Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Warehouse Technology Revolution: Integration of Drones and Deep Learning Algorithms for Stock Identification and Calculation Automation

Chandra Hermawan Heruatmadja[1,*], Harjanto Prabowo[1], H. Leslie Hendric Spits Warnar[1], Yaya Heryadi[1]

[1] Doctor of Computer Science Binus University Jakarta, Indonesia

ARTICLE INFO

Keywords:
Deep learning algorithms; drones; inventory management; warehouse automation; YOLO

ABSTRACT

The technological revolution in warehouse management is accelerating, especially with the use of drones and deep learning algorithms to overcome operational challenges. This research aims to develop a deep learning-based automation system that integrates drones for accurate and efficient identification and inventory calculation in retail warehouses. The research method uses the Design Science Research Methodology (DSRM) approach, which includes problem identification, solution development, demonstration, evaluation, and communication. The dataset contains more than 4,000 images of annotated goods and is used to train YOLO, Mask R-CNN, and RetinaNet models. The evaluation was conducted using Precision, Recall, and Average Precision (mAP) metrics. The test results show that YOLO has the best performance with an average mAP of 0.5 of 0.978 and a processing time of only 4.82 seconds, far superior to Mask R-CNN and RetinaNet. In addition, drone integration allows for efficient imaging of goods, reduces the risk of work accidents, and improves inventory accuracy by up to 100%. The results of this study highlight the importantance of combining drone technology and deep learning algorithms to create reliable solutions in inventory management, especially in dynamic retail environments. This research makes a significant contribution to the optimization of logistics processes and warehouse operations, as well as being the basis for further development in the industry.

1. Introduction

In the highly competitive business world, inventory management is a vital component that ensures the smooth operation, efficiency, sustainability, and growth of a company [1,2]. Optimal inventory management aims to maintain a balance between the demand for and availability of goods. Without this balance, a company may face losses either due to excess stock, which burdens the budget, or stock shortages, which lead to missed sales opportunities. These challenges necessitate effective management strategies to enhance customer satisfaction [3,4].

* *Corresponding author.*
E-mail address: chandra.heruatmadja@binus.ac.id

https://doi.org/10.37934/araset.64.4.7490

Inventory management generally involves three main stages: receipt of goods, storage, and distribution. The receipt process includes checking the condition of the goods and recording initial data [5]. Once received, the goods are placed on storage shelves and later distributed to stores or end customers. However, in practice, significant challenges arise, particularly in warehouses handling large and diverse quantities of goods. For instance, one of Indonesia's largest retail industries manages over 4,000 types of goods across 34 warehouses that supply 19,000 stores. Due to labor and time constraints, only 60% of the total goods can be counted daily. This limitation results in inaccuracies in stock data, increasing operational risks, such as work accidents involving equipment like forklifts [6].

Technology has become a critical solution to address these challenges. Systems such as Radio Frequency Identifier (RFID) improve stock calculation accuracy and support better decision-making. However, the maintenance costs of RFID systems remain a barrier to widespread adoption. Recent technological advancements, such as machine learning, offer significant potential in inventory management. Machine learning-based models can predict demand and optimize inventory with high accuracy. One prominent technique is object detection using the Convolutional Neural Networks (CNN) algorithm, which has demonstrated an accuracy of 98.94% in classifying goods in previous studies [7,8]. However, speed limitations pose a challenge to its practical implementation.

Other research highlights that deep learning-based object identification techniques can be applied to a variety of needs. For instance, Egi *et al.,* (2022) utilized this technique to count tomatoes in open fields with an accuracy of up to 99%. Nonetheless, further testing is required in closed environments, such as warehouses. As object identification technology evolves, various models continue to be refined to enhance both accuracy and speed. From traditional methods to the era of deep learning, this technology has been applied across diverse fields [9,10]. This progress presents significant opportunities to improve operational efficiency in the logistics and retail sectors.

This research focuses on developing a deep learning-based object identification model to calculate inventory in retail warehouses. Using devices such as drones to capture images of goods, this technology aims to overcome labor and time constraints while reducing work-related risks. The proposed model offers an efficient solution for retail companies to process inventory data more quickly and accurately, support improved operational strategies, and enhance overall logistics efficiency.

2. Methodology

2.1 Proposed Research Framework

This study uses the Design Science Research Methodology (DSRM) framework to develop practical solutions with a large impact on business practices. DSRM was chosen because it integrates scientific research with practical, relevant applications to improve the operational efficiency of the organization. This framework includes the stages of problem identification, solution development, demonstration, evaluation, and communication of results [1-3]. The artifacts produced serve as innovations that support the improvement of industrial performance. Based on the DSRM method developed by Dewi *et al.,* [4], Ramirez *et al.,* [5], and Juneja *et al.,* [6], the study is prepared with systematic steps that are further described in the stages of the research framework described in the Figure 1.

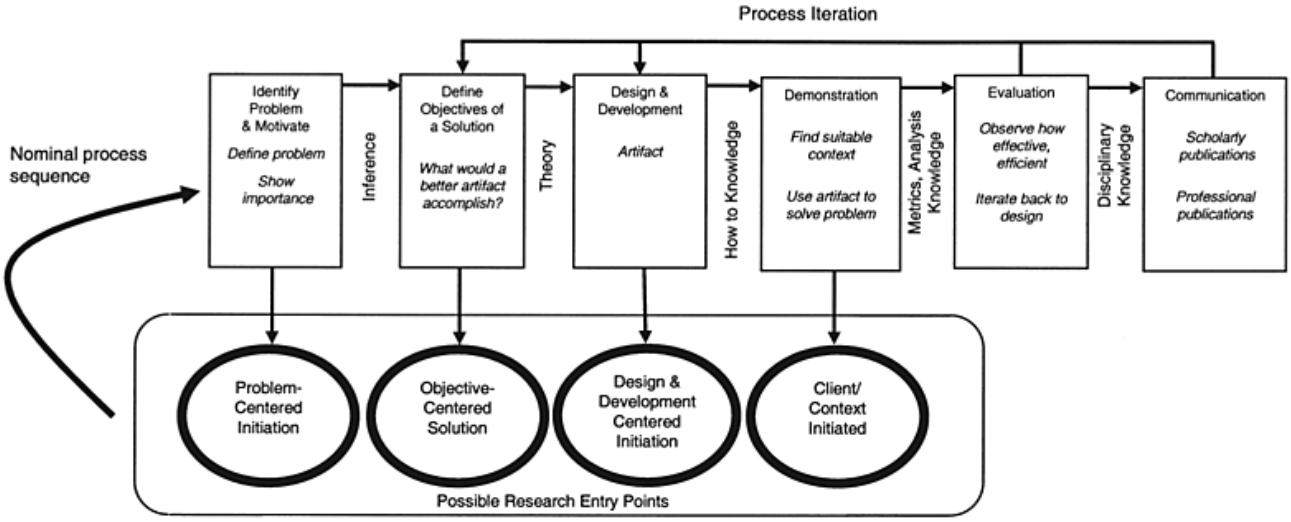


Fig. 1. DSRM-based framework

The DSRM approach allows research to start from various stages, such as problem identification, design and development, or demonstration, depending on the need. This research started from the identification of inventory counting system problems in retail, which motivated the development of solution models to improve operational efficiency and accuracy.

2.2 Research Stages

In order to carry out research based on the DSRM thinking framework, the stages of research can be described as follows:

2.2.1 Problem identification and motivation

The data collection stage involves Focus Group Discussions (FGD) with major retail industry players in Indonesia, including Logistics Directors, IT Directors, Logistics General Managers, and IT teams, to identify inventory calculation problems in large warehouses. FGD results in research gaps and research questions. In addition, images of goods from warehouses and simulations are collected as datasets for deep learning models [7]. These datasets are annotated using the MS-COCO format, which supports segmentation of objects with polygon markers. The dataset is divided into training and testing data to support the design and development stages, ensuring the model can accurately recognize objects under various conditions. The annotation was performed using the MS-COCO toolkit (version 1.4.0) on Ubuntu 20.04 LTS, and data augmentation (flipping, scaling, rotation) was applied using the Albumentations library (version 1.3.0), ensuring reproducibility with a fixed random seed of 42 (see Figure 2).

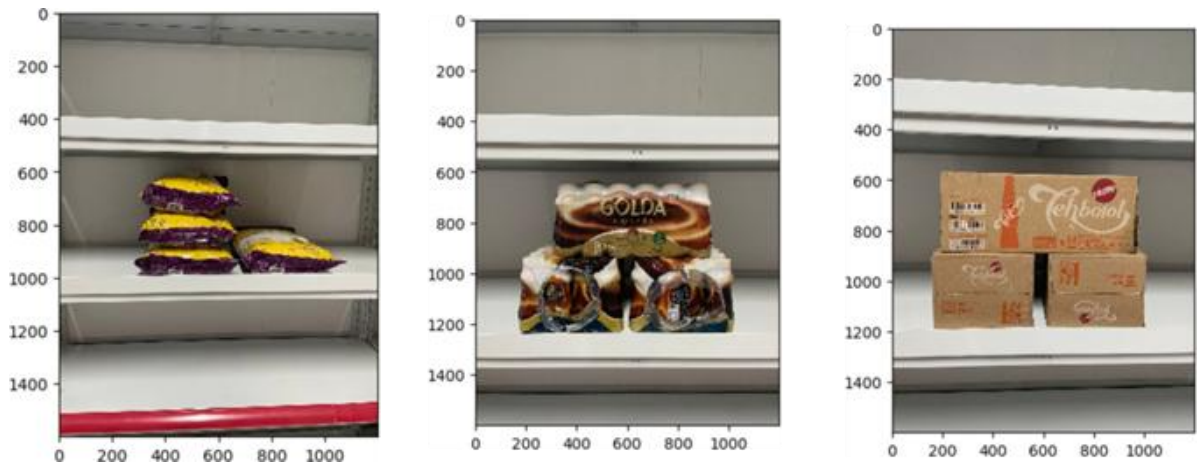


Fig. 2. Variety of objects for the training and testing process

In addition to dataset descriptions, detailed annotation processes were performed using MS-COCO toolkit version 1.4.0 on Ubuntu 20.04 LTS, ensuring consistent formatting. Data augmentation involved transformations like horizontal flipping, scaling, and rotation using the Albumentations library version 1.3.0. The exact random seed used was set to 42 to ensure reproducibility.

2.2.2 Defining the purpose of the solution

The purpose of the study is to develop a solution for accurate and thorough inventory calculation (100%) without high dependence on human resources, thereby reducing the risk of work accidents. A literature review of previous research was conducted to identify relevant deep learning solutions and models, resulting in several model options to answer the first research question (RQ1) related to technology-based inventory calculations.

2.2.3 Design and development

The initial stage of design and development is carried out through a literature review to determine relevant deep learning models, such as RetinaNet, Fast R-CNN, Mask-RNN, and YOLO. The selection of the model was based on an evaluation of accuracy and speed using the Precision, Accuracy, Recall, and F1-Score metrics. As a result, the best model was selected based on optimal performance in recognizing cardboard, plastic, and sack packaging in the collected dataset. The model was implemented using PyTorch version 2.0.1 and trained on an NVIDIA RTX 4080 GPU with CUDA 11.8. Key hyperparameters included:

i. Learning Rate: 0.001
ii. Weight Decay: 0.0005
iii. Momentum: 0.9
iv. Optimizer: Stochastic Gradient Descent (SGD)

The model's configuration file (yolov8-nano.yaml) and training checkpoints were tracked using MLflow. The model was trained with 80% of the dataset, and 20% was used for testing to ensure accuracy. Data augmentation techniques were applied to make the model robust against various real-world conditions, and results from the model were evaluated for recognition accuracy on goods such as cardboard, plastic, and sack packaging, as shown in Figure 3.

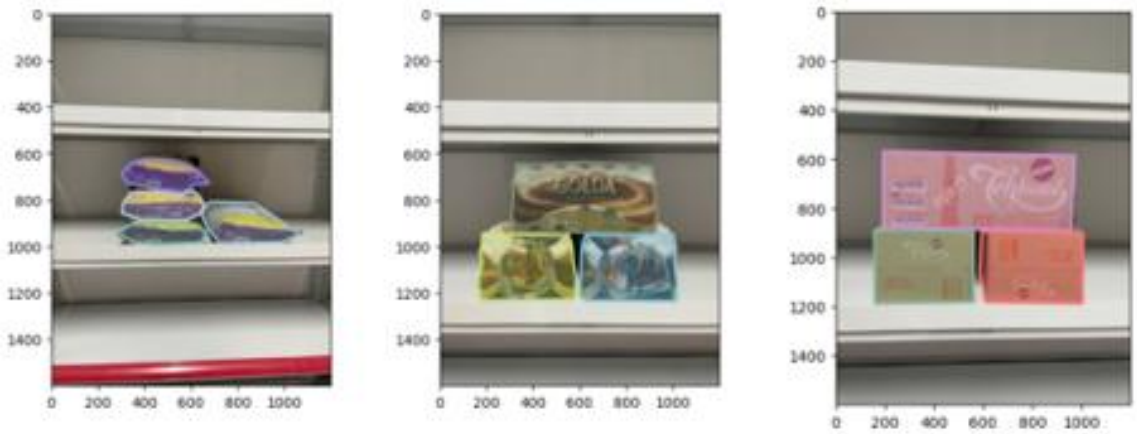


Fig. 3. Variety of objects with segmentation notation

The YOLO model was implemented using PyTorch version 2.0.1 and trained on the NVIDIA RTX 4080 GPU with CUDA 11.8. Hyperparameters included a learning rate of 0.001, weight decay of 0.0005, and momentum of 0.9. Optimizer: SGD. Exact model configuration files (yolov8-nano.yaml) and checkpoints were logged using the MLflow tracking system (see Figure 4).

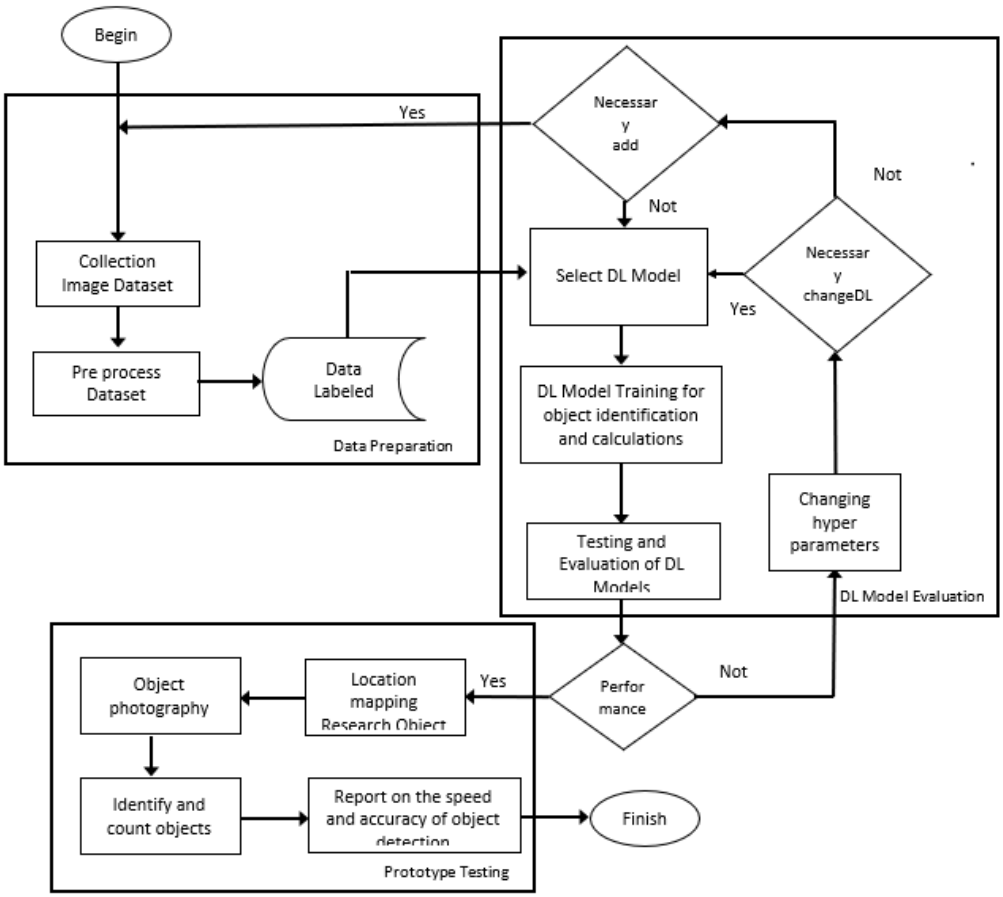


Fig. 4. Proposed development methods

The design and development stage of the research includes five steps: model training using 80% of the dataset to generate the model at optimal speed, model testing using 20% of the dataset to ensure accuracy, mapping the location of objects for drone maneuvering, designing UML-based process flows, and developing prototypes as real solutions [8,9]. This process aims to solve inventory

calculation problems with an efficient and accurate deep learning-based solution as shown in Figure 5.

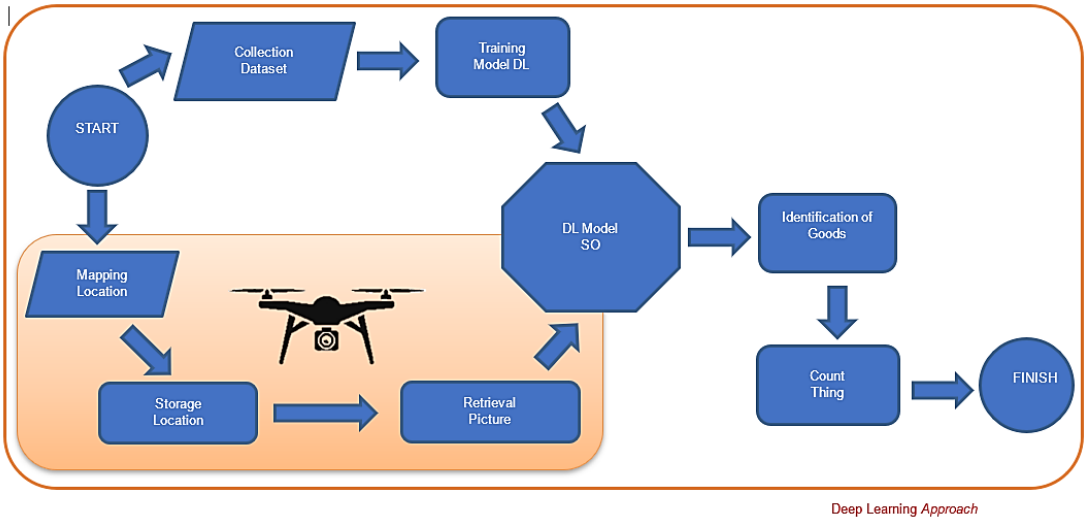


Fig. 5. Research contribution to the inventory calculation process

2.2.4 Demonstration

This stage tests a prototype deep learning model in a warehouse simulation that replicates real conditions. Testing involves the movement of drones, taking images, identifying objects, and calculating inventories using real products. Resources such as hardware, software, and guides are prepared to ensure smooth operation [10]. The results of the demonstration are documented, including scenarios, methods, and test results, in order to evaluate the effectiveness of the model and answer the second research question (RQ2).

2.2.5 Evaluation

The evaluation stage compares the objectives of the solution with the test results through two stages: first, measuring the accuracy of the model using a measurement matrix; Second, assess the suitability of the function based on acceptance tests and user feedback with the black box and white box methods. The results of the evaluation determine whether the model meets the objectives of the solution or requires redevelopment [11]. If appropriate, the research proceeds to the communication stage, documenting the findings, and providing development suggestions for subsequent research. The evaluation phase consisted of two parts:

i. Performance Evaluation: The accuracy of the model was assessed using performance matrices like Precision, Recall, and F1-Score.
ii. Suitability Evaluation: The function of the model was evaluated based on user feedback through acceptance testing, using both black-box and white-box testing methodologies. The results informed whether the model met the intended objectives or required further refinement.

2.2.6 Communication

The last stage of using the DSRM methodology is the communication stage, where the results of the stages that have been passed, starting from problem identification, solution proposals, design and development, demonstration and submission, are outlined in the research results report [12]. The report of the research results will be carried out in the form of academic publications in the form of 1 conference article and 2 journal articles by selecting reputable conferences and journals to share novelties, methods, and research findings with other researchers and get feedback that can improve the quality of the research.

2.3 Input Process Output (IPO)

In the explanation above, it has been conveyed how the stages of research to be carried out, the proposed model, to the demonstration and communication stage based on the DSRM approach, and to further clarify how the work system of each stage can produce an output that is in harmony with all existing stages, and the output produced becomes an input (input) needed for the next stage [13], can be presented in the Figure 6 below.

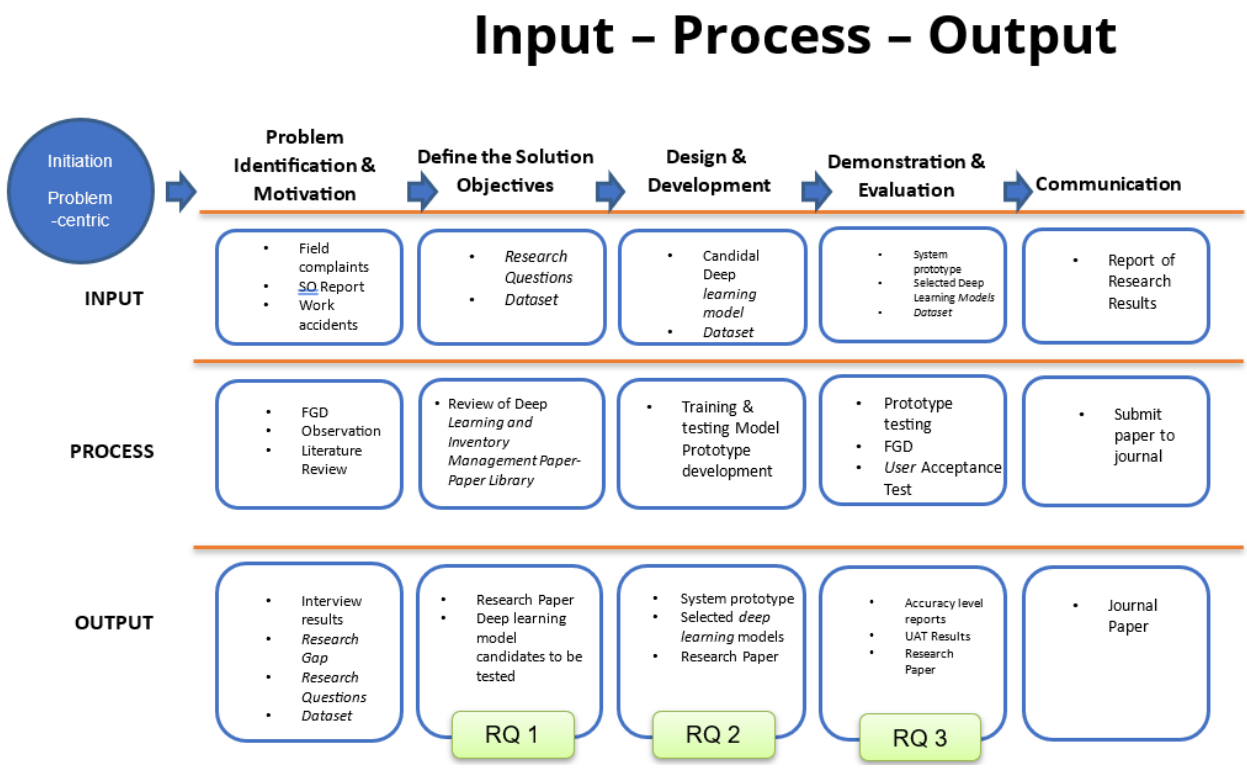


Fig. 6. DSRM-based research stage model IPO

3. Results

This to obtain research results that can answer the problems that have been formulated in the research question, a series of research stages are carried out based on the DSRM framework as described in Chapter 3. As the first stage of the entire research stage based on DSRM, is the identification of problems and motivations, with the results of the research at this stage having been outlined in the background and formulation of the problem, while the second and subsequent stages can be explained as follows.

3.1. Defining Goals and Solutions

This research aims to answer three main questions with a literature review-based solution. Automated inventory counting using object identification, as studied by Alburshaid and Mangoud [14], and Sadaiyandi *et al.,* [15], has been shown to improve time efficiency, maintain accuracy, and reduce labor requirements. Deep learning models, such as YOLOv5, are identified as an ideal solution for inventory management due to their efficiency and ability to accurately classify objects in Figure 7.

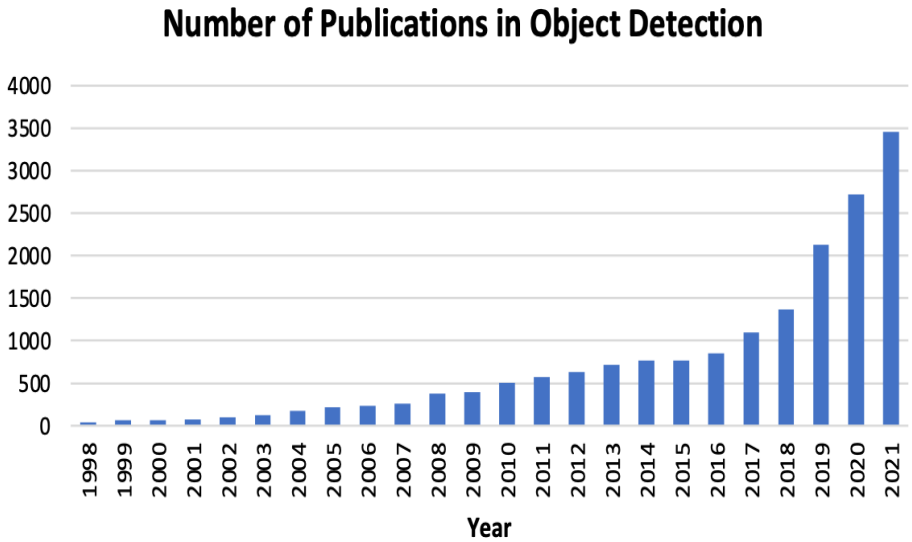


Fig. 7. Development of object identification research deep learning

This study resulted in a systematic literature review of deep learning models for object identification in inventory management. YOLO, Mask R-CNN, and RetinaNet were identified as the most widely used models. This study was presented at the ICICoS 2024 conference. Further research was conducted to test all three models to determine the best model based on speed and accuracy, which will be used in prototyping inventory management in retail warehouses that can be described, as following subsection.

3.1.1 Dataset collection

Dataset collection is an important step in training deep learning models to recognize objects accurately. The dataset includes 4,332 images from real retail warehouses, consisting of cartons (3,359), sacks (79), and plastics (500), and reflects a diversity of lighting conditions, layouts, and packaging variations (see Figure 8).

Fig. 8. Example of an image with 2 types of objects

The dataset was divided into three subsets: 80% for training (3,464 images) to train the model to recognize patterns, 10% for validation (434 images) to evaluate the model's performance during training, and 10% for testing (434 images) to measure the model's generalization ability on new data.

3.1.2 Data pre-processing

Data pre-processing is an important step after dataset collection to prepare the data to be relevant and as required by the deep learning model, ensuring optimal performance. This step includes overcoming dataset imbalances by cleaning up the data, maintaining only the relevant classes and annotations focus on the objects in the center of the image. In addition, the annotation format is tailored to the needs of models such as YOLO (.txt with bounding box coordinates), Mask R-CNN (VGG with segmentation), and RetinaNet (COCO). Conversions are performed to ensure the dataset is compatible with the specifications of each model. This step ensures the quality and structure of the dataset as per the training needs as shown in the following Figure 9.

COCO

{
"info":
{
"description": "Converted from VIA format", "url": "",
"version": "1.0", "year": 2024, "contributor": "",
"date_created": "2024-11-07 14:06:36"
},
"licenses":
[{"id": 1, "name": "Unknown", "url": ""}],
"images":
[{"id": 1, "file_name": "20230929_GOPR0145_000000.jpg",
"width": 1080, "height": 1080, "license": 1, "flickr_url": "",
"coco_url": "", "date_captured": "2024-11-07 14:06:36"}],
"annotations": [{"id": 1, "image_id": 1, "category_id": 0,
"segmentation": [[534.0, 1054.7, 534.8, 1020.4, 537.3, 966.2, 686.4,
967.2, 679.0, 1032.0, 676.4, 1056.2, 611.8, 1056.2, 568.2, 1055.5]],
"area": 13013.435030516237,
"bbox": [534.0, 966.2, 152.39999999999998, 90.0], "iscrowd": 0}]

YOLO

0 0.49444444444444446 0.9765740740740741 0.4951851851851851
0.9448148148148148 0.49749999999999994 0.8946296296296297
0.6355555555555555 0.8955555555555555 0.6287037037037037
0.9555555555555556 0.6262962962962962 0.977962962962963
0.5664814814814815 0.977962962962963 0.5261111111111112
0.9773148148148149

VGG

{"20230929_GOPR0145_000000.jpg141699": {"filename":
"20230929_GOPR0145_000000.jpg", "size": 141699, "regions":
[{"shape_attributes": {"name": "polygon", "all_points_x": [534.0,
534.8, 537.3, 686.4, 679.0, 676.4, 611.8, 568.2], "all_points_y":
[1054.7, 1020.4, 966.2, 967.2, 1032.0, 1056.2, 1056.2, 1055.5],
"bbox": [534.0, 966.2, 152.39999999999998, 90.0]},
"region_attributes": {"type": "0"}}

Fig. 9. Example of dataset annotation format required for the training process

Data augmentation techniques include changes in brightness, contrast, rotation, blur, and resize to improve the generalization of the model, using the Albumentations library. Normalization is performed by changing the pixel value of the image from a scale of 0-255 to 0-1, ensuring the stability and speed of the model during deep learning training. The formula is shown in Eq. (1).

$$Pixel_{norm} = \frac{Pixel}{255} \tag{1}$$

Normalization improves model training efficiency by adjusting image pixel values. The image size is adjusted to 640x640 pixels to meet the needs of models such as YOLO, Mask R-CNN, and RetinaNet, maintaining consistency without sacrificing detail or memory. Stratified sampling ensures a balanced distribution of datasets across all subsets (training, validation, testing), prevents bias, and improves model generalization [16].

3.1.3 YOLO test results

The YOLO (nano) deep learning model, developed by Ultralytics, was chosen for this study because it supports segmentation and is efficient on hardware with limited resources. The training was conducted using an Intel Core i7-13700K CPU, NVIDIA RTX 4080 GPU (16GB), 64GB RAM, and 2TB storage. Library Albumentations makes it easy to augment data during training. With a batch size of 16, 3,464 training images result in 216 iterations per epoch, enabling computational efficiency and memory management when handling large datasets. YOLO ensures faster convergence balance during model training (see Figure 10).

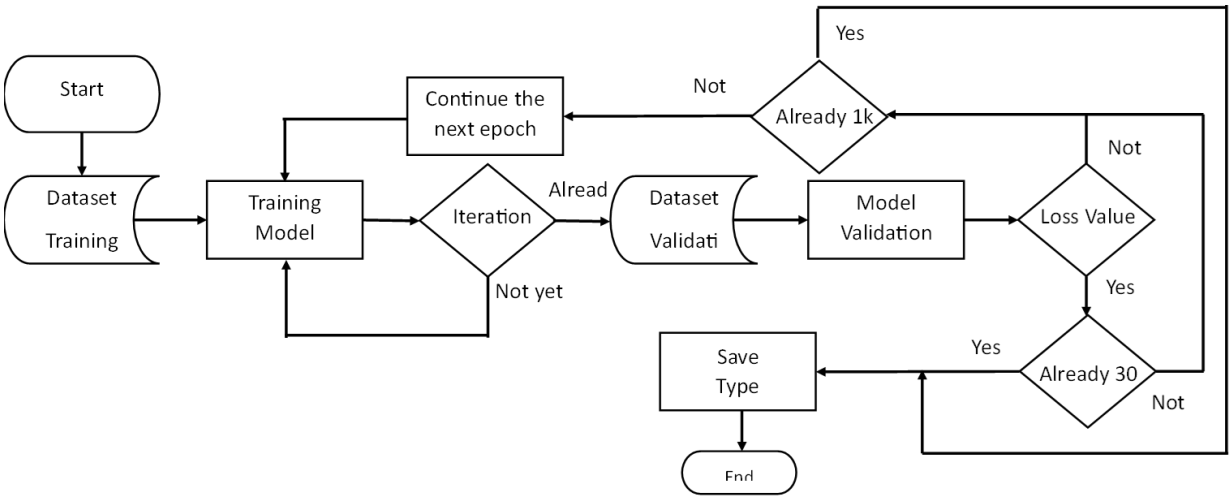


Fig. 10. Flow of the YOLO training process

YOLO's superior performance can be attributed to several intrinsic features that differentiate it from Mask R-CNN and RetinaNet. YOLO employs a unified detection framework that performs object localization and classification in a single network pass, significantly reducing inference time. In contrast, Mask R-CNN utilizes a two-stage process, first generating region proposals and then performing classification and segmentation, which increases computational overhead. RetinaNet, while efficient, relies on the focal loss function to address class imbalance, which can lead to slower convergence compared to YOLO's optimized anchor-free approach.

Moreover, YOLO's grid-based detection mechanism enables faster and more precise bounding box predictions by directly predicting object locations and confidence scores. YOLO's lightweight

architecture (nano variant) is specifically optimized for resource-constrained environments, achieving high accuracy without sacrificing speed. On the other hand, Mask R-CNN requires substantial memory for segmentation masks, and RetinaNet's reliance on multiple feature pyramid levels adds latency in real-time applications [17]. Model configurations are as follows:

i. Batch size: 16
ii. Learning rate schedule: cosine annealing with a warm-up phase of 10 epochs
iii. Number of classes: 3 (cardboard, plastic, sack)
iv. Image resolution: 640x640 pixels
v. Validation split: 20% of the dataset
vi. Random seed for shuffling: 12345

The training process leveraged checkpoint saving every 50 epochs and logging using TensorBoard for visual monitoring.

The YOLO training process is limited to 1,000 epochs for efficiency and preventing overfitting. The early stopping technique is applied to stop training early if the model's performance on the validation data does not improve after 30 epochs, a common parameter to prevent overtraining. Early stopping monitors training loss to optimize training time and model performance (see Figure 11).

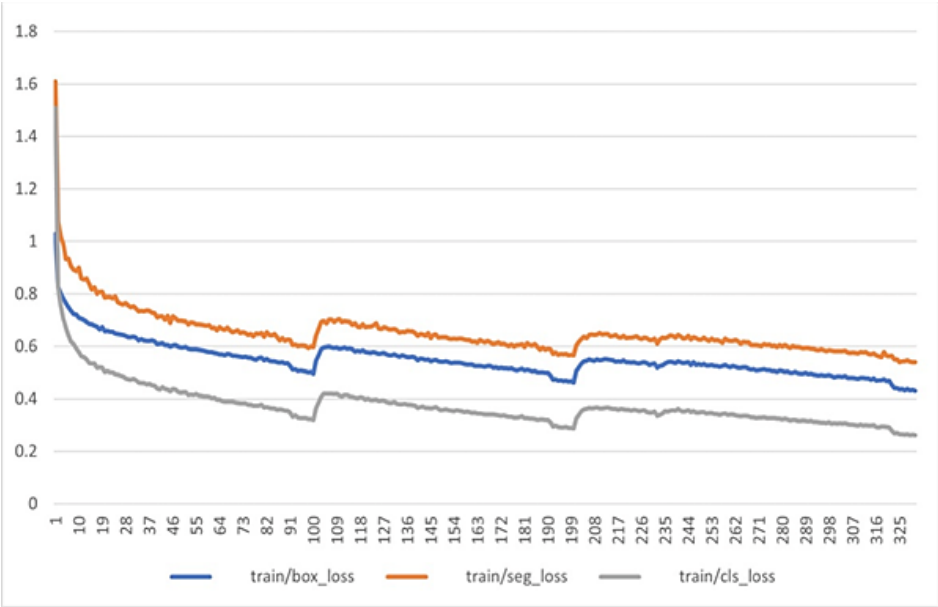
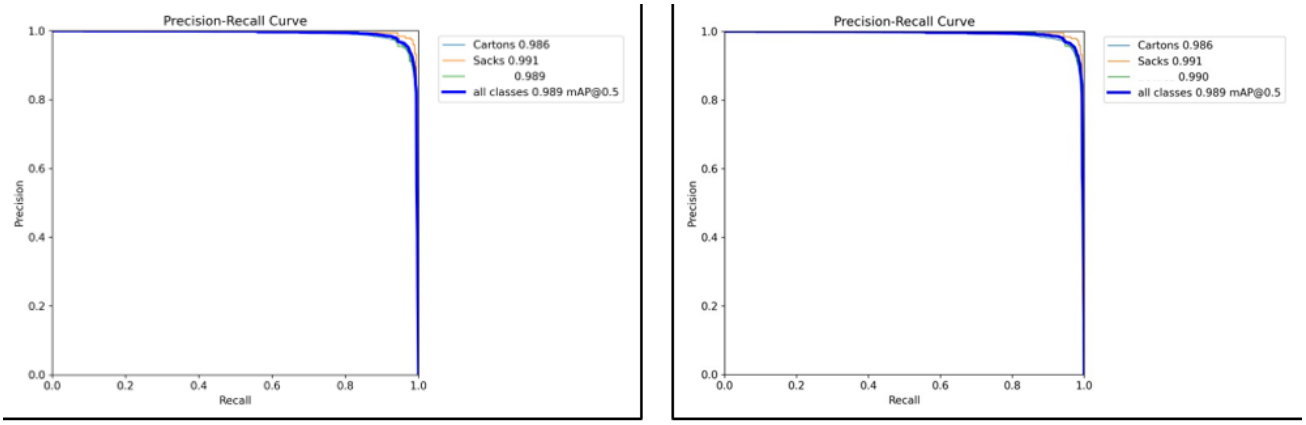


Fig. 11. Loss graph of the YOLO training process

YOLO's training loss graph shows a high initial value (>0.4) that decreases with training, signaling an increase in object identification capabilities. The training process was stopped in the 331st epoch after convergence was reached. The model was then validated to measure the accuracy and speed of identification using Precision, Recall, and Average Precision metrics as shown in Figure 12.



(a) (b)

Fig. 12. Precision vs YOLO recall chart (a) Image for bounding box (b) Image for segmentation

The Precision vs Recall graph of YOLO validation shows the model's ability to identify objects using bounding boxes and segmentation. A precision value close to 1 indicates an accurate prediction, while a recall value indicates the detection of almost any object with very low False Positives. The model managed to recognize three categories (cardboard, sack, plastic) with an average accuracy (mAP@0.5) of 0.978. The training process was completed in 1 hour and 30 minutes (331 epochs), validation in 4.79 seconds, and testing in 4.82 seconds. Compared to Mask R-CNN and RetinaNet, YOLO exhibited a threefold speed improvement in training and testing phases while maintaining comparable or superior accuracy, making it ideal for real-time applications. The model shows high speed and accuracy, ready to use for testing datasets.

3.1.4 R-CNN mask test results

R-CNN Mask testing is conducted using the Detectron2 framework, based on PyTorch, to detect objects and segmentation. The R-CNN Mask model is a two-stage detection with three main blocks: the ResNet50-FPN backbone for feature extraction, the Region Proposal Network (RPN) for object field selection, and the Region of Interest (ROI) Head for prediction. ResNet50 was chosen because it is efficient and accurate, lighter than ResNet101. The same dataset as YOLO is used in the training, validation, and testing process. The R-CNN training mask requires 25,000 epochs with a duration of about 2 hours, but the Detectron2 framework does not support early stopping, unlike YOLO (see Figure 13).

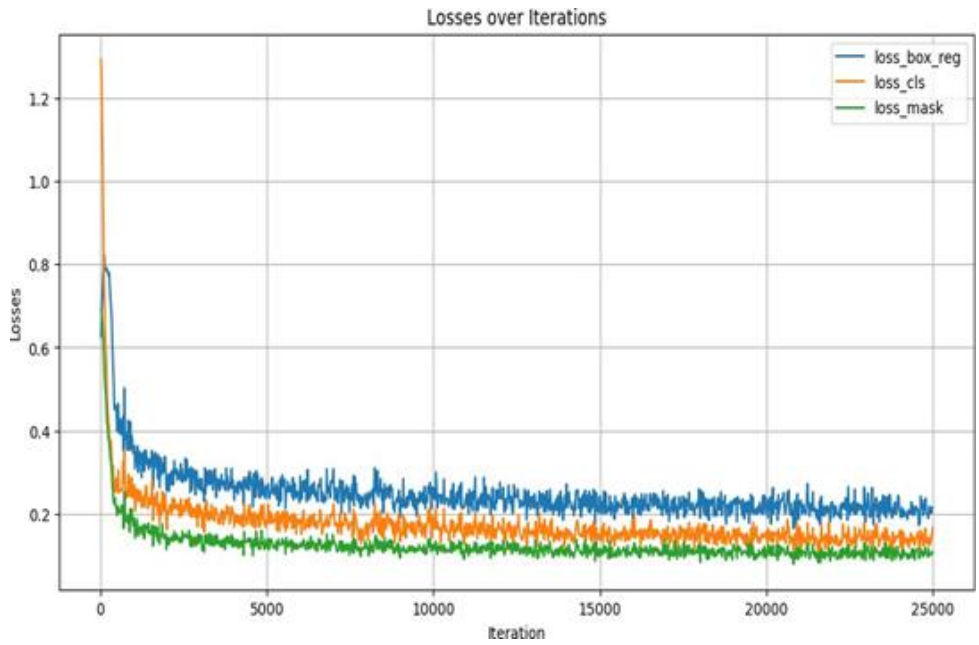


Fig. 13. Loss graph of the *R-CNN Mask* training process

Using a total epoch of 25,000, based on the graph, it appears that the value of the loss of training results in both bounding boxes, segmentation, and classification, shows that there is no significant change after the epoch to 15,000, which indicates that the model has reached the convergence point as shown in Figure 14.

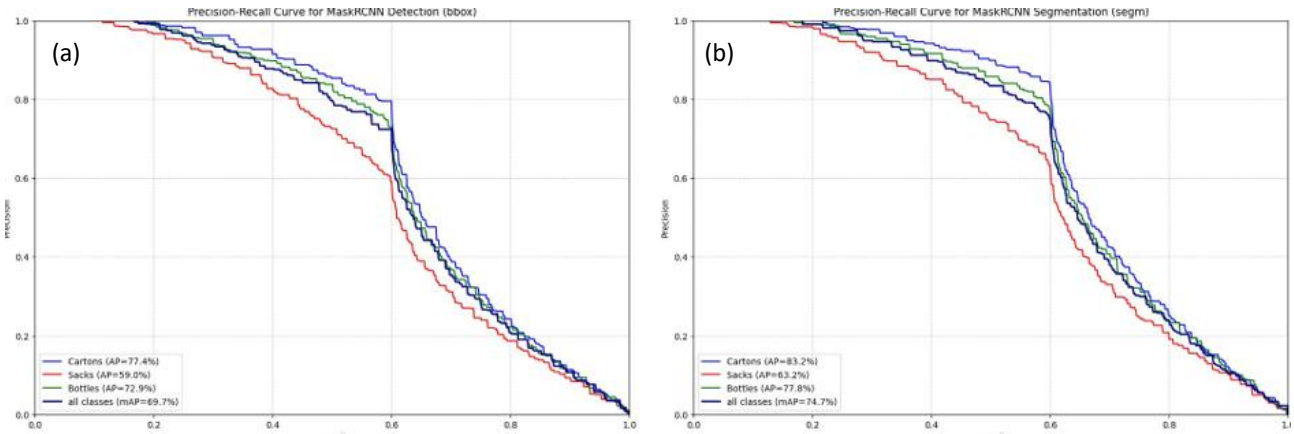


Fig. 14. Precision vs Recall Mask R-CNN Chart (a) Image for bounding box (b) Image for segmentation

After training, the validation of the R-CNN Mask showed high precision (close to 1) on low recalls, but the precision decreased as the recall increased due to more predictions and increased False Positives. The performance in the sak class is lower than other classes. The test results with an IoU threshold of 0.5 produced accuracy: carton 0.814, sack 0.555, plastic 0.768, and an average mAP@0.5 of 0.712. The training was completed in 2 hours and 20 minutes, validation was 1 minute 5 seconds, and testing was 1 minute 7 seconds.

To gain deeper insights, an error analysis was conducted to identify the types of errors encountered by the R-CNN Mask model. False Positives were prevalent in the sack class, which could be attributed to similarities in texture with the cardboard class, leading to misclassification. Conversely, False Negatives were higher in detecting small or partially occluded plastic objects, suggesting limitations in the model's feature extraction capabilities [18]. These findings highlight that

while R-CNN Mask excels in overall accuracy, it struggles with intricate object details, particularly under occlusion or class overlap.

3.1.5 Retina net test results

Retina Net testing was conducted using PyTorch and PyTorch Lightning for ease of modularity. The model has three main blocks: Backbone (ResNet-34), Feature Pyramid Network (FPN), and Head, but it does not have a Region Proposal Network (RPN), making it a one-stage model like YOLO. Retina Net only supports object detection without segmentation, making it the last choice among the three models tested. Testing was carried out for 4,500 epochs with a batch size of 16, requiring 216 iterations per epoch. The learning rate of 0.0005 is used for better stability and accuracy than the standard value of 0.001, helping the model converge optimally (see Figure 15).

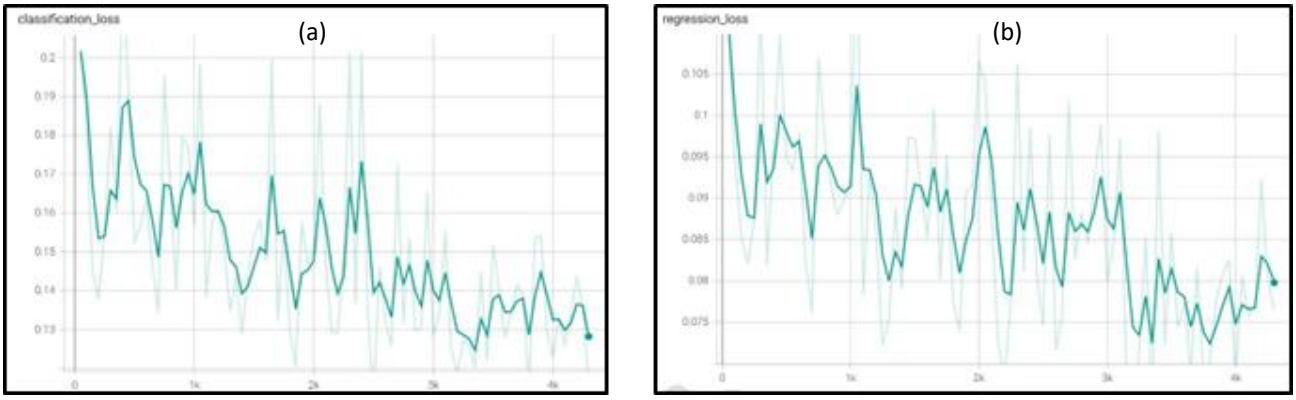


Fig. 15. Loss graph of the RetinaNet training process (a) Image shows the loss of classification (b) Image shows the loss of the bounding box

After the training process, a graph of Loss values was obtained as seen in Figure 15. The figure shows that, even though the loss value from the beginning to the end has decreased, as expected from the results of the deep learning model training process, the resulting loss value is not consistent so it is difficult to get a convergence point [19]. This will result in low prediction quality from deep learning models.

RetinaNet validation results show low precision, especially in the cardboard class, while the sack class has the best performance with stable to medium recall as shown in Figure 16. The plastic prediction shows a decrease in precision when the recall > 0.5. In the test with an IoU threshold of 0.5, the accuracy was recorded: carton 0.421, sack 0.578, plastic 0.410, with an average mAP@0.5 of 0.469. The training was completed in 2 hours and 50 minutes, validation took 25 seconds, and testing took 27 seconds.

Error analysis of RetinaNet revealed that False Positives were frequent in the cardboard class, likely due to the model's limited ability to distinguish between similar shapes. False Negatives were notably higher in detecting plastic objects, particularly those overlapping with other objects. This indicates that RetinaNet's feature pyramid architecture, while efficient, may lack the fine-grained resolution needed for precise classification and localization in complex scenes. These errors underscore RetinaNet's trade-off between computational efficiency and prediction quality.

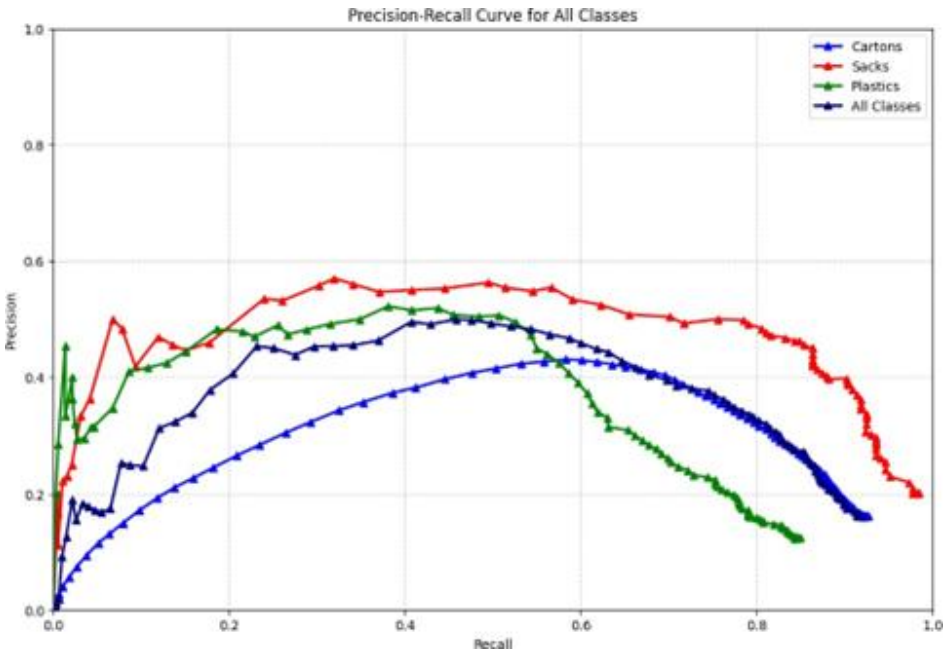


Fig. 16. Precision vs RetinaNet recall chart, available for bounding box only

3.1.6 Final results and answer research questions

In testing the three deep learning models using the same dataset, YOLO showed better training results compared to the other two models, namely Mask R-CNN and RetinaNet. YOLO only took 1 hour and 30 minutes to complete the training, much faster than Mask R-CNN which took 2 hours and 20 minutes and RetinaNet which took 2 hours and 50 minutes. This indicates that the YOLO architecture is lighter and designed for computing efficiency without sacrificing significant performance [20-22].

In the evaluation of accuracy and completion speed using the same test dataset, YOLO again excelled. This model achieved a mAP@0.5 value of 0.978 with a completion time of only 4.82 seconds, far surpassing the Mask R-CNN which had a mAP@0.5 of 0.712 with a time of 1 minute 7 seconds, and RetinaNet with a mAP@0.5 of 0.410 which took 27 seconds. With these results, YOLO shows superiority not only in terms of time efficiency but also prediction accuracy, making it an ideal choice for fast, and even real-time identification.

RetinaNet, on the other hand, also has significant limitations compared to YOLO and Mask R-CNN. RetinaNet only supports object detection using bounding boxes, without supporting object segmentation, which is a key feature of R-CNN and YOLO Masks. YOLO and Mask R-CNN enable object segmentation, which is important for research where pixel-level detail is required in the presence of a variety of possible shapes of staple items that are stacked and have boundaries between crowded items [23]. This puts YOLO as a more versatile and superior solution in scenarios that require fast detection as well as segmentation. Thus, based on the formulation of the problem of what deep learning model is most suitable to be used in the deep learning-based object identification model for inventory calculation in retail warehouses of groceries, based on testing of three deep learning models, it can be concluded that the most suitable choice is to use the deep learning model YOLO.

4. Conclusions

This research addresses the technological revolution in warehouse management by integrating drones and deep learning algorithms for automated stock identification and calculation. Using the

Design Science Research Methodology (DSRM) approach, this study successfully developed a deep learning-based model to overcome operational challenges in retail warehouses, particularly in object identification and inventory calculation. The dataset used consists of over 4,000 annotated images of goods, and three primary deep learning models were tested: YOLO, Mask R-CNN, and RetinaNet. The test results indicate that YOLO performed the best, achieving an mAP@0.5 score of 0.978 with a processing time of 4.82 seconds. This is significantly superior to Mask R-CNN, which had an mAP@0.5 score of 0.712 with processing time exceeding one minute, and RetinaNet, which achieved an mAP@0.5 score of 0.469. The integration of drones enabled more efficient image capture, reduced the risk of workplace accidents, and improved inventory accuracy by up to 100%. Additionally, YOLO demonstrated advantages in time efficiency and prediction accuracy, making it an ideal solution for fast and real-time object identification. This study not only presents innovations in logistics and warehouse operational processes but also highlights the significant potential of deep learning technology in enhancing industrial efficiency. These findings can serve as a foundation for further developments, including algorithm optimization and broader-scale applications. The research also makes academic contributions through the publication of results in reputable conferences and journals. The success of this model paves the way for the wider adoption of technology in inventory management, supporting better operational strategies and enhancing overall logistics efficiency.

Acknowledgement
This research was not funded by any grant.

References

[1] Sukmawati, Wati, Adlawan, Hendely A, Amirullah, Gufron, Yatri, Ika, and Wahjusaputri, Sintha. "Development of Learning Media in Science Based-Solar : An Analysis Using the Many-Facet Rasch Model." *Journal of Advance Research in Applied Science and Engineering Technology* 1, no. 1 (2026): 23–37.

[2] Sukmawati, Wati, Asep Kadarohman, O. M. A. Y. Sumarna, and W. A. H. Y. U. Sopandi. "The relationship of basic chemical concepts in pharmaceutical learning." *Journal of Engineering Science and Technology* 16 (2021).

[3] Sukmawati, Wati, Asep Kadarohman, Omay Sumarna, and Wahyu Sopandi. "Analysis of reduction of COD (Chemical Oxygen Demand) levels in tofu waste using activated sludge method." *Moroccan Journal of Chemistry* 9, no. 2 (2021): J-Chem.6

[4] Dewi, Christine, Rung-Ching Chen, Xiaoyi Jiang, and Hui Yu. "Deep convolutional neural network for enhancing traffic sign recognition developed on Yolo V4." *Multimedia Tools and Applications* 81, no. 26 (2022): 37821-37845. https://doi.org/10.1007/s11042-022-12962-5

[5] Ramirez, Ivan, Alfredo Cuesta-Infante, Juan J. Pantrigo, Antonio S. Montemayor, José Luis Moreno, Valvanera Alonso, Gema Anguita, and Luciano Palombarani. "Convolutional neural networks for computer vision-based detection and recognition of dumpsters." *Neural Computing and Applications* 32, no. 17 (2020): 13203-13211. https://doi.org/10.1007/s00521-018-3390-8

[6] Juneja, Abhinav, Sapna Juneja, Aparna Soneja, and Sourav Jain. "Real time object detection using CNN based single shot detector model." *Journal of Information Technology Management* 13, no. 1 (2021): 62-80.

[7] Shorten, Connor, and Taghi M. Khoshgoftaar. "A survey on image data augmentation for deep learning." *Journal of big data* 6, no. 1 (2019): 1-48. https://doi.org/10.1186/s40537-019-0197-0

[8] Chen, Jun, Shu-Lin Wang, and Hong-Li Lin. "Out-of-stock detection based on deep learning." In *Intelligent Computing Theories and Application: 15th International Conference, ICIC 2019, Nanchang, China, August 3–6, 2019, Proceedings, Part I 15*, pp. 228-237. Springer International Publishing, 2019. https://doi.org/10.1007/978-3-030-26763-6_22

[9] Fan, Wu, Zhuoqun Xu, Huanghe Liu, and Zhu Zongwei. "Machine Learning Agricultural Application Based on the Secure Edge Computing Platform." In *International Conference on Machine Learning for Cyber Security*, pp. 206-220. Cham: Springer International Publishing, 2020. https://doi.org/10.1007/978-3-030-62223-7_18

[10] Piaskowski, Karol, and Dominik Belter. "Fast Object Detector Based on Convolutional Neural Networks." In *Computational Modeling of Objects Presented in Images. Fundamentals, Methods, and Applications: 6th International Conference, CompIMAGE 2018, Cracow, Poland, July 2–5, 2018, Revised Selected Papers 6*, pp. 173-185. Springer International Publishing, 2019. https://doi.org/10.1007/978-3-030-20805-9_15

[11] Beloiu, Mirela, Lucca Heinzmann, Nataliia Rehush, Arthur Gessler, and Verena C. Griess. "Individual tree-crown detection and species identification in heterogeneous forests using aerial RGB imagery and deep learning." *Remote Sensing* 15, no. 5 (2023): 1463. https://doi.org/10.3390/rs15051463

[12] Keßler, René, Christian Melching, Ralph Goehrs, and Jorge Marx Gómez. "Using Camera-Drones and Artificial Intelligence to Automate Warehouse Inventory." In *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering*. 2021.

[13] Corneli, Alessandra, Berardo Naticchia, M. Vaccarini, Frédéric Bosché, and Alessandro Carbonari. "Training of YOLO neural network for the detection of fire emergency assets." In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, vol. 37, pp. 836-843. IAARC Publications, 2020. https://doi.org/10.22260/ISARC2020/0115

[14] Alburshaid, Ebrahim Ali, and Mohab A. Mangoud. "Palm trees detection using the integration between GIS and deep learning." In *2021 International Symposium on Networks, Computers and Communications (ISNCC)*, pp. 1-6. IEEE, 2021. https://doi.org/10.1109/ISNCC52172.2021.9615721

[15] Sadaiyandi, Jeyabharathy, Padmapriya Arumugam, Arun Kumar Sangaiah, and Chao Zhang. "Stratified sampling-based deep learning approach to increase prediction accuracy of unbalanced dataset." *Electronics* 12, no. 21 (2023): 4423. https://doi.org/10.3390/electronics12214423

[16] Nabil, Mohammed, Mohamed Helmy Megahed, and Mohamed Hassan Abdel Azeem. "Design and simulation of new one-time pad (OTP) stream cipher encryption algorithm." *Journal of Advanced Research in Computing and Applications* 10, no. 1 (2018): 16-23.

[17] Mahmoud Ahmed, Moustafa Abdelrahman, Idris, Syahril Anuar, Md Yunus, Nur Arzilawati, Mohd Ali, Fazlina, and Sofian, Hazrina. "CyberShield Framework: Attacks and Defense Modelling for Cybersecurity in Internet of Things (IoT)." *International Journal of Advance Research in Computational Thinking and Data Science* 1, no. 1 (2024): 30-39.

[18] Zakaria, Fathiah, Musirin, Ismail, Aminudin, Norziana, Johari, Dalina, Shaaya, Sharifah Azwa, Ismail, Nor Laili, and Kumar, A.V. Senthil. "Enhancing Power System Resilience Through Optimized Load-Shedding Strategies." *Journal of Electronic System Engineering* 1, no. 1 (2024): 1-9.

[19] Herrera-Toranzo, Piero, Juan Castro-Rivera, and Willy Ugarte. "Detection and Verification of the Status of Products Using YOLOv5." In *ICSBT*, pp. 83-93. 2023. https://doi.org/10.5220/0012123500003552

[20] Kartika, Dhian Satria Yudha, and Hendra Maulana. "Preprosesing dan normalisasi pada dataset kupu-kupu untuk ekstraksi fitur warna, bentuk dan tekstur." *Journal of Computer Electronic and Telecommunication* 1, no. 2 (2020). https://doi.org/10.52435/complete.v1i2.76

[21] Wang, Yongsheng, Duanli Yang, Hui Chen, Lianzeng Wang, and Yuan Gao. "Pig counting algorithm based on improved yolov5n model with multiscene and fewer number of parameters." *Animals* 13, no. 21 (2023): 3411. https://doi.org/10.3390/ani13213411

[22] Li, Michael, and Wei Yao. "3D map system for tree monitoring in hong kong using google street view imagery and deep learning." *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 3 (2020): 765-772. https://doi.org/10.5194/isprs-annals-V-3-2020-765-2020

[23] Abang Shakawi, Abang Mohammad Hudzaifah, and Shabri, Ani. "Adaptability of Statistical and Deep Learning Models to Volatile Market Conditions in Bursa Malaysia Stock Index Forecasting." vol. 1, no. 1, pp. 1–13, 2024. *Semarak International Journal of Machine Learning* 1, no. 1 (2024): 1-13. https://doi.org/10.5194/isprs-annals-V-3-2020-765-2020