




Apply Deep Learning in Real-time Customer Detection and Classification System for Advertisement Decision Making at Supermarket

Dang Thi Phuc¹^a, Dau Sy Hieu²^b, Nguyen Manh Hoang¹ and Tran Thi Minh Khoa¹^c

¹*Department of Computer Science, Faculty of Information Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam*

²*Department of Applied Physics, University of Technology, Vietnam National University HCMC, Ho Chi Minh City, Vietnam*

Keywords: Internet of Thing, Artificial Intelligence, YOLOv4, Embedded System, Jetson Nano.


Abstract: Shopping in shopping malls and supermarket is gradually increasing and replacing traditional market because of the convenient conditions such as full of products, clean place with all time air-conditioner, modern environment, etc. Supermarket owner always want to find the way that attract more and more customers come to the supermarket as well as advertise their products to the customers as many as possible. In order to offer relevant and attractive advertising to customers, detection and classification customers entering to the supermarket is taken into consideration. Due to the characteristics different customer groups, the relevant products should be showed to attract customer, help them save the time for finding products. In this paper, we build a real-time customer detection and classification system at the supermarket. The goal of this proposed Internet of Things (IoT) system is automatically show the suitable advertising clips to many customers at the right time. We build a classification model using deep learning with a large amount of data. The dataset is collected from reality and labelled with five different object classes. To ensure reliability, 7000 images are collected from different conditions such as variations in camera used, bad lighting, angles, and not stable background. The data is trained on YOLOv4 and YOLOv4-tiny models. The models are deployed on the embedded system with the Jetson Nano device as the processor. We compare the accuracy and speed of the two models on the same embedded system, analyse the results, and chose the best model according to the specific system requirements.


1 INTRODUCTION


Automatically detecting and classifying customer groups entering to the shopping malls or supermarkets is very interesting as well as profitable for suppliers advertising their products to customers. Based on the detecting and classifying results, advertising system can quickly pick suitable advertisements clips to be displayed on the certain screens which are close to customers position to facilitate customers choosing products or suggest more available products that may suit to the customers needs. In this research, we aim to develop a real-time customer detection and classification

system for supermarket advertising using deep learning. This system requires high accuracy of detecting and classifying different types of customers, and in some different conditions such as lighting, camera quality, too many customers at the same time, and fast real-time execution speed, etc. In addition, the devices need to be compact enough in order to be easily mounted to the available system in the supermarket. Therefore, the system face to the two important problems:

- Build a detection and classification model must satisfy with the above requirements.
- Deploy the model to a compact system and ensure real-time data processing.

^a <https://orcid.org/0000-0003-0984-3912>

^b <https://orcid.org/0000-0003-4507-7856>

^c <https://orcid.org/0000-0002-2668-5998>

There are many IoT systems were built for detecting and classifying objects extracting from a real-time camera using image processing techniques (Dorothy et al., 2017). The problem can only detect a few of large and clear objects (Rane et al., 2017) but under different conditions of environment, especially in the lighting condition, the system's efficiency was quite strongly affected. A new research direction helps IoT systems become more flexible and intelligent is adding Artificial intelligence (AI) algorithms to the system. Some AI algorithms can improve the efficiency of computer vision methods are Artificial Neural Network (ANN) (Ahmed et al., 2020), Support Vector Machine (SVM) (Lalitha et al., 2021)... For images processing, deep learning technique (Convolutional Neural Networks (CNN)) is an effective algorithm for object classification problem whether there is impact from the environment such as noise, changing in distance between object and the camera (Hsieh and Jeng, 2018). Deep learning models for effective object detection and classification with high accuracy and achieved remarkable achievements in problems requiring accuracy such as in medicine (Saadawy et al., 2021), industry or robotics (Hosseini et al., 2020) are VGG16 (Simonyan and Zisserman, 2015), AlexNet (Krizhevsky et al., 2012), Xception (Chollet et al., 2017)... However, the disadvantages of these models includes cumbersome, implementation time and real-time problem. In other hand, they require powerful hardware for deployment and operating. Therefore, it faces challenge to deploy these models to current common Internet of things (IoT) systems.

One of solutions for the above problems is deploy the model on a powerful servers. Images taken from cameras are pushed to server for processing, and then send results back to devices. However, the processing and transmission time surely affect and lead to a long latency that may not be suitable for many applications (Hsieh and Jeng, 2018). In order to meet the real-time requirement, we prefer to process at the device so that the system can avoid the time for transferring data and ensure processing speed (Lalitha et al., 2020).

Many deep learning models were created for real-time problems such as proposed Region Convolutional Neural Networks (R-CNN), Fast R-CNN (Girshick, 2015), Mask R-CNN (He et al., 2015), Single Shot Multi Box Detector (SSD) (Liu et al., 2016), RetinaNet (Lin et al., 2020), You Only Look Once (YOLO) (Redmon et al., 2016). YOLOv4 is the new YOLO algorithm created by Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao announced on April 23, 2020, with outstanding advantages such as easy-to-access model

architecture, ensuring very fast speed, suitable for real time. Compared with the state-of-art models mentioned above on the MS COCO dataset, YOLOv4 achieved the best results : YOLOv4 hit 43.5% AP (average precision) on MS COCO dataset at 65 FPS (frame per second) on Tesla V100 GPU (Bochkovskiy et al., 2020). In addition, YOLO also has tiny versions with simpler network architecture to increase processing speed and still ensure accuracy such as YOLOv3-tiny, YOLOv4-tiny (Jiang et al., 2020).

In this paper, we propose two models in YOLOv4 and YOLOv4-tiny solving the problem of real-time people detection and classification in video capturing by digital camera. Training dataset is created with 7000 images collected by digital camera under different conditions to ensure the accuracy of the model. We recommend the Jetson Nano as a device for deployment the YOLOv4 model in order to ensure accuracy, speed and compactness (Uddin et al., 2021). It is compact with a reasonable price with a powerful and modern configuration which well supports TensorRT library that allows to optimize deep learning models, accelerate and ensure accuracy.

2 RELATED WORKS

2.1 Deep Learning Model for Object Recognition and Classification Problem

Convolutional Neural Network (CNN) is one of the deep learning algorithms that gives the best results in most of machine vision problems such as classification and recognition (Ahmed et al., 2020). CNN is designed by combining multiple convolutional layers, pooling layers, and fully connected layer. In which, convolution layers play the role of extracting diverse features of the image using combination of different filters, pooling layer reduces the number of parameters to help speed up the calculation and fully connected layer classifies objects based on probability. The advantage of the CNN is that it can process large images, can achieve high accuracy but the network architecture of CNN use to be quite cumbersome. Training CNN model usually requires high costs. Moreover, operating the model requires a powerful hardware configuration. Our propose system requires not only the accuracy but also the ability to perform real-time on compact and less-powerful devices. In order to meet the above

requirements, we aim to design a suitable network architecture for the CNN models. Additionally, the proposed system classifies not only objects but also identifies which objects are detected in each frame in a not stable background caused by surrounding environment.

Many advanced CNN algorithms that can be used for object detection such as R-CNN series, SSD, YOLO, RetinaNet. R-CNN uses AlexNet network architecture to detect and classify images from bounding boxes packed by selective search algorithm and classify by SVM or full connected. R-CNN is limited in speed and number of classified objects of about 2000 objects. Other improvements help to overcome this drawback can be listed out such as Faster R-CNN, Fastest R-CNN by using Region Proposal Networks (RPN) to predict bounding box. Those are predicted can achieves high efficiency in processing image regions that contain objects and can be applied in real-time problems. However, the weakness of this network is the prediction of the bounding box and the classification of objects in the box do not take place at the same time (Girshick, 2015). SSD, YOLO, and RetinaNet have a more modern network architecture than R-CNN in which those can combine bounding box prediction and object detection at the same time. SSD detects objects using a regression algorithm can increase processing speed, but the probability of object detection is reduced. SSD uses VGG16 network model for feature extraction and objects detection in 2 stages: feature extraction and convolutional filter application for object detection (Liu et al., 2016). RetinaNet uses additional Focal Loss to increase accuracy of predicting the object’s location (Lin et al., 2020). YOLO divides the image into grid cells, on each grid cell runs a feature extraction algorithm and image classification and restriction. Compared to R-CNN, SSD and YOLO have made great improvements over the versions, especially YOLOv4 with high accuracy and outstanding speed. YOLOv4 use to be chosen for solving fast-paced problems and small objects

identification. One more advantage of YOLO is its open source so that researchers can easily used for AI applications, for example self-driving cars, cancer detection, etc... (Srivastava et al., 2021).

You Only Look One (YOLO) model was first described by Joseph Redmon, et al. in 2015. Unlike R-CNN, YOLO divides image into grid cells and then traverses each grid cell, classifies object classes, and bounding box each object in the image. The improved version YOLOv2 proposed in 2016 is capable for predicting up to 9000 different types of objects. YOLOv2 uses Darknet-19 network architecture. YOLOv3 2018 improves to the Darknet-53 network model to increase accuracy, predicts an objectness score for each bounding box using logistic regression (Redmon and Farhadi, 2018). Darknet-53 is built on 53 layers, and so, it improves both accuracy and speed by 2 times compared to ResNet-152. YOLOv4 is more completed in network architecture with the result of increasing accuracy and speed by 10-12 % compare to YOLOv3.

In this paper, we propose a detection and classification system using YOLO model, especially YOLOv4, to reach our system requirements. The network architecture of YOLOv4 is depicted in Figure 1. The model consists of three main parts: the Darknet-53 CSP backbone, the neck part and the YOLO head.

The Darknet-53 CSP backbone increases computation speed and accuracy. The CSPNet architecture reduces the number of parameters. Moreover, the Mish activation function improves accuracy of object classification for YOLOv4.

The neck part includes SPP block (He et al., 2015) and PANet model (Liu et al., 2018). SPP blocks extracts important feature regions without slowing down the model. However, the deeper neural networks, the easier to lose information. There are many approach are proposed to detect small objects.

PAN, an improvement of FPN, is one of them. PANnet connects the learned local with global features of an object for more accurate prediction.

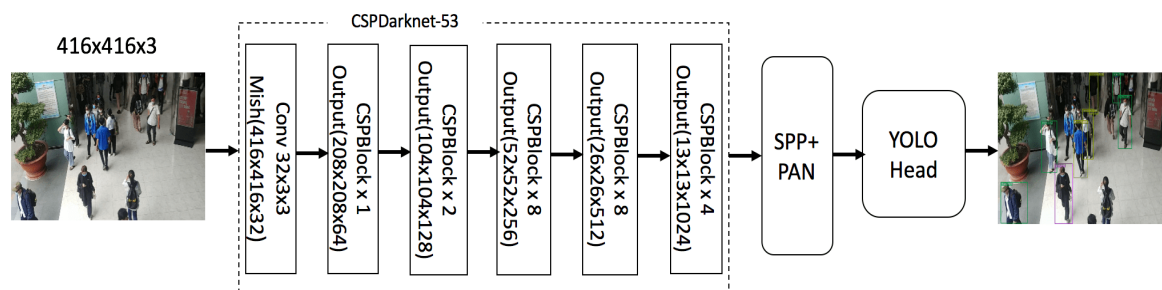


Figure 1: Network architecture of YOLOv4.

The YOLO head is used to increase feature discrimination for object classification and bounding box objects. Same to YOLOv3, YOLOv4 predicts object and bounding-box using logistic regression. They also classifies objects using independent logistic classifier instead of softmax function.

YOLOv4 calculates Loss Value using Complete Intersection over Union (CIoU) value. CIoU can achieve better convergence speed and accuracy of bounding box regression problem based on below formulas (Du et al.,2021):

$$L_{CIoU} = 1 - CioU \quad (1)$$

where,

$$CIoU = IoU - \frac{\rho^2(b, b^{gt})}{c^2} - \alpha v, \quad (2)$$

$$\alpha = \frac{v}{1 - IoU + v}, \quad (3)$$

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2 \quad (4)$$

where: α - a positive trade-off parameter; v - the consistency of aspect ratio; w^{gt} with h^{gt} - the width and height of the real frame; w , h - the width and height of the prediction box; $\rho(\cdot)$ - Euclidean distance; b , b^{gt} - center point coordinates of prediction box and real frame; c - diagonal distance of the minimum rectangle that can cover real frame and prediction box simultaneously.

$IoU = \frac{box A \cap box B}{box A \cup box B}$ - Intersection ratio - the ratio of the two boxes A and B. The IoU is used to evaluate the distance between the predict box and the ground-truth (Nowozin et al.,2014).

To evaluate the accuracy of the YOLOv4 model, we also use mAP (mean Average Precision) in the classification task. The mAP is the average of the average precision values for all categories. Average Precision(AP) for each category can be computed on below formula:

$$R = \frac{TP}{(TP+FN)} \quad (5)$$

$$P = \frac{TP}{(TP+FP)} \quad (6)$$

R – recall, is the proportion of the number correctly recognized by the network for each category; P – precision, is the number of correct results identified by the network; TP – true positive samples; FP - false positive samples; FN – false negative samples.

$$P = \sum_{k=0}^{n-1} [R(k) - R(k+1)] * P(k) \quad (7)$$

The mAP is obtained by formula:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (8)$$

To evaluate performance of model when deploying on embedded devices, we use FPS (frames per second). FPS denotes the number of images that can be detected successfully in one second.

To improve the real-time of object detection, other thin and light versions of YOLO such as YOLOv2-tiny, YOLOv3-tiny, YOLOv4-tiny were created. YOLOv2-tiny delete convolution layers in Darknet19 network to 9 layers to reduce the network complexity. YOLOv3-tiny is proposed by compressing the network model of YOLOv3, using seven layers of convolutional networks and six layers of maximum aggregation instead of the ResBlock structure in the DarkNet53 network. YOLOv4-tiny uses CSPDarknet53-tiny instead of CSPDarknet53 network. CSPDarknet53- tiny uses the LeakyReLU function as the activation function instead of the Mish activation function, which simplifies the calculation process. However, with this network architecture, YOLOv4-tiny gives low prediction efficiency compared to YOLOv4 model.

Based on the advantages and disadvantages of the models, we decided training our model on the two models YOLOv4 and YOLOv4-tiny. Also, we deploy the model on device and make an evaluation. And then, we select the best results based on the criteria of our system requirements.

Hardware: In this implementation, we use NVIDIA Jetson Nano development toolkit with 4GB RAM version, shown in Figure 2, that support object recognition using AI algorithms. It is a compact integrated AI computer with extremely powerful that allows to run multiple neural networks in parallel for image processing applications. However, the greatest strength of Jetson Nano is the Nvidia's GPU-128-core Maxwell inside that allows the deployment of deep learning algorithms much more smoothly. Besides, Jetson Nano is also capable for decoding 8 video streams extract from the high resolution camera.

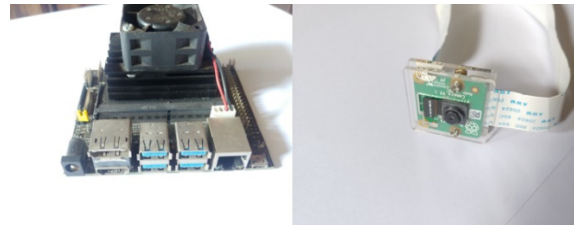


Figure 2: Jetson Nano and its camera.

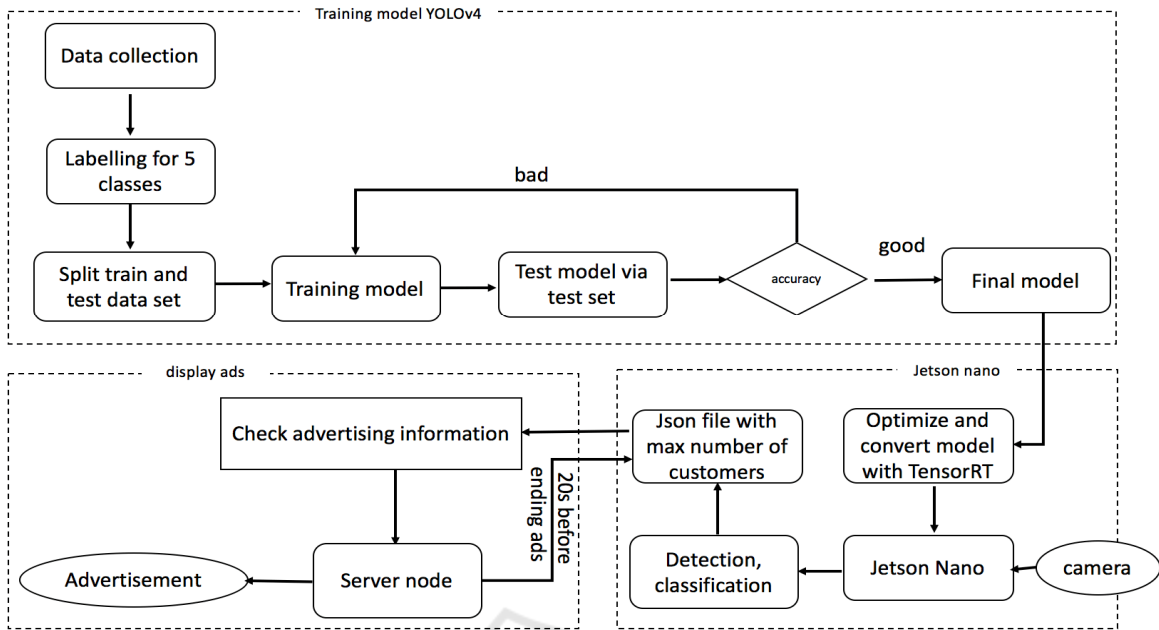


Figure 3: System architecture.

2.2 System Architecture

The system architecture is shown in Figure 3. The customer classification using deep learning is solved in two stages:

Stage 1: Train the deep learning model.

Stage 2: Deploy the model to the real-time guarantee system.

In model training stage, we perform following steps:

Data Collection: Due to our research on customer types and their needed products when they come into the supermarkets or shopping malls, we split customers into 4 main groups: male from 18-35 years old, female from 18-35 years old, the elder and children. Besides, we target people who is entering supermarket, it means we need to classify those who have a direction towards the supermarket only. Hence, we must classify total of 5 groups of customer: one group of not-coming customer and four main groups of in-coming customer. For each certain group, advertising system have suitable advertising clips of products that can attract the customer. When the classification process finish, the system counts the number of customers of each group. The advertising system will decide to play advertising clip which suitable for the biggest group entering supermarket at that moment.

In order to ensure the accuracy of customer classification, we collect data from many different areas such as: Industrial University of Ho Chi Minh

City, Gigamall Thu Duc, Saigon CENTER, CCTV in Shopping Mall. From the recorded videos, images are extracted at different moment to avoid data duplication. Cameras were mounted in different locations, with the right distances to create diverse data sources (see Figure 4). Our total classification class consists of 5 classes corresponding to the 5 customer groups. Except for 80% actual collected data, we also add 20% data collected images and videos from the internet to increase the data's diversity. The total of collected data is 7000 photos.

Data Labeling for 5 Classes: Our collected images will be labeled with the 5 corresponding classes as above. We determined coordinates of the boxes containing the customer object and labeled them with Labelling tool.

We divide collected data into 6000 images for training and 1000 images for testing.

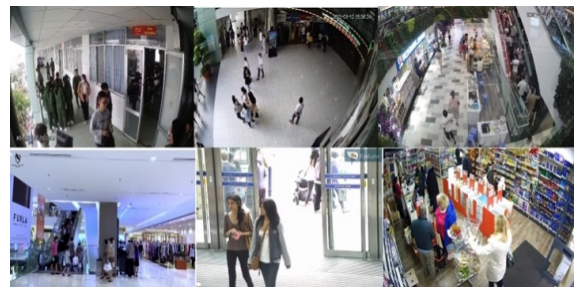


Figure 4: Images taken in different place and conditions.

Model Training: We use 2 models for training: YOLOv4 and YOLOv4-tiny. During the training process, we refine the model hyperparameter, add data and continue training many times to achieve the most optimal model. Models were trained using Google Colab, accuracy is evaluated based on the following values: Loss function, CIOU avg, mAP. Based on the training results, we evaluate the advantages and disadvantages of YOLOv4 and YOLOv4-tiny models, and then choose the better one to deploy to Jetson Nano.

Deploying the Model to Jetson Nano: The current YOLOv4 and YOLOv4-tiny models running on an existing NVIDIA Jetson Nano give a very low FPS (less than 1) that's not acceptable for real-time requirement. To solve this problem, we provide a solution to improve the FPS of the model by compressing and optimizing the model using the TensorRT library. TensorRT is built on the parallel programming model of CUDA and NVIDIA. It allows optimization of libraries, development tools, and technologies in CUDA-X for AI. It also optimizes deep learning models and archive high performance of the system.

Display Advertising: Videos extracted from the camera will be sent to Jetson Nano for prediction. Prediction results include information such as: object identification, object classification and statistics the number objects of each type. The advertising system will check the biggest group of customer and pick the suitable advertising clips to show on the display screen. 20 seconds before current advertising clip end, the advertising system will send a request to Jetson Nano for new prediction result, select biggest group and play the next clip.

3 RESULTS

3.1 Model Training Results

For a dataset of 7000 images classified on 5 classes: Man: 18-35 years old male, Girl: 18-35 years old female, Children: under 15 years old children, Elder: Old people and Person - customers do not go into the supermarket. We divide dataset into 6000 images for training set and 1000 images for testing set. YOLOv4 and YOLOv4-tiny models were trained with the parameters: momentum of the stochastic gradient descent was set as 0.949 and the learning rate 0.001, the weight decay was set as 0.0005. The batch size was set to 64 to improve the utilization of the GPU and its memory. The input image will be resized to 416 x 416 x 3. Models were trained using Google

Colab. The training process of both models is 15000 iterations. Training time on YOLOv4 and on YOLOv4-tiny are 84 hours and 22 hours respectively. Figure 5 shows that YOLOv4 model achieves mAP value after 15000 iterations of 91.7%, with avg loss of 5.3869. Figure 6 shows that the YOLOv4-tiny model achieved mAP value after 15000 iterations of 74.6%, avg loss value of 1.7. Both two models converge well.

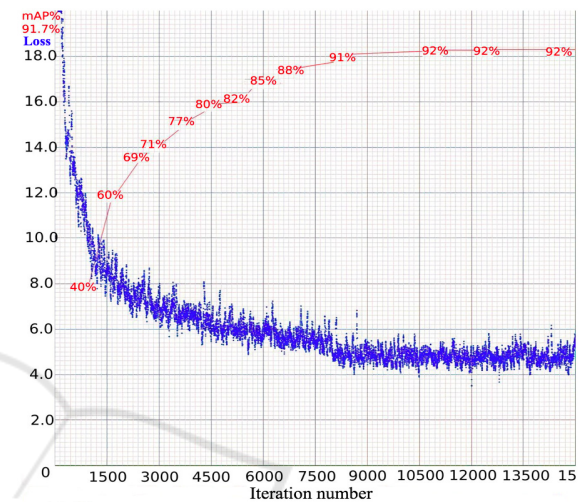


Figure 5: Model training graph of YOLOv4.

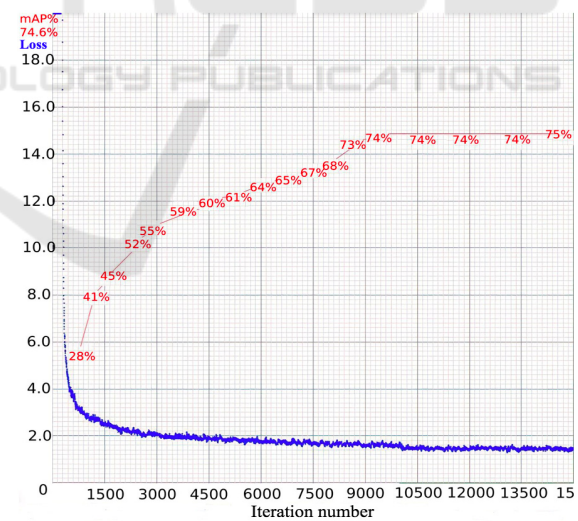


Figure 6: Model training graph of YOLOv4-tiny.

The accuracy values of the corresponding classes of the YOLOv4 and YOLOv4-tiny models are described in Table 1. Results shown that all the feature classes of the YOLOv4 model are superior to the YOLOv4-tiny model.

Table 1: Accuracies (%) of YOLOv4 and YOLOv4-tiny models.

Class	YOLOv4	YOLOv4-tiny
Man	93.74	78.72
Girl	90.41	67.96
Elder	88.66	75.88
Children	95.69	81.43
Person	89.97	68.87

Based on the results of testing dataset (Figure 7 and Figure 8), the YOLOv4 model recognizes more objects than the YOLOv4-tiny model, even objects far away from the camera. Moreover, the accuracy of object recognition is also higher than YOLOv4-tiny model. Some objects are misidentified with YOLOv4-tiny.



Figure 7: Test results on the model YOLOv4.

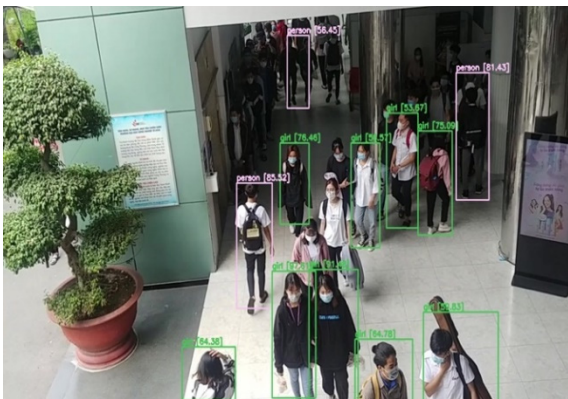


Figure 8: Test results on the model YOLOv4-tiny.

In the bad lighting condition, the results of object recognition with YOLOv4 model is also reach the high accuracy, shown in Figure 9.



Figure 9: Recognition results in different lighting conditions of YOLOv4.

3.2 Deployment Result

We optimize 2 models YOLOv4 and YOLOv4-tiny using TensorRT library, and deploy to Jetson Nano. The compare deployment results is shown in Table 2.

Table 2: Deployment results to Jetson Nano.

Model	File weight (MB)	FPS	mAP(%)
YOLOv4	224.2	4.19	91.7
YOLOv4-tiny	22.5	40	74.57

Since the YOLOv4 model has a more complex architecture than the YOLOv4-tiny model, a larger file weight capacity is obtained after training process. Therefore, it run slower than YOLOv4-tiny when deployment to Jetson Nano. However, the accuracy of YOLOv4 model is higher than that of YOLOv4-tiny.

The proposed system requirements focus on accuracy and speed. We can see from the above experimental results that the speed of the YOLOv4-tiny model is faster than YOLOv4. However, in both of object detection and classification, the YOLOv4-tiny is worse than YOLOv4 in terms of accuracy. This surely affects to make decision on choosing advertising clip.

The YOLOv4 model is better in term of object recognition and classification capabilities, even with difficult cases of small and fuzzy objects. This advantage is more suitable for the initial requirements. Since the objects are walking with normal speed and there is a gap time among advertising clips, the continuous identification is not needed. Hence, disadvantage of YOLOv4's speed is still acceptable.

3.3 Advertising Clips Displaying

The advertising clips displaying process is described in Figure 10. Images are extracted from camera. Using YOLOv4 model deployed on Jetson Nano to identify objects. Based on recognition results, the system can define certain group with largest number of objects type. Then, system make decision for playing the corresponding advertising.



Figure 10: The advertising displaying process.

4 CONCLUSIONS

In this paper, we build an IoT system of detecting and classifying customer groups from video recorded by a digital camera. Based on the results, advertising displaying system can make decision on playing advertising clips related to certain group of people (in this propose, the advertising clip is chosen for the customer group with largest number of peoples come into the supermarket). We propose a deep learning technique with 2 models YOLOv4 and YOLOv4-tiny for object classification and detection. Our collected data set from the reality with 7000 images under different conditions. Final results show that the YOLOv4 model reach a high accuracy of 91,7%. It also can identify small objects far from camera and in the poor lighting conditions. Meanwhile, the YOLOv4-tiny model has a lower accuracy of 74%. It still has some limitations in object recognition and in different conditions. However, with a smaller model size and when deploy to Jetson Nano embedded system, the YOLOv4-tiny model achieves higher speeds than the YOLOv4 model.

In order to reach the goal of a real-time detection and classification system with high accuracy and fast decision making, the YOLOv4 is better for advertising clip making decision system. Some

limitations in the this propose includes: the data set needs to be more diverse to increase the accuracy of the model in many different conditions. This model also needs to be tested with other deep learning models to get more results. Besides, the hardware configuration can also be considered in order to improving system's ability.

REFERENCES

- Ahmed, I., Din, S., Jeon, G., Piccialli, F. (2020). Exploring Deep Learning Models for Overhead View Multiple Object Detection. In *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5737-5744, doi: 10.1109/JIOT.2019.2951365.
- Bochkovskiy, A., Wang, C., & Liao, H.M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. *ArXiv, abs/2004.10934*.
- Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1251-1258.
- Cortes, C., Vapnik, V. (1995). Support vector machine. *Mach. Learn.*, vol. 20, no. 3, pp. 273-297.
- Dorothy, A. B., Kumar, S. B. R. and Sharmila, J. J. (2017). IoT Based Home Security through Digital Image Processing Algorithms. In *World Congress on Computing and Communication Technologies (WCCCT)*, 2017, pp. 20-23, doi: 10.1109/WCCCT.2016.15.
- Du, S., Zhang, B., Zhang, P., Xiang, P. (2021). An Improved Bounding Box Regression Loss Function Based on CIOU Loss for Multi-scale Object Detection. In *2021 IEEE 2nd International Conference on Pattern Recognition and Machine Learning (PRML)*, pp. 92-98, doi: 10.1109/PRML52754.2021.9520717.
- El-Saadawy, Tantawi, M., Shedeed, H. A., Tolba, M. F. (2021). A Hybrid Two-Stage GNG-Modified VGG Method for Bone X-Rays Classification and Abnormality Detection. In *IEEE Access*, vol. 9, pp. 76649-76661, doi: 10.1109/ACCESS.2021.3081915.
- Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, p. 1440-8.
- He, K., Gkioxari, G., Dollár, P., Girshick, R. (2017). Mask R-CNN. In *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2961-2969.
- He, K., Zhang, X., Ren, S., Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. In *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904-1916.
- Hosseini, H., Masouleh, M. T., Kalhor, A. (2020). Improving the Successful Robotic Grasp Detection Using Convolutional Neural Networks. In *6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, pp. 1-6, doi: 10.1109/ICSPIS51611.2020.9349542.
- Hsieh, Y., Jeng, Y. (2018). Development of Home Intelligent Fall Detection IoT System Based on

- Feedback Optical Flow Convolutional Neural Network. In *IEEE Access*, vol. 6, pp. 6048-6057, doi: 10.1109/ACCESS.2017.2771389.
- Jiang, Z., Zhao, L., Li, S., & Jia, Y. (2020). Real-time object detection method based on improved YOLOv4-tiny. *ArXiv, abs/2011.04244*.
- Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *F. Pereira, C. J. C. Burges, L. Bottou & K. Q. Weinberger (ed.), Advances in Neural Information Processing Systems 25* (pp. 1097--1105) . Curran Associates, Inc.
- Kumar, A., Kalia, A., Sharma, A. et al. (2021). A hybrid tiny YOLO v4-SPP module based improved face mask detection vision system. In *J Ambient Intell Human Comput* (2021). <https://doi.org/10.1007/s12652-021-03541-x>
- Lalitha, V. L., Raju, S. H., Sonti, V. K., Mohan, V. M. (2021). Customized Smart Object Detection: Statistics of detected objects using IoT. In *International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, pp. 1397-1405, doi: 10.1109/ICAIS50930.2021.9395913.
- Lin, T., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S. (2017). Feature Pyramid Networks for Object Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936-944, doi: 10.1109/CVPR.2017.106.
- Lin, T., Goyal, P., Girshick, R., He, K., Dollár, P. (2020). Focal Loss for Dense Object Detection. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318-327, doi: 10.1109/TPAMI.2018.2858826.
- Liu W, Anguelov D, Erhan D, SzegedyC, Reed S, Fu CY, Berg, A. (2016).SSD: single shot MultiBox detector. *arXiv. https://arxiv.org/abs/1512.02325*.
- Liu, S., Qi, L., Qin, H., Shi, J., Jia, J. (2018). Path Aggregation Network for Instance Segmentation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8759-8768, doi: 10.1109/CVPR.2018.00913.
- Nowozin S. (2014). Optimal decisions from probabilistic models: the intersection-over-union case. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp 548–555. <https://doi.org/10.1109/CVPR.2014.7>.
- Rane, S., Dubey, A., Parida, T. (2017). Design of IoT based intelligent parking system using image processing algorithms. In *International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 1049-1053, doi: 10.1109/ICCMC.2017.8282631.
- Redmon J, Divvala S, Girshick R, Farhadi A. (2016). You only look once: unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA 2016, pp 779–788. <https://doi.org/10.1109/CVPR.2016.91>.
- Redmon J, Farhadi A. (2017). YOLO9000: better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp 6517–6525. <https://doi.org/10.1109/CVPR.2017.690>.
- Redmon, J., Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
- Simonyan, K. and Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. In *The 3rd International Conference on Learning Representations (ICLR2015)*. <https://arxiv.org/abs/1409.1556>.
- Srivastava, S., Divekar, A.V., Anilkumar, C. et al. (2021). Comparative analysis of deep learning image detection algorithms. In *J Big Data* 8, 66 <https://doi.org/10.1186/s40537-021-00434-w>
- Uddin, M. I., Alamgir, M. S., Rahman, M. M., Bhuiyan, M. S., Moral, M. A. (2021). AI Traffic Control System Based on Deepstream and IoT Using NVIDIA Jetson Nano. In *2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, pp. 115-119, doi: 10.1109/ICREST51555.2021.9331256.
- Wu, X., Xu, H., Wei, X., Wu, Q., Zhang, W., Han, X. (2020). Damage Identification of Low Emissivity Coating Based on Convolution Neural Network. In *IEEE Access*, vol. 8, pp. 156792-156800, doi: 10.1109/ACCESS.2020.3019484.
- Zhang, Y., Zhao, P., Li, D., Konstantin, K. (2020). Spatial Attention Based Real-Time Object Detection Network for Internet of Things Devices. In *IEEE Access*, vol. 8, pp. 165863-165871, doi: 10.1109/ACCESS.2020.3022645.