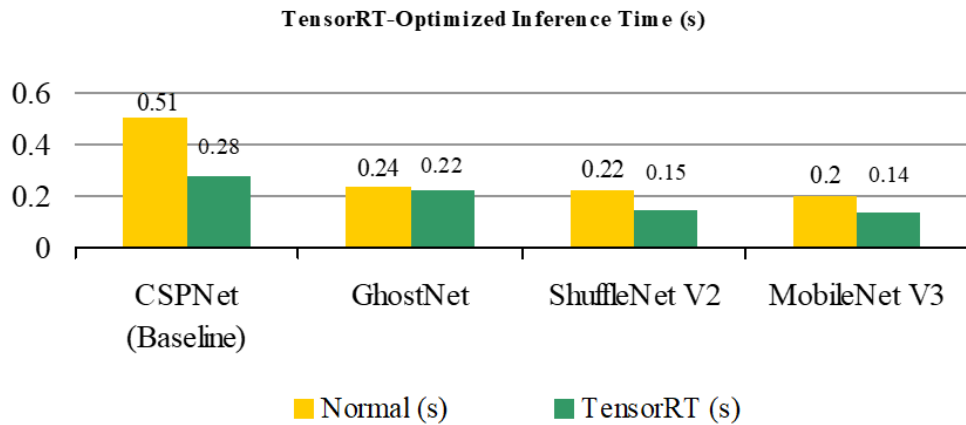


Fig. 10: TensorRT-optimized inference time.

Finally, software prototype was constructed using the most optimized model which is MobileNet V3 variant. By utilizing frame difference in OpenCV, the software will detect for user’s hand movement before performing inference, which can essentially reduce false positive rate. Moreover, Additional features such as model weight update and continuous price computation were included to further reduce human requirement in system maintenance and update. The features are shown in Figure 11 and Figure 12.



Fig. 11: Frame difference for movement detection.



Fig. 12: Continuous price computation.

5. Conclusion

In brief, this paper proposes a software prototype of cashierless checkout system by utilizing a YOLOv5 deep learning model at its core for product recognition and price computing to reduce the risk of virus spread chain during in-store checkout. The training of the product recognition model is supported by a novel 3-stage image synthesis framework that includes crop and paste algorithm, CycleGAN, and conventional image augmentation to effectively simulate the actual checkout scenario with less human involvement. With the highly correlated training data, mAP and cAcc of YOLOv5L are increased to 98.5% and 97.33% under checkout scene with occluded products. Furthermore, multiple lightweight models of YOLOv5L were developed and MobileNet V3 variant achieved an optimal inference time of 0.142s on Jetson Nano while having minimal degradation in mAP of 0.3% and cAcc of 8.16% after optimized with TensorRT. Thus, it can be concluded that the proposed software prototype is suitable to be implemented in small-scale smart retail. For future works, further studies can be done regarding incremental learning technique to allow minimal training time especially when new retail product is introduced.

References

- Amazon.com: Amazon Go. (n.d.). From <https://www.amazon.com/b?ie=UTF8&node=16008589011>.
- Bankim, R. V. (2015), Retail management. *IJRAR- International Journal of Research and Analytical Reviews*, 2(1). http://ijrar.com/upload_issue/ijrar_issue_139.pdf.
- Bi, X. & Wang, L. (2021), Performing weakly supervised retail instance segmentation via region normalization. *IEEE Access*, 9, 67761–67775. DOI:<https://doi.org/10.1109/ACCESS.2021.3077031>.
- Follmann, P., Böttger, T., Härtinger, P., König, R., & Ulrich, M. (2018). MVTec D2S: Densely segmented supermarket dataset. *Proceedings of the European Conference on Computer Vision (ECCV)*, 11214, 581–585. DOI:<https://doi.org/10.48550/arXiv.1804.08292>.

Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., & Xu, C. (2020), GhostNet: More features from cheap operations. *ArXiv:1911.11907 [Cs]*. DOI:<https://doi.org/10.48550/arXiv.1911.11907>.

He, K., Zhang, X., Ren, S., & Sun, J. (2014), spatial pyramid pooling in deep convolutional networks for visual recognition. *ArXiv:1406.4729 [Cs]*, 8691, 346–361. DOI:https://doi.org/10.1007/978-3-319-10578-9_23.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. *ArXiv:1512.03385 [Cs]*. DOI:<https://doi.org/10.48550/arXiv.1512.03385>.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2018). GANs trained by a two time-scale update rule converge to a local nash equilibrium. *ArXiv:1706.08500 [Cs, Stat]*. <http://arxiv.org/abs/1706.08500>.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., & Adam, H. (2019). Searching for MobileNetV3. *ArXiv:1905.02244 [Cs]*. DOI:<https://doi.org/10.48550/arXiv.1905.02244>.

Jocher, G., Stoken, A., Borovec, J., NanoCode012, Chaurasia, A., TaoXie, Changyu, L., V, A., Laughing, tkianai, yxNONG, Hogan, A., lorenzomamma, AlexWang1900, Hajek, J., Diaconu, L., Marc, Kwon, Y., oleg, & Ingham, F. (2021). *ultralytics/yolov5: V5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations*. DOI:<https://doi.org/10.5281/zenodo.5563715>.

Koturwar, S., Shiraiishi, S., & Iwamoto, K. (2019). Robust multi-object detection based on data augmentation with realistic image synthesis for point-of-sale automation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(1), 9492–9497. DOI:<https://doi.org/10.1609/aaai.v33i01.33019492>.

Kumar, M. S., Raut, D. R. D., Narwane, D. V. S., & Narkhede, D. B. E. (2020). Applications of industry 4.0 to overcome the COVID-19 operational challenges. *Diabetes & Metabolic Syndrome*, 14(5), 1283–1289. DOI:<https://doi.org/10.1016/j.dsx.2020.07.010>.

Li, C., Du, D., Zhang, L., Luo, T., Wu, Y., Tian, Q., Wen, L., & Lyu, S. (2019). Data priming network for automatic check-out. *ArXiv:1904.04978 [Cs]*. DOI:<https://doi.org/10.48550/arXiv.1904.04978>.

Li, F.-F., Fergus, R., & Perona, P. (2006). One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4), 594–611, DOI:<https://doi.org/10.1109/TPAMI.2006.79>.

Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2018). Focal loss for dense object detection. *ArXiv:1708.02002 [Cs]*, 1–29. DOI:<https://doi.org/10.48550/arXiv.1708.02002>.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. *ArXiv:1612.03144 [Cs]*. DOI:<https://doi.org/10.48550/arXiv.1612.03144>.

Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018), Path aggregation network for instance segmentation. *ArXiv:1803.01534 [Cs]*. DOI:<https://doi.org/10.48550/arXiv.1803.01534>.

Ma, N., Zhang, X., Zheng, H.-T., & Sun, J. (2018), ShuffleNet V2: Practical guidelines for efficient CNN architecture design. *ArXiv:1807.11164 [Cs]*. <http://arxiv.org/abs/1807.11164>.

Ministry of Health Malaysia. (2020), COVID-19: Management Guidelines For Workplaces, From: https://www.moh.gov.my/moh/resources/Penerbitan/Garis%20Panduan/COVID19/Annex_25_COVID_guide_for_workplaces_22032020.pdf.

MongoDB. (n.d.), PyMongo 4.1.1 Documentation. From: <https://pymongo.readthedocs.io/en/stable/>.

Ning, J., Li, Y., & Ramesh, A. (2019). Simplifying Grocery Checkout with Deep Learning. <https://www.semanticscholar.org/paper/Simplifying-Grocery-Checkout-with-Deep-Learning-Ning-Li/975ff328a7c86b593ed2501a1dd33a43e629b8bc>.

NVIDIA. (2016). *NVIDIA TensorRT*. NVIDIA Developer. From: <https://developer.nvidia.com/tensorrt>

Panasonic. (2018). *RFID Based Walk-through Checkout Solution for Future Retail*. Panasonic Newsroom Global. From: <http://news.panasonic.com/global/topics/2018/55288.html>.

Periyasamy, M. & Dhanasekaran, R. (2014). Assessment and analysis of performance of 13.56 MHz passive RFID in metal and liquid environment. *2014 International Conference on Communication and Signal Processing*, 1122–1125. DOI:<https://doi.org/10.1109/ICCSP.2014.6950023>.

Polacco, A. & Backes, K. (2018). The amazon go concept: Implications, applications, and sustainability. *Journal of Business and Management*.

Python Software Foundation. (2022). Tkinter—Python interface to Tcl/Tk—Python 3.10.4 documentation. <https://docs.python.org/3/library/tkinter.html>.

Rathnayake, L. R. & Nawarathna, R. D. (2020). Semi-supervised learning approach to multiclass object detection with obscure and overlapping boundaries. *2020 4th SLAAI International Conference on Artificial Intelligence (SLAAI-ICAI)*.

Redmon, J. & Farhadi, A. (2018), YOLOv3: An Incremental improvement. *ArXiv:1804.02767 [Cs]*. DOI:<https://doi.org/10.48550/arXiv.1804.02767>.

- Rigner, A. (2019), AI-based machine vision for retail self-checkout system. *Master's Theses in Mathematical Sciences*. <http://lup.lub.lu.se/student-papers/record/8985308>.
- Tang, H., Liu, H., Xu, D., Torr, P. H. S., & Sebe, N. (2021). AttentionGAN: unpaired image-to-image translation using attention-guided generative adversarial networks. *ArXiv:1911.11897 [Cs, Eess]*. DOI:<https://doi.org/10.48550/arXiv.1911.11897>.
- Wang, C. Y., Liao, H. Y. M., Yeh, I. H., Wu, Y. H., Chen, P. Y., & Hsieh, J.-W. (2019), CSPNet: A new backbone that can enhance learning capability of CNN. *ArXiv:1911.11929 [Cs]*. DOI:<https://doi.org/10.48550/arXiv.1911.11929>.
- Wei, X. S., Cui, Q., Yang, L., Wang, P., & Liu, L. (2019). RPC: A large-scale retail product checkout dataset. *ArXiv:1901.07249 [Cs]*. DOI:<https://doi.org/10.48550/arXiv.1901.07249>.
- Xiao, Z., Zhao, J., & Sun, G. (2020), Mask data priming network for automatic checkout, *Preprints 2020, 2020060170*. DOI:<https://doi.org/10.20944/preprints202006.0170.v1>.
- Yi, W., Sun, Y., Ding, T., & He, S. (2019). Detecting retail products in situ using CNN without human effort labeling. *ArXiv:1904.09781 [Cs]*. DOI:<https://doi.org/10.48550/arXiv.1904.09781>.
- Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2020), Unpaired image-to-image translation using cycle-consistent adversarial networks. *ArXiv:1703.10593 [Cs]*. <https://doi.org/10.48550/arXiv.1703.10593>.