Jurnal Teknologi Informasi & Komunikasi

Volume 16 Issue 1 Year 2025 | Page 1-11 | e-ISSN: 2477-3255 | ISSN: 2086-4884
Received: 04-03-2024 | Revised: 14-04-2025 | Accepted: 11-05-2025

Improvement of FPS and Efficiency of Parameters Mask R-CNN with MobileNetV3 Small for Cardboard Detection

Vikha Tri Vicika[1], Jamaludin Indra[2], Sutan Faisal [3], Hanny Hikmayanti [4]
[1,2,3,4] Faculty of Computer Science
[1,2,3,4]Jl. Ronggowaluyo, Karawang 41311, Jawa Barat, Telp. 0267-8403140
*Corespondence: if21.vikhavicika@mhs.ubpkarawang.ac.id

Abstract: *Inventory management in warehouses often experiences discrepancies in recording the number of cardboard boxes due to errors during the manual recording process. To overcome this problem, a cardboard detection method was developed using the Default Mask R-CNN model and a modified model using MobileNetV3 Small. The training data was obtained from a collection of cardboard photos which then went through an annotation stage. In the cReonfiguration stage, various anchor scales were applied to determine the bounding box parameters, while the training process used Stochastic Gradient Descent (SGD). The default model is trained with the initial Mask R-CNN settings, while the custom model modifies the backbone and Feature Pyramid Network (FPN) adjustments. The test results show that the custom model has higher efficiency with a parameter count of 20,857,704 and an average FPS of 10.92. However, the accuracy level of the custom model is lower than that of the default model*

Keywords: *Mask R-CNN, MobileNetV3 Small, Cardboard Detection, Model Configuration*

1. Introduction

Inventory management is a critical component of warehouse operations, directly impacting the smoothness of business operations and growth. This management system encompasses several key procedures, starting from checking inventory availability to recording the movement of goods in and out of the warehouse [1]. One crucial stage is when cardboard boxes are transferred from the transport vehicle (truck) to the storage location within the warehouse. Discrepancies between actual data and system data often occur, leading to financial losses and eroding customer trust. The root cause of this issue lies in manual recording, as workers may be fatigued during calculations [2].

To address this issue, technology capable of providing accurate and consistent results is needed. One solution is the use of a computer system that can detect objects through images, photos, or videos (known as computer vision) [3]. By utilizing images, photos, and videos (digital imagery), this technology can automatically recognize objects [4], enabling the counting of cardboard boxes to be done quickly and accurately. Additionally, this can reduce the risk of errors commonly encountered when manual recording is performed [5].

Although computer vision systems assist in inventory management, there are still several technical challenges that need to be addressed. One particularly challenging issue is the similarity in appearance among cardboard boxes, whether in terms of dimensions, color, or packaging design, which makes the detection process difficult [6]. Furthermore, the presence of repetitive texture patterns on the surface of the cardboard can confuse the system, especially when the cardboard is stacked. This can lead to errors in counting or even disrupt the predictions made by the model [7].

https://doi.org/10.31849/digitalzone.v16i1. 19680

One of the computer vision technologies is Mask R-CNN. Mask R-CNN is a framework designed to perform object segmentation by instance and is a development of Faster R-CNN [8]. Some research on Mask R-CNN, namely Research by [9] shows that Mask R-CNN with ResNet-50-FPN as the backbone has high performance in keypoint detection with an AP keypoint of 75.76 on real data. In addition, research by [10] shows that Mask R-CNN with ResNeXt backbone produces an mAP of 62.62% for object detection and 57.58% for segmentation. However, this study emphasises that the model requires high computing measured in the number of FLOPs, making it suboptimal for devices with limited resources. Then Research by [11] compared Mask R-CNN with ResNet-50, ResNet-101 and MobileNet-V1 backbones. As a result, ResNet-101 achieved the highest mAP of 98.3%. Furthermore, research by [12] shows that replacing the ResNet101 backbone in Mask R-CNN with MobileNetV2 significantly improves model efficiency while maintaining high detection accuracy. The lightweight architecture of MobileNetV2 reduces computational complexity and accelerates inference speed, achieving 0.525 seconds per image on the internal dataset and 0.467 seconds per image on the open dataset, making it suitable for applications requiring real-time performance and limited computing resources. Finally, another study by [13] developed the Yolov5 algorithm for lightweight real-time detection by replacing the backbone with MobileNetV3, which is able to increase the accuracy of small object detection while reducing parameters by up to 87.4%.

MobileNetV3 is an advancement of MobileNetV1 and MobileNetV2, with several new features added to improve network efficiency without compromising accuracy [14]. In various tests, MobileNetV3 has proven to be superior in maintaining a balance between performance and efficiency compared to other architectures such as ResNet18, SqueezeNet, EfficientNetV2-S, and ShuffleNetV2 [15]. MobileNetV3 has two versions: MobileNetV3-Large and MobileNetV3-Small. For implementation on devices with limited resources, MobileNetV3-Small is a more suitable choice as it is specifically designed for such scenarios, offering a 6.6% improvement in accuracy compared to MobileNetV2 despite similar latency [16].

This study will use Mask R-CNN by replacing the backbone with MobileNetV3 to increase the Frame Per Second (FPS) and reduce the parameters so that it is lighter. This study aims to optimise the Mask R-CNN so that it is lighter and can be used in real-time on devices with limited resources.

2. Method

In this study, there are several stages that are carried out systematically. These stages can be seen more clearly in Figure 1.

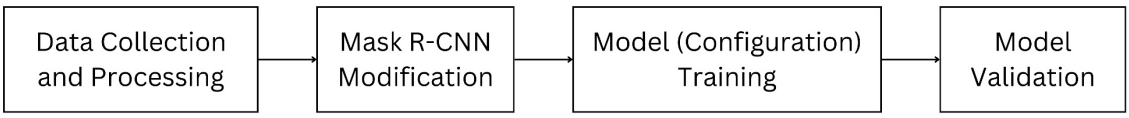


Figure 1. Research Methods

2.1. Data Collection and Data Management

The data was collected at the Anugerah Store Water Distribution Warehouse located in Kedung Lotong Village, Bantarjaya Village, Bekasi Regency. Each photo was taken from a different angle, including front, side, top and bottom to help the model understand the characteristics of the cardboard box. Varying lighting conditions were also arranged to simulate a real environment.

Figure 2. Taking Cardboard Images

The photos were taken using a smartphone camera with 48 megapixel specifications. The shooting distance was 20-40 cm for the single cardboard box shown in Figure 2 image A and 2-3 metres for the cardboard boxes carried by employees in image B. The lighting conditions were set with a 15-watt room light for indoor cardboard shooting and natural lighting around 8 am for cardboard boxes carried by employees. Overall, the data obtained consisted of 639 photos for each product, namely Sky, Fatqua, Levios, and Elvios.

After the shooting process, the next stage is to annotate the dataset to highlight the objects to be detected by the model. Labelling each image is done using the tools available on the Roboflow website [17]. The Roboflow platform is used to upload the collected dataset and the polygon tool is used to perform manual annotation to produce segmentation. Polygon annotation is used to manually mark objects by determining a number of points that form a layer around the object [18]. The annotated categories are sky, fatqua, levios and elvios. The annotation process ensures that each cardboard object is accurately labelled according to its respective product. In Roboflow, in the final stage, the dataset is divided into two, namely for training and validation with a ratio of 88% for training and 12% for validation. As well as adding three times the augmentation available in Roboflow, namely rotation, brightness and blur. Table 1 shows the number of instances for training and validation in each category.

Table 1. Number of datasets and instance distribution

No	Category	Number of Training	Number of Validations
1	Elvios	672	59
2	Sky	717	58
3	Fatqua	735	64
4	Levios	668	55
Total		2792	236

2.2. Modification of Mask R-CNN

Figure 3 shows the modified Mask R-CNN Architecture with the MobileNetV3 Small backbone integrated with the Feature Pyramid Network (FPN) to extract each feature on the cardboard at various scales. This architectural process begins by extracting features from the input image through a backbone that uses several convolutions and Inverted residual blocks. In stages 1 and 2, an extra P2 is added to help improve the representation of features on a small scale, so that small objects can be detected more accurately. Each feature generated by the backbone is forwarded to the FPN, so that it can be processed at various scales with a combination of 1×1 and 3×3 convolutions to improve feature mapping. Then the features from the FPN will be sent to the Region Proposal Network (RPN) to produce potential candidates for new object regions through anchor boxes. These region candidates are then processed on ROI Heads, where further extraction is performed using ROI Align. After that, the extracted features are processed by Box Head to perform object classification and bounding box regression, the goal of which is to determine the location and size of the object accurately. Meanwhile, Mask Head aims to produce object segmentation in the form of a mask.

Figure 4 shows the feature extraction process in the MobileNetV3 Small backbone which is used to detect and recognise objects with high efficiency. The workflow starts from a 224 × 224 × 3 image input, then it is processed through a series of convolution stages and Inverted Residual blocks to produce a more compressed and information-enriched feature representation. Stage 1 reduces the dimensions to 112 × 112 × 16, followed by Stage 2 which produces 56 × 56 × 24 using the Inverted Residual Block to increase processing efficiency. Next, Stage 3 reduces the size to 28 × 28 × 40, followed by Stage 4 with a size of 14 × 14 × 48, which enriches feature information while maintaining object detail. Finally, Stage 5 produces a feature size of 7 × 7 ×

96, which is highly compressed but contains high-level information that will be used in the Feature Pyramid Network (FPN) to improve object detection and mask segmentation.

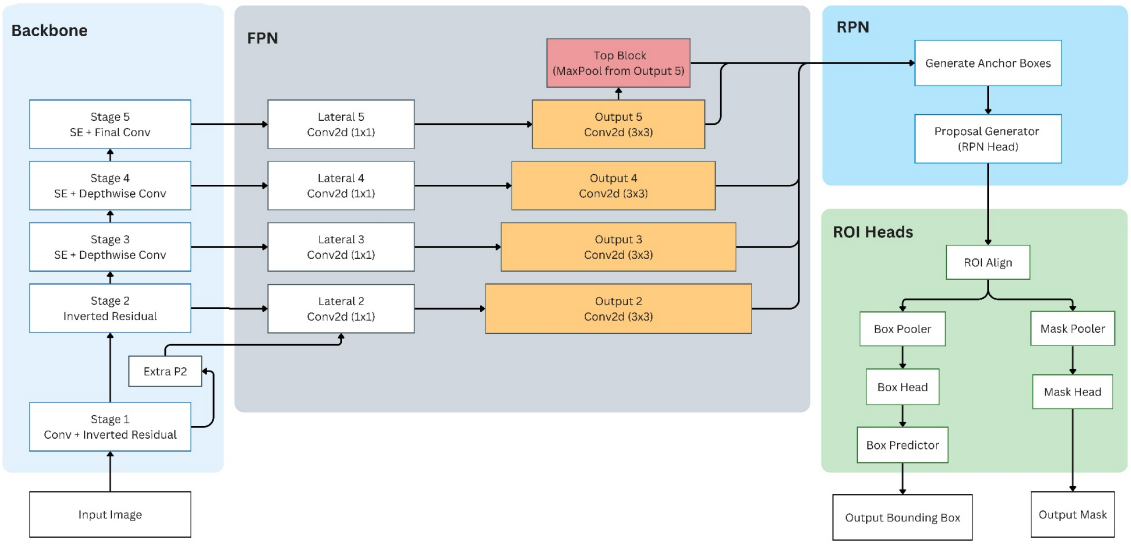


Figure 3. Mask R-CNN + MobileNetV3 Small Architecture

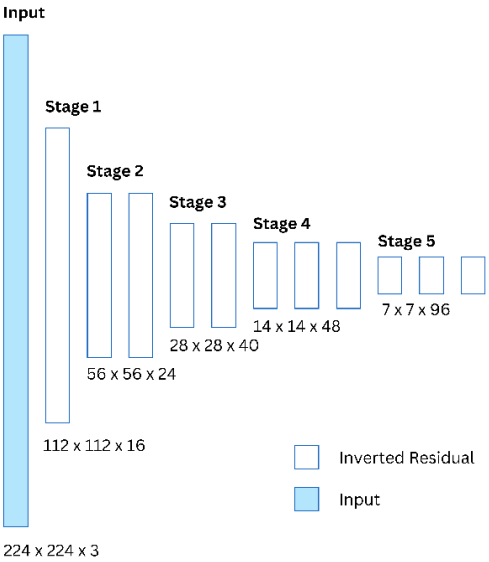


Figure 4. MobileNetV3 Small Backbone

2.3. Model Training

The model training process was carried out using the Google Colab platform with T4 GPU support. As an optimization tool, the Stochastic Gradient Descent (SGD) method was used, which is an approach that updates model parameters based on random data samples to make the model learning process faster and more efficient [19]. In this setup, the SGD momentum value is set to 0.9 and the weight decay to 1e-4, aiming to prevent the model from overfitting—a condition where the model becomes too familiar with the training data and fails to perform well on new data.

The initial learning rate starts at 0.002. During training, the learning rate is reduced periodically, specifically at iterations 12,000 and 16,000, with a reduction factor of 0.1. To enhance model stability at the beginning of training, a warm-up phase of 1,000 iterations is

applied, where the learning rate starts at a very small value (1/1000 of the initial value) and gradually increases.

Each time the model performs one training process (one iteration), only two images are processed at a time (batch size = 2). In the ROI Heads section, each image provides 128 ROI samples for use in training. The training process continues until a total of 20,000 iterations are reached, with the configuration designed to allow the model to achieve its best results.

In the custom model developed, the threshold value used to filter detection results is 0.6, meaning only predictions with a confidence level above 60% are considered valid.

In the architecture used, the anchor generation system is configured to handle various object sizes by applying different scales at each feature level. At level p2, the anchor is 16 pixels in size, level p3 is 32 pixels, level p4 is 64 pixels, level p5 is 128 pixels and level p6 is 256 pixels. This setting aims to ensure that the model can detect cardboard objects of various sizes, from small to large.

2.4. Model Validation

The validation process is carried out using a subset of the cardboard dataset to measure the performance of the model. The Confusion Matrix is used to describe the classification results of four possible outputs, namely true positive (the number of true positive data), true negative (the number of true negative data), false positive (the number of negatf data that is classified correctly) and false negative (the number of negative data that is classified incorrectly) [20].

From the Confusion Matrix, several Evaluation Metrics can be calculated. Evaluation metrics are tools or methods used to assess the performance of a model in completing a particular task. According to [21], some of the validation methods used to assess model performance include:

1. The precision, as formulated in equation (1), refers to the level of accuracy of the results obtained by the model [22].

$$presisi = \frac{true\ positive}{true\ positive + false\ positive} \quad (1)$$

2. The recall, as stated in equation (2), shows the extent to which the model successfully identifies relevant data as a whole [22].

$$recall = \frac{true\ positive}{true\ positive + false\ negative} \quad (2)$$

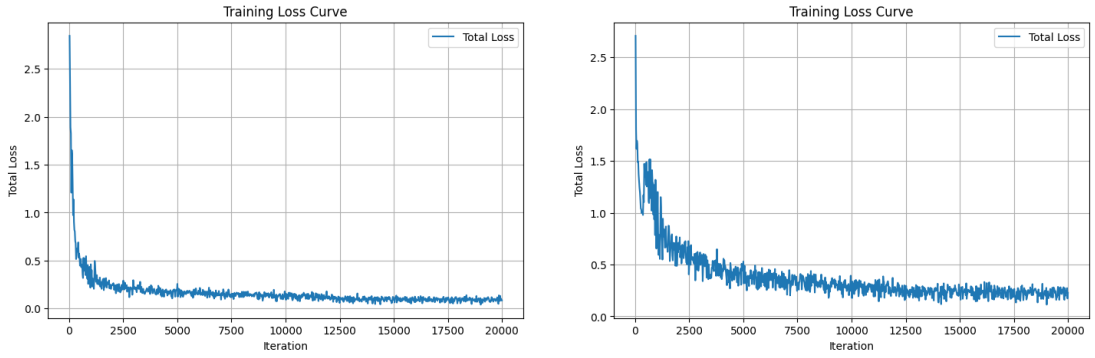
3. The Intersection over Union (IoU) metric serves as a means to gauge how well the predicted bounding box aligns with the actual object in object detection tasks. Essentially, it measures the degree of overlap between the predicted region and the ground truth, offering a practical indicator of the model's precision in localizing objects [23].

3. Results and Discussion

The results and discussion chapter delves into how effectively the Mask R-CNN model, when paired with the MobileNetV3 Small backbone, performs in identifying and segmenting cardboard objects. The evaluation employs a range of performance indicators such as training loss, confusion matrix, precision, recall, and Intersection over Union (IoU) to gauge the accuracy and quality of both the detection and segmentation outcome.

This section also compares the original version of Mask R-CNN, which uses ResNet-50 (Default), and Mask R-CNN, which has been modified with MobileNetV3 Small (Custom) as its backbone. Their performance is analyzed through the lens of training and validation data. Moreover, specific attention is given to evaluating both bounding box precision and mask segmentation capability, in order to determine how well each model recognizes and outlines the target objects.

3.1. Training Loss Graph

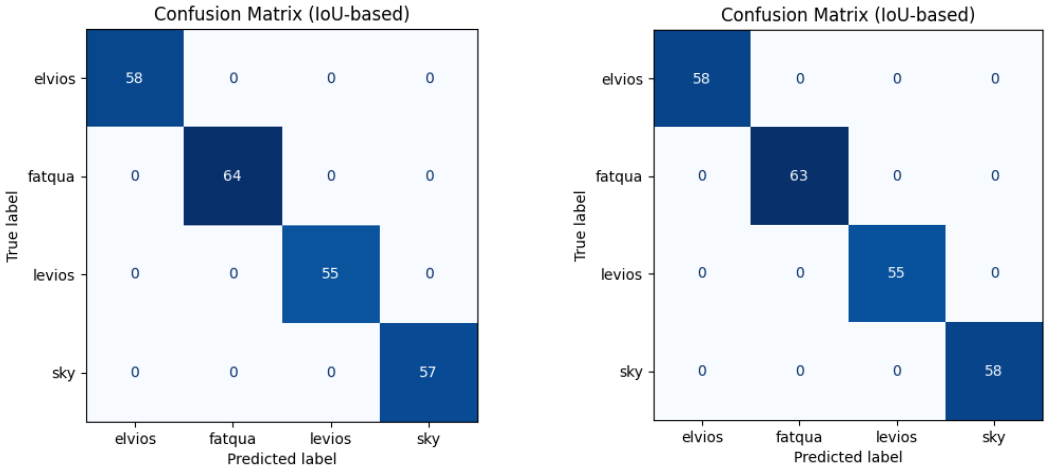


A B

Figure 5. Model Training Results

Figure 5 shows the training loss graph of the two models. Figure A is the graph of the Mask R-CNN model and B is the graph of the Mask R-CNN + MobileNetV3 Small model. Both graphs illustrate the change in total loss over 20,000 iterations. In Figure 5, the total loss initially has a high value of around 2.5 and decreases drastically in the first few thousand iterations. There are differences that can be seen in the loss fluctuation pattern. The graph of curve A shows a more stable decline with little fluctuation, while the graph of curve B has more variation, especially in the early stages of training, which indicates that the model in B has more difficulty in finding the optimal convergence path. Despite the different fluctuations, both models reach a relatively stable loss point after around 10,000 iterations, indicating that the model has learned successfully and has not experienced overfitting.

3.2. Confusion Matrix



A B

Figure 6. Confusion Matrix Results

The number of correct data that the model must achieve is elvios 59, sky 58, fatqua 64 and levios 55. Figure 6 shows the IoU-based confusion matrix on two different models, A is the Mask R-CNN model and B is the Mask R-CNN + MobileNetV3 Small model. Both models have almost the same results. Images A and B both classify elvios (58), levios (55), meaning that each model detects one wrong image of elvios as background and all correct on levios, while there are differences in the fatqua and sky classes, where model A records 64 correct predictions for fatqua and 57 for sky, while model B is slightly lower with 63 for fatqua and 58 for sky. This means that model A predicts all as correct for fatqua and has a slight misdetection of one image on sky and model B has one misdetection on fatqua and all as correct for sky.

3.3. Comparison of Bounding Box Evaluation

Table 2. Comparison of Bounding Box Evaluation

Category		Precision (Average)		Recall (Average)	
		Default	Custom	Default	Custom
Overall (@[IoU=0.50:0.95])	1 maxDets			0.814	0.745
	10 maxDets			0.968	0.880
	100 maxDets	0.955	0.846	0.968	0.880
High (IoU=0.50)		0.983	0.979		
Strict (IoU=0.75)		0.983	0.955		
Object Size	Small	0.900	0.700	0.900	0.700
	Medium	0.778	0.631	0.787	0.649
	Large	0.968	0.859	0.978	0.891

As shown in Table 2, the default Mask R-CNN model shows better performance than the custom model with the MobileNetV3 Small backbone, especially precision and recall on small to large objects. This difference in performance is even more evident in the IoU range of 0.50 to 0.95, where the default model is able to achieve higher average precision and recall, especially when the maximum number of detections (maxDets) is increased.

3.4. Comparison of Segmentation Evaluation

Table 3. Comparison of Segmentation Evaluation

Category		Precision (Average)		Recall (Average)	
		Default	Custom	Default	Custom
Overall (@[IoU=0.50:0.95])	1 maxDets			0.814	0.787
	10 maxDets			0.968	0.927
	100 maxDets	0.955	0.903	0.968	0.927
High (IoU=0.50)			0.974		
Strict (IoU=0.75)			0.963		
Object Size	Small	0.900	0.700	0.900	0.700
	Medium	0.778	0.658	0.787	0.722
	Large	0.968	0.920	0.978	0.937

Based on Table 3, the default Mask R-CNN model has higher average precision and recall on various object sizes, especially small objects (the difference is quite significant between 0.900 and 0.700) and solid results on strict IoU (0.75) with a precision score of 0.983 compared to 0.963. Custom models show competitive performance, especially for large objects.

3.5. Comparison of Evaluations by Category

Table 4. Comparison of Evaluations by Category

Category	Bounding box Mask R-CNN		Segmentation Mask R-CNN	
	Default	Custom	Default	Custom
Elvios	90.00	86.679	92.00	92.768
Fatqua	89.00	82.494	90.50	87.575

Category	Bounding box Mask R-CNN		Segmentation Mask R-CNN	
	Default	**Custom**	**Default**	**Custom**
Levios	91.50	88.260	93.00	93.572
Sky	87.20	81.155	88.50	87.572

Based on Table 4, the performance of the default Mask R-CNN bounding box model tends to be slightly higher in each category, while in segmentation it appears that the custom model with MobileNetv3 Small backbone is able to rival the performance of the default model, and in some cases even shows better values.

3.6. Comparison of Model Parameters and Sizes

Table 5. Comparison of Model Parameters and Size

Model	Parameters	Size
Mask R-CNN Default	43934184	334,9 Mb
Mask R-CNN + MobileNetV3 Small	20857704	159,3 Mb

Based on Table 5, the default Mask R-CNN has 43,934,184 parameters with a model size of 334.9 MB, while the Mask R-CNN version with MobileNetV3 Small backbone only requires 20,857,704 parameters and is 159.3 MB in size, making it lighter and more efficient in the use of computing resources. This difference indicates that the custom model can meet the needs of cardboard detection on devices that have limited resources without compromising the quality of the detection results.

3.7. Testing

A B

Figure 7. Live Test Results

Figure 7 shows the results of object detection testing using two different models, namely the default R-CNN Mask (Figure A) and the R-CNN Mask with a customised MobileNetV3 Small backbone (Figure B). In Figure A, object detection looks quite good with precise and minimal error segmentation in identifying relevant areas. Meanwhile, Figure B shows that the model successfully detects objects well but there are areas that should be the background but are detected as objects. This indicates that the model with the MobileNetV3 Small backbone has higher sensitivity but requires further optimisation to improve accuracy in distinguishing objects from the surrounding environment.

3.8. FPS Comparison

Table 6. FPS Comparison

Model	FPS									
	1	2	3	4	5	6	7	8	9	10
Default	8.17	8.08	7.84	7.98	7.96	7.90	7.88	7.83	7.83	7.76
Custom	11.00	11.23	11.33	10.30	10.35	10.75	11.41	10.93	10.80	11.11

Table 6 compares the Frame Per Second (FPS) performance between the two models. FPS testing was carried out 10 times in Figure 7. The default model has an average FPS of 7.92 and the Custom model 10.92. This shows that the custom model is more efficient in processing than the default model so that the MobilenetV3 Small backbone manages to increase detection speed without compromising quality.

3.9. Discussion of the differences between the Default and Custom Models

This study evaluates the ability of the Mask R-CNN model, which uses MobileNetV3 Small as its main backbone, to detect and segment cardboard boxes. Several metrics are used to assess its performance, including: training loss, confusion matrix, precision, recall, and Intersection over Union (IoU) score, which is the main indicator of the quality of detection and segmentation results.

The model developed in this study demonstrates a good balance between computational efficiency and detection accuracy compared to previous models. Specifically, the number of model parameters was reduced by 52.5% from 43.9 million in the standard Mask R-CNN model with ResNet-50 to just 20.86 million. At the same time, inference speed increased to 10.92 frames per second (FPS). This result even outperforms the MobileNetV2-based Mask R-CNN variant reported in [12], which has an inference time of approximately 0.525 seconds per image.

This improvement aligns with findings in [13], which noted that MobileNetV3 has better efficiency than its predecessors without significantly reducing accuracy.

However, the precision and recall rates of this model are still slightly below those of more complex models. For example, the model achieves an average precision (mAP) of 74.5% over the IoU range of 0.50–0.95, while the ResNet-101-based Mask R-CNN in [11] achieves up to 98.3% mAP. However, that model requires significantly more computational resources.

This comparison demonstrates that using a lightweight architecture like MobileNetV3 is highly suitable for real-time applications, especially when speed and efficiency are more important than a small increase in accuracy.

Additionally, the relatively small model size of 159.3 MB and high inference speed make it highly suitable for resource-constrained environments, such as warehouse management systems. This helps address real-world challenges like hardware costs and overlapping object issues.

5. Conclusions

This study introduces a Mask R-CNN model specifically developed using MobileNetV3 Small as the main architecture. This model is designed to detect and segment cardboard boxes. With this approach, computational efficiency has been significantly improved.

The number of model parameters has been reduced by 52.5% from 43.9 million in the standard ResNet-50-based model to just 20.86 million. Additionally, the model size has become smaller, at 159.3 MB, making it highly suitable for implementation on devices with limited resources. Inference speed has also improved, reaching 10.92 frames per second (FPS), outperforming the default model, which only achieves 7.92 FPS.

However, the accuracy rate in terms of mean average precision (mAP) within the IoU range of 0.50–0.95 stands at 74.5%, slightly below the ResNet-50-based model's 75.76%. Nevertheless,

this minor decrease is considered worthwhile given the improvements achieved in efficiency and speed, particularly for real-time applications.

This model demonstrates good performance in detecting medium to large objects. However, there are still challenges in detecting small objects, likely due to limitations in image resolution and feature extraction capabilities at small scales.

The main contributions of this research include: Integration of MobileNetV3 Small into the Mask R-CNN architecture, specifically for use in warehouse environments, Public availability of the model via the GitHub repository: https://github.com/mhsvikhatrivicika/custom-maskrcnn-mobilenetv3.git repository.

This study identified two main limitations: reduced accuracy for small objects and limited dataset variation, which affects generalization. To address this, future research should focus on improving the backbone to increase sensitivity to small objects, data augmentation techniques, and expanding the dataset with diverse warehouse scenarios to improve robustness. These improvements will strengthen the model's application in real-world environments while maintaining its efficiency advantages.

References

[1] Yohanes Mbiri, Kristina Sara, and Anastasia Mude, "Rancang Bangun Sistem Informasi Adiministrasi Kependudukan Berbasis Website Menggunakan Metode Agile Di Desa Nginamanu Barat Kecamatan Wolomeze Kabupaten Ngada," *Simtek J. Sist. Inf. dan Tek. Komput.*, vol. 8, no. 1, pp. 148–153, Apr. 2023, doi: 10.51876/simtek.v8i1.236.

[2] S. Teja, M. I. Jalil, S. Nurakmalia, F. A. Rizaldi, and A. Saifudin, "Analisis dan Perancangan Sistem Informasi Inventory pada PT Insan Data Permata," *JURIHUM J. Inov. dan Hum.*, vol. 1, pp. 231–239, Apr. 2023, doi: 10.30998/jrami.v1i02.231.

[3] E. Fontana, W. Zarotti, and D. Lodi Rizzini, "A Comparative Assessment of Parcel Box Detection Algorithms for Industrial Applications," in *2021 European Conference on Mobile Robots (ECMR)*, IEEE, Aug. 2021, pp. 1–6. doi: 10.1109/ECMR50962.2021.9568825.

[4] M. A. Masril and D. P. Caniago, "Optimasi Teknologi Computer Vision pada Robot Industri Sebagai Pemindah Objek Berdasarkan Warna," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 11, no. 1, p. 46, Jan. 2023, doi: 10.26760/elkomika.v11i1.46.

[5] T. Anjali Dompeipen, S. R. U. . Sompie, and M. E. . Najoan, "Computer Vision Implementation for Detection and Counting the Number of Humans," *J. Tek. Inform. vol. 16 no. 1*, vol. 16, no. 1, pp. 65–76, 2021, doi: 10.35793.

[6] A. Y. Firmandicky and Y. A. Susetyo, "Klasifikasi Kardus Barang di PT XYZ Menggunakan Convolutional Neural Network dengan Pendekatan Fine Grained Image Classification," *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 8, no. 4, pp. 954–964, Oct. 2024, doi: 10.35870/jtik.v8i4.2337.

[7] J. Yang *et al.*, "SCD: A Stacked Carton Dataset for Detection and Segmentation," *Sensors*, vol. 22, no. 10, p. 3617, May 2022, doi: 10.3390/s22103617.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, 2020, doi: 10.1109/TPAMI.2018.2844175.

[9] A. Naumann, F. Hertlein, L. Dörr, and K. Furmans, "TAMPAR: Visual Tampering Detection for Parcel Logistics in Postal Supply Chains," in *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, IEEE, Jan. 2024, pp. 8061–8071. doi: 10.1109/WACV57701.2024.00789.

[10] S. Fang, B. Zhang, and J. Hu, "Improved Mask R-CNN Multi-Target Detection and Segmentation for Autonomous Driving in Complex Scenes," *Sensors*, vol. 23, no. 8, 2023, doi: 10.3390/s23083853.

[11] R. Rubin, C. Jacob, S. M. Anzar, and A. Panthakkan, "Mask R-CNN with Multi-

Backbones - A Comparative Analysis," *2022 5th Int. Conf. Signal Process. Inf. Secur. ICSPIS 2022*, no. December, pp. 55–60, 2022, doi: 10.1109/ICSPIS57063.2022.10002546.

[12] C. Huang, Y. Zhou, and X. Xie, "Intelligent Diagnosis of Concrete Defects Based on Improved Mask R-CNN," *Appl. Sci.*, vol. 14, no. 10, 2024, doi: 10.3390/app14104148.

[13] L. Cao, P. Song, Y. Wang, Y. Yang, and B. Peng, "An Improved Lightweight Real-Time Detection Algorithm Based on the Edge Computing Platform for UAV Images," *Electron.*, vol. 12, no. 10, 2023, doi: 10.3390/electronics12102274.

[14] M. Abd Elaziz, A. Dahou, N. A. Alsaleh, A. H. Elsheikh, A. I. Saba, and M. Ahmadein, "Boosting covid-19 image classification using mobilenetv3 and aquila optimizer algorithm," *Entropy*, vol. 23, no. 11, pp. 1–17, 2021, doi: 10.3390/e23111383.

[15] T. Shahriar, "Comparative Analysis of Lightweight Deep Learning Models for Memory-Constrained Devices," pp. 1–22, 2025, [Online]. Available: http://arxiv.org/abs/2505.03303

[16] A. Howard *et al.*, "Searching for mobileNetV3," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2019-Octob, pp. 1314–1324, 2019, doi: 10.1109/ICCV.2019.00140.

[17] D. D. Karyanto, D. D. Karyanto, J. Indra, A. R. Pratama, and T. Rohana, "Detection of the Size of Plastic Mineral Water Bottle Waste Using the Yolov5 Method," *JIKO (Jurnal Inform. dan Komputer)*, vol. 7, no. 2, pp. 123–130, 2024, doi: 10.33387/jiko.v7i2.8535.

[18] R. Budi, R. A. Harianto, and E. Setyati, "Segmentasi Citra Area Tumpukan Sampah Dengan Memanfaatkan Mask R-CNN," *J. Intell. Syst. Comput.*, vol. 5, no. 1, pp. 58–64, 2023, doi: 10.52985/insyst.v5i1.305.

[19] N. Sarasuartha Mahajaya, P. Desiana, W. Ayu, and R. R. Huizen, "Pengaruh Optimizer Adam, AdamW, SGD, dan LAMB terhadap Model Vision Transformer pada Klasifikasi Penyakit Paru-paru," *Pros. Semin. Has. Penelit. Inform. dan Komput.*, vol. 1, no. 2, pp. 818–823, 2024.

[20] N. Uly, H. Hendry, and A. Iriani, "CNN-RNN Hybrid Model for Diagnosis of COVID-19 on X-Ray Imagery," *Digit. Zo. J. Teknol. Inf. dan Komun.*, vol. 14, no. 1, pp. 57–67, 2023, doi: 10.31849/digitalzone.v14i1.13668.

[21] A. Ardiansyah, J. Triloka, and Indera, "Evaluasi Kinerja Model YOLOv8 dalam Deteksi Kesegaran Buah," *JUPITER*, vol. 16, no. 2, pp. 357–368, 2024, doi: 10.5281/zenodo.11296226.

[22] A. P. Nardilasari, A. L. Hananto, S. S. Hilabi, T. Tukino, and B. Priyatna, "Analisis Sentimen Calon Presiden 2024 Menggunakan Algoritma SVM Pada Media Sosial Twitter," *JOINTECS (Journal Inf. Technol. Comput. Sci.*, vol. 8, no. 1, p. 11, 2023, doi: 10.31328/jointecs.v8i1.4265.

[23] M. alfin Mansyur and N. Pratiwi, "Deteksi manusia dengan algoritma yolo untuk pemutaran audio otomatis di area tertentu," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 10, no. 1, pp. 667–674, 2025, doi: doi.org/10.29100/jipi.v10i1.5967.