Accredited SINTA 2 Ranking
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online at: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 8 No. 2 (2024) 265 - 271 e-ISSN: 2580-0760

YOLO-based Small-scaled Model for On-Shelf Availability in Retail

Arrie Kurniawardhani[1], Dhomas Hatta Fudholi[2]*, Gabriel Imam Andaru[3], Ahmad Azzam Alhanafi[4], Nabil Najmudin[5]
[1,2,3,4,5] Department of Informatics, Universitas Islam Indonesia, Yogyakarta, Indonesia
[1]arrie.kurniawardhani@uii.ac.id, [2]hatta.fudholi@uii.ac.id, [3]gabriel.andaru@students.uii.ac.id,
[4]ahmad.alhanafi@students.uii.ac.id, [5]nabil.najmudin@students.uii.ac.id

Abstract

On-shelf availability (OSA) in the retail industry plays a very crucial role in continuous sales. Product unavailability may lead to a bad impression on customers and reduce sales. The retail industry may continue to develop through the rapidly advancing technology era to thrive in a market where competition is increasingly tough. Along with technological advancements in recent decades, Artificial Intelligence has begun to be applied to support OSA, particularly using object detection technology. In this research, we develop a small-scaled object detection model based on four versions of You Only Look Once (YOLO) algorithm, namely YOLOv5-nano, YOLOv6-nano, YOLOv7-tiny, and YOLOv8-nano. The developed model can be used to support automatic detection of OSA. A small-scale model has developed in the sense of post-practical implementation through low-cost mobile applications. We also use the quantization method to reduce the model size, INT8 and FP16. This small-scaled model implementation also offers implementation flexibility. With a total of 7697 milk-based retail product images and 125 different product classes, the experiment results show that the developed YOLOv8-nano model, with a mAP50 score of 0.933 and an inference time of 13.4 ms, achieved the best performance.

Keywords: YOLO; small-scaled model; on-shelf availability; retail; deep learning

How to Cite: D. H. Fudholi, A. Kurniawardhani, G. I. Andaru, A. A. Alhanafi, and N. Najmudin, "YOLO-based Small-scaled Model for On-Shelf Availability in Retail", *J. RESTI (Rekayasa Sist. Teknol. Inf.)* , vol. 8, no. 2, pp. 265 - 271, Apr. 2024.
DOI: https://doi.org/10.29207/resti.v8i2.5600

1. Introduction

On-shelf availability (OSA) is a significant issue in the retail sales industry that must be addressed alongside the growth of numerous retail outlets and the rapid movement of products [1], [2]. OSA checks the stock-keeping units (SKU) of products or services that are in a store. OSA is critical because it can maintain customer satisfaction and increase sales at the store. When the product is not on the store shelves, the seller cannot sell it [2]. The unavailability of products on store shelves within a certain time or Out-of-Stock (OOS) can create a negative impression from customers and can result in losses, both for retail stores and product manufacturers [3], [4].

A system called 3D Vision-Based Shelf Monitoring (3D-VSM) was built in 2021 to estimate the percentage of OSA and provide warnings if OOS occurs [5]. The system obtains input from the RGB-D camera in the form of 3D data. Surface Fitting and Occupancy Grid techniques are used to compare current shelf photos with reference data. The data is a reference to the condition of the shelf when it is empty or when it is filled. Researchers stated that 3D-VSM can estimate OSA accurately. However, the system cannot yet recognize the type of product.

A quite different approach is taken by Vision Pillar to calculate the availability of products on the shelves [6]. The vision pillar was built using 2 RGB cameras and 2 ToF Depth cameras. Each camera will scan the shelves and take photos of the shelves, section by section. Each section will be combined into one shelf photo and then the product category in the photo will be identified using a Neural Network approach. This combination allows the system to know a product and its position. However, implementing the system may have high costs and are less adaptive.

Artificial Intelligence (AI) is considered a revolutionary technology that can change various aspects of life, society, work, and business [7]. AI implementation is becoming increasingly massive because it can increase

Received: 10-12-2023 | Accepted: 31-03-2024 | Published Online: 25-04-2024

the efficiency of human work, which was previously done manually. The application of AI is benefiting several industries in today's market, including the retail sector [7]. In the retail sector, AI can be implemented to improve On-Shelf Availability (OSA). OSA reflects the number of goods that are available in stores and ready for customers to buy [3], [8].

In 2021, research aimed at increasing the effectiveness and efficiency of product inspection activities using the object recognition concept was carried out [9]. The architecture used is You Only Look Once version 3 (YOLOv3). The study's results are pretty good. For single objects, the accuracy was 90%, and for multiple objects, it was about 80%. The results are less desirable if the object is in a heterogeneous condition.

YOLOv4 is one of the Deep Learning (DL) methods used to detect objects on shelves. In research [10], YOLO is combined with the concept of Semi-Supervised Learning to overcome the small number of annotated datasets. Tests were carried out on four dataset scenarios: (1) only having 20% labelled data, (2) 40% labelled data, (3) 60% labelled data, and (4) 80% labelled data. The test results show that the application can still recognize the product well. In the best scenario, the application has better accuracy than RetinaNet and YOLOv3 [10].

YOLO architecture was also used in OSA problems to produce an algorithm called Hyb-SMPC [11]. This algorithm consists of two modules. The first module is tasked with recognizing products from various categories, and the second module is tasked with checking whether the planogram rules have been fulfilled. The first module uses a one-stage deep learning detector (OSDLD). There are three OSDLD methods compared YOLOv4, YOLOv5, and You Only Learn One Representation (YOLOR). The second module is carried out by comparing the JSON file, which contains the product layout rules that should be arranged, with photos of the shelves after the products are arranged. Test results show that Hyb-SMPC with YOLOv4 can achieve the highest accuracy, above 90%.

Another approach emerged in proposing a model that can estimate product position and product weight distribution on the shelf. Research [12] uses a weight sensor to estimate the weight of the product and Machine Learning to determine the position of the product. Test results show that the implementation of this approach can produce an accuracy of 97% [12].

We have not seen any research development in small-scaled models for hundreds of retail product OSA so far, which can be developed as flexible and low-cost applications. This research develops a model that identifies OSA automatically using Deep Learning's object detection algorithm, YOLO. We focus on the architectures that have the smallest scale for each different YOLO version. In the future, the developed model can be implemented in mobile-based applications. The mobile approach is seen as a low-cost and flexible approach. Both retail shop owners and product manufacturer salesmen can easily use this application to take photos of shelves in a shop to check OSA automatically.

2. Research Methods

In this research, the study was carried out in mainly four stages, as shown in Figure 1. The stages are data gathering, data labeling and splitting, modeling, and evaluation. Each stage is elaborated as follows.

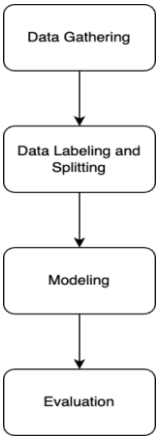

Figure 1. Research Methodology

2.1 Data Gathering

The dataset used is image data. One image consists of more than five objects and more than three categories of products or SKUs arranged on shelves in the shop. Example data is shown in Figure 2. The images for the dataset were taken with a mobile device directly at the local grocery store. The total data collected was 7784 images, for a total of 125 SKUs.

Figure 2. An example of gathered data

Images are taken from different angles, with the most collected viewpoint being perpendicular to the product. Images are also taken at an oblique angle from the right or left side, being careful not to tilt too far to maintain product visibility. Some images are also taken at angles slightly above or below the product to provide alternative viewpoints.

Apart from the viewing angle, lighting is also an important factor in the image dataset. Most of the

pictures in this dataset have relatively similar lighting. The brightness level is like the room lighting conditions. This can be seen in most pictures taken inside retail with the lights on. This constant lighting enables consistency in the visual observation of the products.

2.2 Data Labeling and Splitting

The image is then labelled with the labelling tool [13] by drawing a bounding box around the object and adding an annotation with the SKU name on each object. The information provided by labelling is the two-point coordinates (x, y) of each object, which represent the top-left and bottom-right points of the bounding box.

The results of labelling each object will be written in the format: <label> <x top-left> <y top-left> <x bottom-right> <y bottom-right>. These coordinates determine the location and size of objects in the image. After each image is annotated along with the SKU name, two folders are generated. One folder contains all the annotated images, and another folder contains the labelling and annotation information for each image.

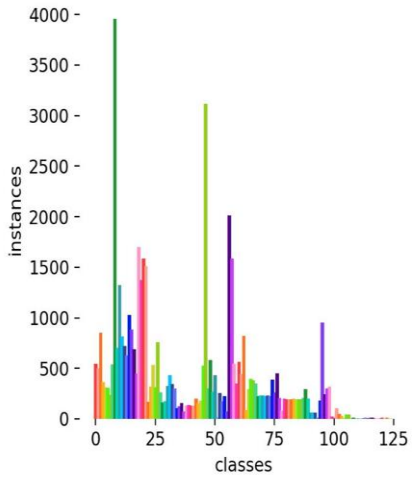

Figure 3. Distribution of training data

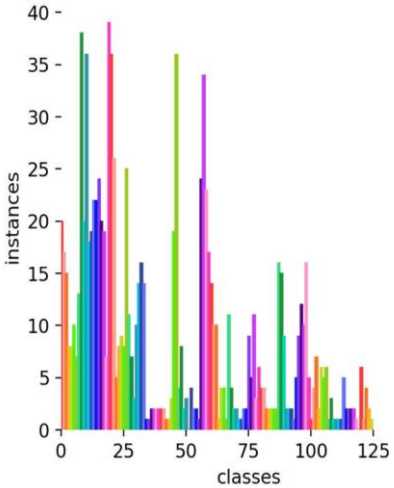

Figure 4. Distribution of testing data

After all the data is labelled, it is divided into two, 7541 images for training and 156 images for testing. To note that, one image can have many annotated products (instance) from different SKUs. Figure 3 shows the distribution of the number of instances per SKU in the training data, while Figure 4 shows the distribution of the number of instances per SKU in the testing data. The imbalanced dataset depicts the real-world case of the different stock numbers of different products. Moreover, certain products with different SKUs have very similar designs (such as the same design but different product grammation). Hence, we tried to add more variation samples.

2.3 Modeling

After the data and environment are prepared, the YOLO algorithm is trained using Transfer Learning. The transfer learning method is implemented using weights that have been previously trained by other researchers on large-scale general datasets. These weights store information on the general characteristics of objects in the dataset. These weights are used as initial weights when we train our model [14]. Transfer Learning can shorten training time compared to creating a new model from scratch because the model already knows the general characteristics of common objects [15]. The initial weights used in this research were taken from the repository in which each model had been trained using the Microsoft COCO dataset [16-23].

You Only Look Once (YOLO) is an object detection algorithm that has been very popular in recent years [24]. YOLO has several versions, starting from its first introduction in 2015, namely YOLOv1, to the latest version released in 2023, namely YOLOv8 [25]. YOLO architecture combines several object detection components into a single neural network [26]. Consistently, YOLO focuses on maintaining a balance between speed and accuracy so that it can be implemented in real-time applications, such as robotics, driverless cars, and video monitoring applications, without having to sacrifice prediction results [25]. YOLO's reliability as an object detection model is characterized by its tiny size and fast computation time.

The four versions of YOLO used in this research are YOLOv5n [16,25], YOLOv6-N [17-18,25], YOLOv7-tiny [20,21,25], and YOLOv8n [22,25]. The scalability version used in each YOLO version is the smallest because this model will later be embedded in mobile devices. At the end of the research, we will compare the most effective and efficient YOLO models used for mobile devices. The following settings are used to execute the YOLO algorithm in this study. The computer specifications used are Ubuntu Linux Operating System, Intel Xeon Gold 6138 2.00GHz CPU, NVIDIA Tesla V100 16GB x 4 VGA, 32 GB RAM. The software used is Python 3.8.16 and Jupyter Hub. The python libraries used include matplotlib>=3.2.2, numpy>=1.22.2, opencv-python>=4.6.0, pillow>=7.1.2, pyyaml>=5.3.1,

requests>=2.23.0, scipy>=1.4.1, torch>=1.8.0, torchvision>=0.9.0, tqdm>=4.64.0.

Hyperparameters are parameters in the YOLO algorithm whose values must be determined at the beginning before carrying out training. Hyperparameter values can affect the performance of model training. Determining the right hyperparameter values can optimize training results. The three hyperparameters set in this research are batch size, number of epochs, and size of the input image.

The batch size chosen is 32. A batch size of 32 is a good starting value. Larger batch sizes can speed up network calculations but will reduce the number of updates required for the network to reach convergence [27]. Apart from being able to record the model's performance at each epoch during training, the library used can also store the weights of the model that has the best performance and the weights of the model in the last epoch. [28]. Epoch with a total of 300 will be used to monitor whether the model built is good enough and has not experienced overfitting or underfitting. The size of the input image used is 1024 x 1024 pixels. This size is the average resolution of the images collected at the data collection stage.

In general, Neural Network (NN) is trained on a 32-bit floating point (FP32) [29]. In a computing context, FP32 refers to the use of 32 bits to store numbers with decimals, which include fractional numbers and exponents. The optimization algorithm is iterative and often achieves very precise values at high resolution. Apart from consuming computational time when processing them, values that are too precise often do not have a significant influence on the classification process [29].

Quantization is the process of reducing the precision of the weights and activations of a Neural Network to a lower bit size, usually 8-bit or less [30]. The goal of Quantization is to reduce size and computation while maintaining accuracy. Quantization can be applied to various layers, such as convolutional layers, normalization layers, and skip connections [31].

The quantization technique allows reducing the NN resolution to INT8 or FP16 by scaling the FP32 weights. This low-precision format provides several performance benefits. (i) Many processors provide faster mathematical computing paths for low-bit formats. This can speed up mathematically intensive operations such as convolution and matrix multiplication. (ii) The smaller word size reduces memory bandwidth pressure and improves performance for limited computing. (iii) Smaller word sizes result in lower memory size requirements, which can improve cache usage as well as other aspects of memory system operation. [32]

Machine Learning models are quantized and converted to run on mobile devices. This process is carried out using the TensorFlow library. As an example, we will use the TensorFlow Lite converter function, optimization function, and interpreter TensorFlow Lite. After that, the detection object model will be in TensorFlow Lite (.tflite) format. After conversion, the model can be implemented on mobile devices, such as smartphones or embedded systems and run locally using the interpreter TensorFlow Lite [33].

2.4 Evaluation

Intersection over Union (IoU) is the most used loss function in object detection. This loss function is used to optimize the object detection model to produce more accurate bounding box predictions. The concept used by IoU is areas intersection and combination of predicted bounding box with ground truth bounding box. IoU loss calculates the ratio between the intersection and the union of those two bounding boxes [34]. The IoU value ranges from 0 to 1. The closer to 1, the more precise the predicted bounding box produced.

The trained model is then evaluated with test data. The metrics used to evaluate model performance are Precision (P), Recall (R), mAP, and inference time. The IoU threshold value is 0.5. In this case, if the IoU between the model inference result and the actual label is more than 50%, then the detection is a TP, otherwise, the detection is an FP. Labels that do not have a matching detection are considered False Negatives (FN).

Precision is the probability of correctly predicting a positive sample from all samples predicted as positive, while Recall is the probability of predicting a positive sample from all true positive samples. The formulas for Precision (P) and Recall (R) are shown in Formula 1 and 2. False Negative (FN) refers to predicting positive samples as negative incorrectly, True Positive (TP) refers to predicting positive samples as positive correctly, and False Positive (FP) refers to predicting negative samples as positive incorrectly [35].

$$R = \frac{TP}{TP+FN} \quad (1)$$

$$P = \frac{TP}{TP+FP} \quad (2)$$

To evaluate object detection, apart from Precision and Recall, Average Precision (AP) and mean Average Precision (mAP) also need to be calculated. The higher the AP score, the higher the model's precision in detecting objects. The mAP value is the average precision score of all product categories detected. The mathematical equations for AP and mAP are shown in Formula 3 and 4, where N is the number of classes.

$$AP = \int_0^1 P(R)dR \quad (3)$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \quad (4)$$

3. Results and Discussions

The experimental results of YOLOv5n, YOLOv6-N, YOLOv7-tiny, and YOLOv8n are shown in Table 1.

For mAP50 and Recall score, YOLOv8n achieve the most optimal score among the other models, 0.933 for mAP50 and 0.827 for Recall. Based on the results obtained, YOLOv8, which is the newest model in this research, can achieve the highest mAP50 score and has a low inference time. On the other hand, YOLOv5, which is the oldest model in this study, has the lowest mAP50 score, but it has the smallest number of parameters among the others.

In terms of precision, YOLOv7-tiny achieve the highest Precision score, 0.863. However, the number of parameters of YOLOv7-tiny reached 37 million, while the other three versions of YOLO are just under 5 million. YOLOv7-tiny takes the longest to infer since has the most parameters compared to other YOLO versions.

Even though it has a relatively large number of parameters, YOLOv6-N only takes around 13.1 ms to detect objects in an image. This makes YOLOv6-N the fastest model in detecting objects compared to the other eleven models. YOLOv8n is only 0.3 ms slower than YOLOv6-N.

Even though the overall model performance obtained is quite good, there are still several classes that get poor prediction results. Figure 5 shows the sample image where the object was successfully detected correctly. Meanwhile, Figure 6 shows the image where the object detection gets some errors. The errors may include undetected products or misclassified products. Most founded misclassified products are products that have similar designs but are different in product gramation. Both images are inferenced using the developed YOLOv8n model.

When attempting to embed a model on a mobile device, the file size of the model is another factor to consider in addition to inference time. Hence, quantization is done to optimize it. Table 2 shows the result details for the quantization model.

From Table 2, we can see that the V8n model, as the latest state-of-the-art architecture, still has the highest mAP and Recall score. The V7-tiny model still also has the highest precision after quantization. However, it is not significantly higher than the other model.

Figure 5. Successful prediction

Figure 6. Predictions with errors

Specifically, in terms of file size, the YOLOv5n that has been converted to tflite with INT8 quantization is the model that has the smallest file size. This is because YOLOv5n has the fewest number of parameters among the three other YOLO versions. Additionally, the model is quantized using INT8.

Table 1. Experiment Results

No	Model YOLO	Layers	Params	mAP50	Precision	Recall	Inference Time
1	V8n	168	3348299	0.933	0.831	0.827	13.4 ms
2	V7-tiny	314	37150628	0.881	0.863	0.818	25 ms
3	V6-N	142	4651467	0.903	0.837	0.787	13.1 ms
4	V5n	157	1928290	0.824	0.769	0.791	14.2 ms

Table 2. Experiment Results in Quantization Model

No	Model YOLO	mAP50	Precision	Recall	File Size (MB)
1a	V8n TFLite (FP 16)	0.932	0.829	0.818	6.91
1b	V8n TFLite (INT 8)	0.925	0.833	0.819	4.98
2a	V7-tiny TFLite (FP 16)	0.861	0.844	0.791	74.8
2b	V7-tiny TFLite (INT8)	0.859	0.839	0.791	38.1
3a	V6-N TFLite (FP 16)	0.896	0.809	0.772	9.5
3b	V6-N TFLite (INT 8)	0.897	0.813	0.778	4.98
4a	V5n TFLite (FP 16)	0.815	0.761	0.783	4.04
4b	V5n TFLite (INT 8)	0.822	0.777	0.772	2.22

Interestingly, in our experiment, there is no noticeable difference in performance metrics (mAP50, Precision, and Recall) between FP 16 and INT 8 quantization. However, using INT 8 as the quantization method gives

quite a lower file size which may impact a faster inference time when implemented in mobile devices. Implementing and further detailed study of this small-scale model in mobile device systems will be done in the future.

4. Conclusions

Automatic identification of On-Shelf Availability (OSA) becomes a significant issue in the retail sales industry as more stores open and products move more quickly. You Only Look Once (YOLO) is an object detection algorithm that has been very popular in recent years. YOLOv5n, YOLOv6-N, YOLOv7-tiny, and YOLOv8n as the small-scaled architecture are used as the base of the objects detection model on the shelf to optimize OSA identification. The scalability version used in each YOLO version is the smallest because this model will later be embedded in mobile devices. Among the three other models, YOLOv8n can achieve the highest mAP50 score and has a low inference time. We also implement the quantization to see the optimization for the model. It is found that using INT 8 can help reduce the file size which may impact the higher performance of the system when later implemented in mobile devices.

Acknowledgements

This research is supported by the Fundamental Regular Research fund scheme from the Directorate General of Higher Education, Research, and Technology (Ditjen Diktiristek), Ministry of Education, Culture, Research, and Technology (Kemendikbudristek), with grant number 0536/E5/PG.02.00/2023.

References

[1] A. S. Villarreal-Navarro, K. V. Vazquez-Preciat, L. Camacho-Alcala, and B. Villarreal, "Increasing on-shelf stock availability of a Mexican convenience store network: A case study.", In Proceedings of the International Conference on Industrial Engineering and Operations Management, pp. 10-16, Jan. 2017.

[2] A. Trautrims, D. B. Grant, J. Fernie, and T. Harrison, "Optimizing on-shelf availability for customer service and profit." Journal of Business Logistics, vol. 30, no. 2, pp. 231-247, Sept. 2009. https://doi.org/10.1002/j.2158-1592.2009.tb00122.x

[3] Y. Ettouzani, N. Yates, and C. Mena, "Examining retail on shelf availability: promotional impact and a call for research." International Journal of Physical Distribution & Logistics Management, vol. 42, no. 3, pp. 213-243, Apr. 2012. doi:10.1108/09600031211225945

[4] M. Ezhilkumar, "Enhancing behavioural intention in out-of-stock situations the mediating role of perceived product uniqueness and perceived consumption risk." Academy of Marketing Studies Journal, vol. 24, no. 2, pp. 1-15, 2020.

[5] A. Milella, R. Marani, A. Petitti, G. Cicirelli, and T. D'Orazio. "3d vision-based shelf monitoring system for intelligent retail." In International Conference on Pattern Recognition, pp. 447-459. Cham: Springer International Publishing, 2021.

[6] M. Crăciunescu, D. Baicu, S. Mocanu, and C. Dobre. "Determining on-shelf availability based on RGB and ToF depth cameras." In 2021 23rd International Conference on Control Systems and Computer Science (CSCS), pp. 243-248. IEEE, Jul. 2021. DOI: 10.1109/CSCS52396.2021.00047

[7] L. Cao, "Artificial intelligence in retail: applications and value creation logics", vol. 49, no. 7, pp. 958-976, Jul. 2021. https://doi.org/10.1108/IJRDM-09-2020-0350

[8] T. Kent, O. Omar, "Retailing in the Economy". Retailing. pp. 3-1, 2003.

[9] M. N. Ardiansyah, P. S. Muttaqin, M. D. Prasetio, N. Novitasari, "Identifikasi Objek/Produk untuk Proses Stock Taking Barang menggunakan Konsep Object Recognition", Jurnal Rekayasa Sistem & Industri (JRSI), vol. 8, no. 1, Jun. 2021. DOI: https://doi.org/10.25124/jrsi.v8i1.455

[10] R. Yilmazer, D. Birant, "Shelf auditing based on image classification using semi-supervised deep learning to increase on-shelf availability in grocery stores." Sensors, vol. 21, no. 2, pp. 327, Jan. 2021. https://doi.org/10.3390/s21020327

[11] M. Saqlain, S. Rubab, M. M. Khan, N. Ali, S. Ali, "Hybrid approach for shelf monitoring and planogram compliance (hyb-smpc) in retails using deep learning and computer vision." Mathematical Problems in Engineering. pp. 1-18, Jun. 2022. https://doi.org/10.1155/2022/4916818

[12] M. H. Lin, M. A. Sarwar, Y. A. Daraghmi, T U. İk, "On-shelf load cell calibration for positioning and weighing assisted by activity detection: Smart store scenario", IEEE Sensors Journal, vol. 22, no.4, pp.3455-3463, Jan 2022. DOI: 10.1109/JSEN.2022.3140356

[13] Tzutalin. "LabelImg". Git code, https://github.com/tzutalin/labelImg, 2015. Accessed: Dec 12, 2023.

[14] A. Hosna, E. Merry, J. Gyalmo, Z. Alom, Z. Aung, and M. A. Azim. "Transfer learning: a friendly introduction." Journal of Big Data, vol. 9, no. 1 pp. 102, Oct 2022. https://doi.org/10.1186/s40537-022-00652-w

[15] M. Christopher, A. Belghith, C. Bowd, et al, "Performance of Deep Learning Architectures and Transfer Learning for Detecting Glaucomatous Optic Neuropathy in Fundus Photographs", Scientific Reports, vol 8, no.1, pp. 16685, Nov 2018. https://doi.org/10.1038/s41598-018-35044-9

[16] G. Jocher, "YOLOv5 by Ultralytics." https://github.com/ultralytics/yolov5, 2020. Accessed: Dec 12, 2023.

[17] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, et al., "Yolov6: A single-stage object detection framework for industrial applications," arXiv preprint arXiv:2209.02976, 2022.

[18] M. Contributors, "YOLOv6 by Meituan Vision AI Department" https://github.com/meituan/YOLOv6, 2023. Accessed: Dec 12, 2023.

[19] C. Li, L. Li, Y. Geng, H. Jiang, M. Cheng, B. Zhang, Z. Ke, X. Xu, X. Chu, "YOLOv6 v3.0: A Full-Scale Reloading", https://docs.ultralytics.com/models/yolov6, arXiv preprint arXiv:2301.05586, 2023. Accessed: Dec 12, 2023.

[20] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," arXiv preprint arXiv:2207.02696, 2022. https://docs.ultralytics.com/models/yolov7

[21] W.K. Yiu, "Official YOLOv7" https://github.com/WongKinYiu/yolov7, 2023. Accessed: Dec 12, 2023.

[22] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8" https://github.com/ultralytics/ultralytics, ultralytics, 2023. Accessed: Dec 12, 2023.

[23] T.Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, "Microsoft coco: Common objects in context", In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, Proceedings, Part V 13 (pp. 740-755). Springer International Publishing, September 6-12, 2014. https://doi.org/10.1007/978-3-319-10602-1_48

[24] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma. "A Review of Yolo Algorithm Developments." Procedia Computer Science, vol. 199, pp. 1066-1073, 2022. https://doi.org/10.1016/j.procs.2022.01.135

[25] J. R. Terven and D. M. Cordova-Esparza. "A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond." arXiv preprint arXiv:2304.00501, Oct 2023.

[26] 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, pp. 779-788. 2016.

[27] Y. Bengio. "Practical recommendations for gradient-based training of deep architectures." In Neural Networks: Tricks of the Trade: Second Edition, pp. 437-478. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. DOI https://doi.org/10.1007/978-3-642-35289-8_26

[28] G. Jocher, "Tips for Best Training Results", https://docs.ultralytics.com/yolov5/tutorials/tips_for_best_training_results, Ultralytics, Nov 2023. Accessed: Dec 12, 2023.

[29] S. C. Magalhães, F. V. Santos, P. Machado, A. Paulo Moreira, J. Dias, "Benchmarking edge computing devices for grape bunches and trunks detection using accelerated object detection single shot multibox deep learning models", Engineering Applications of Artificial Intelligence, vol. 117, Part A, pp. 105604, Jan. 2023. https://doi.org/10.1016/j.engappai.2022.105604

[30] D. Zhang, J. Yang, D. Ye, and G. Hua. "Lq-nets: Learned quantization for highly accurate and compact deep neural networks." In Proceedings of the European conference on computer vision (ECCV), pp. 365-382. 2018. https://arxiv.org/pdf/1807.10029.pdf

[31] P. Chen, J. Liu, B. Zhuang, M. Tan, C. Shen, "AQD: Towards Accurate Quantized Object Detection", arXiv:2007.06919, May 2021. https://arxiv.org/pdf/2007.06919.pdf

[32] H. Wu, P. Judd, X. Zhang, M. Isaev, and P. Micikevicius. "Integer quantization for deep learning inference: Principles and empirical evaluation." arXiv preprint arXiv:2004.09602 Apr 2020. https://doi.org/10.48550/arXiv.2004.09602

[33] R. David, J. Duke, A. Jain, V. J. Reddi, N. Jeffries, J. Li, N. Kreeger et al. "Tensorflow lite micro: Embedded machine learning for tinyml systems." Proceedings of Machine Learning and Systems, vol. 3, pp. 800-811, 2021. https://arxiv.org/pdf/2010.08678.pdf

[34] D. Zhou, J. Fang, X. Song, C. Guan, J. Yin, Y. Dai, and R. Yang. "Iou loss for 2d/3d object detection." In 2019 international conference on 3D vision (3DV), pp. 85-94. IEEE, Sept 2019. DOI: 10.1109/3DV.2019.00019

[35] F. Bashir, and F. Porikli. "Performance evaluation of object detection and tracking systems." In Proceedings 9th IEEE International Workshop on PETS, pp. 7-14, Jun. 2006.

