Application of Object Detection and Face Recognition with Customize Dataset on Service Robot

Ryan Satria Wijaya[1], Wiki Saputra[2], Senanjung Prayoga[3], and Eko Rudiawan Jamzuri[4]

[1.2.3.4] Barelang Robotics Artificial and Intellegence Lab (BRAIL), Department of Electrical Engineering, Politeknik Negeri Batam, Kepulauan Riau 29461, Indonesia
ryan@polibatam.ac.id

Abstract. Computer vision technology is currently gaining traction in all industries, including manufacturing, agriculture, healthcare, and services. Computer vision technology now incorporates robotics technology, making it very dynamic and flexible. Computer vision, like a service robot, is used in a variety of applications, including safety, analysis, and service. One of the skills of service robots is the ability to recognize its users through the use of computer vision. So that by recognizing the user, the robot can be commanded according to the user's wishes. Computer vision on service robots is trained using a special dataset that includes the object of a user's face. The computer will be trained by recognizing its users through digital images and annotating their faces, followed by training using the yolov5 architecture and applying the resulting data to the robot.

Keywords: Service Robot, Computer Vision, Yolov5

1 Introduction

Technology is currently undergoing rapid development, which has a significant impact on daily life. Computer vision is a massive technological advancement that enables computers to analyze and interpret visual data like images and videos with improved efficiency. It has greatly influenced many aspects of human life[1]. Object detection[2] involves identifying objects of interest in pictures and videos, while face recognition defines individuals based on facial features. These tasks have real-world applications the same as surveillance, security, and biometrics[3] Object detection includes datasets[4], algorithms, and techniques. There are two types of learning: supervised and unsupervised[5]. In supervised learning, the model labels data before detecting it during testing. In unsupervised models, no labeled data is provided and the model learns by calculating loss[6]Advanced structures and unequal background information distribution can pose new challenges in complex landscapes and urban environments[7]. On this project, the researcher is using the supervised learning method. As we know, in terms of creating to create a customized model, we should first create a dataset from scratch with labeled images, and the dataset represents as the input to the model. Researcher developed a service robot, which is a computer vision-based robot that can

L. Lumombo et al. (eds.), *Proceedings of the 7th International Conference on Applied Engineering (ICAE 2024)*, Advances in Engineering Research 251,
https://doi.org/10.2991/978-94-6463-620-8_5

recognize objects based on the name of the robot user, allowing a robot to be programmed to be commanded by its own users in future research.

2 Method

2.1 Service Robot

Robots created with the intention of enhancing human life are known as service robots. These robots can be found in a variety of environments, including manufacturing, restaurants, and airports. A prototype robot was created by researchers it has computer vision, manipulators, and autonomous movement features.

Table 1. Main components service robot

Components	Name	Quantity
Wheel	Mecanum Wheel	4
Actuator	Motor Encoder	4
Lidar	RPLIDAR A3	1
Camera	Webcam	1
Microcontroller	ESP32	2
Mini PC	Jetson Nano 2GB	1
Battery	Li-Po Battery	1

The hardware components listed in Table 1 are all necessary to support the robot's performance. Despite being built with low-quality hardware, this robot has a high performance level that is supported by lightweight software. This study applies a computer vision system to a service robot. The computer vision system's function on the robot is useful in detecting and recognizing the following objects. In general, the purpose of face recognition is when the object detected by the robot is the user's face. The user's face serves as a sample object for the machine to recognize from. Face recognition can be used for a variety of purposes, including security systems in which the robot can only operate if it recognizes the user.

2.2 Dataset

Table 2. Grouping classes

Class	Images	Total Images
arya	2.022	8.099
dikki	2.032	
muammar	2.026	
wiki	2.019	

Table 2 displays the quantity of images required for each object, as the robot must identify a very specific type of object. The type of digital image[8] loaded is a close-up of a service robot project member's face. It was chosen so that it could be uniquely identified as a human face since all human faces have eyes, mouths, mustaches, and beards. In light of this, the computer must be able to identify a face with very little variation.

Fig. 1. Sample of Classes

Figure 1 shows a sample of each object used in the custom dataset, each image shows a minor difference, the objects of dikki and arya each have a beard and mustache, whereas wiki and muammar have a non-dominant mustache. This has very little difference, finding it challenging for researchers to create a dataset using the following objects.

2.3 Train Settings

Table 3. Train settings

Parameter	Args
img	416
batch	16
epochs	100
weights	Yolov5s.pt

Table 3 show lists the train parameter configuration. Training settings represented hyperparameters and configurations used during the training process. Image is the size of the input image, batch is the batch size, epochs is the number of training iterations and weights is pretrained checkpoint. The purpose of training is to create a model for object detection implementation. A hyperparameter configuration needs to be set before training a customize model to influence the model's speed and accuracy.

2.4 YOLOv5 Architecture

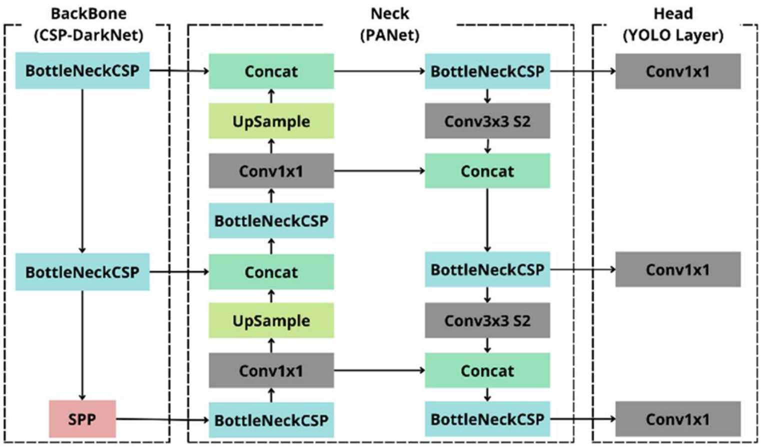

Fig. 2. YOLOv5 Architecture

The standard construction of model configuration in YOLOv5 has various implications. The model configuration has now been divided into three main parts: backbone, neck, and head [9]. YOLO has several versions, the most recent being YOLOv8, which has an accuracy level comparable to YOLOv5. However, YOLOv5's performance in real-time system detection is more stable and faster than previous versions of YOLO[10]. Based on figure 2, The YOLOv5[11], architecture is composed of three main main components: the backbone, the neck, and the head. The backbone is the network's main body and was designed using the new CSP-Darknet structure. The neck connects the backbone and the head using SPP and the new PANet structure[12], and the head generates the final output as YOLO layer.

Researchers use YOLOv5 because they have experimented with using the YOLOv5 and YOLOv8 architectures in this project. However, the YOLOv8 architecture has limitations in the model conversion process that do not support when parsing resize nodes, and YOLOv8 does not support the INT64 weight model, requiring the weight model's inference precision to be reduced to INT32. As a result, during the conversion process, the expected input type differs from the actual input type.

2.5 Annotating Raw Images

A platform called Roboflow is a open source web based tool intended to make deep learning computer vision tasks easier. With its extensive feature set, it helps developers create computer vision applications. These consist of the following: annotating data sets[13], pre-processing data sets[14], combining projects or data sets, confirming the conditions of datasets, exporting datasets, and model training[15]. Roboflow's features make it easier for developers working on computer vision projects to streamline their workflow[16].

2.6 Training Dataset

The Google Colab platform, a cloud-based platform for writing and executing Python code through a web browser, is used to train customized models. Google Colab's Python-based deep learning teaching approach is ideal for classroom use due to its user-friendly interface and free GPU access [17]. The Colab notebooks were designed for students that have never programmed before. There are exercises in every Colab notebook as well as solutions to the suggested exercises everywhere[18].

2.7 Training Flow Process

Neural networks are the tool for designing artificial intelligence technology, and machine learning is the science that studies the design of intelligent machines[19]. Deep learning is a type of machine learning technique that uses multiple layers of information processing stages in hierarchical architectures for pattern classification and unsupervised feature learning. It is at the intersection of the fields of signal processing, graphical modeling, neural networks[20], optimization, and pattern recognition. Deep learning is considered a type of supervised learning algorithm. Besides that, deep learning is a unique category within the larger field of machine learning techniques due to its unique architecture and methodology[21]. The massive increases in chip processing power (such as GPU units) and the markedly reduced cost of computing hardware are two key factors in the current popularity of deep learning. [22].

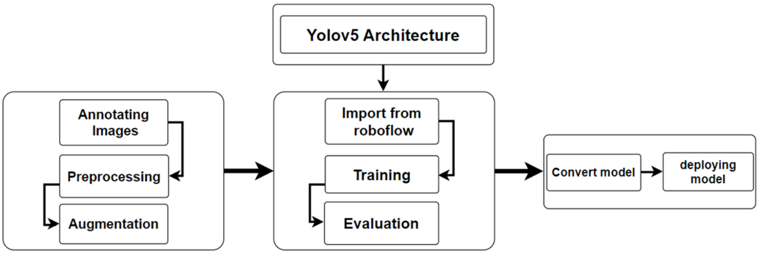

Fig. 3. Train custom models flow process

Figure 3 shows the process to become an object detection model, this customize model must go through a number of steps. These include obtaining digital images, annotation of images based on object classes, preparation of preprocessing and augmentation to convert images annotation format for easy reading during training, and finally training the model with multiple configurations to control it's own quality. Because the Yolov5 custom model file was created on low-spec hardware, it needed to be converted to a TensorRT (.engine) file.The output file of the custom training model is a (.pt) file, which is a checkpoint file.

3 Result and Discussion

3.1 Performance Training

This is a crucial step in determining whether and how much bias has been towards either class. The confusion matrix contains four entries, which is the relevant number for determining the metric for a binary classifier. According to equation (1), there are various ways to interpret certain metrics when assessing multi-class classification techniques[23]. While these conversations are outside the purview of this one, the fundamental ideas are still relevant in situations involving multiple classes.

$$M = \begin{pmatrix} TP & FN \\ FP & TN \end{pmatrix} \quad (1)$$

The confusion matrix includes four classifications: True Negative (TN), True Positive (TP), False Negative (FN), and False Positive (FP) based on actual and predicted values. The TP (True Positive) value represents the number of correctly classified positive samples. TN (True Negative) refers to the number of negative samples correctly classified. False Positive (FP) refers to the number of negative samples incorrectly classified as positive. FN (False Negative) refers to the number of positive samples incorrectly classified as negative[24].

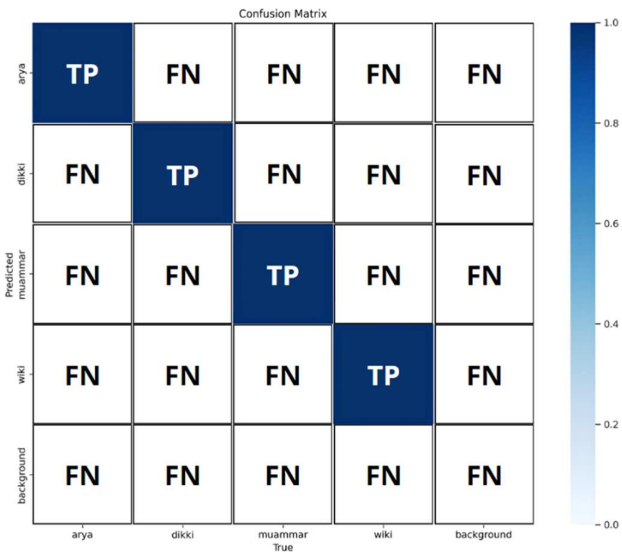

Fig. 4. Confusion matrix multi class

The confusion matrix, which shows the number of true positives, true negatives, false positives, and false negatives for each class, gives a thorough summary of the outcomes in Figure 4. Since there are no other values in the other columns and each class is displayed with a value of 1, it is clear that each class is performing well.

Table 4. Confusion matrix calculation on test data

Class	Images	Total Images	TP	TN	FP	FN	P	R
arya	214	804	214	590	0	0	1.00	1.00
dikki	209		209	594	0	1	1.00	0.995
muammar	188		188	613	2	1	0.989	0.994
wiki	193		193	611	0	0	1.00	1.00

Table 4 is a test on test data that contains 804 images divided into four classes, the results show that the model can detect objects and identify classes with more than 85% confidence level. The values in Table 3 are the result of the precision and recall values that were obtained on the classification result samples with IoU threshold values greater than 50%. Implying that recall and precision are nearly equal to 1.0, indicating that the model can predict accurately.

Precision. In figure 5, The curve is consistently at 0.995 all classes, that also, when compared to the detailed score mAP@0.5 of each class, reaches 0.995 from 1, the blue graph visualize that mean indicating all classes consistently at 0.995 from first iteration untill process finished. This graph indicating that the model detects well when using an IoU threshold of 0.50. Instead of plotting different confidence thresholds, the Precision-Confidence Curve plots precision. This graph illustrates how accuracy varies depending on the confidence level used to classify a prediction as positive.

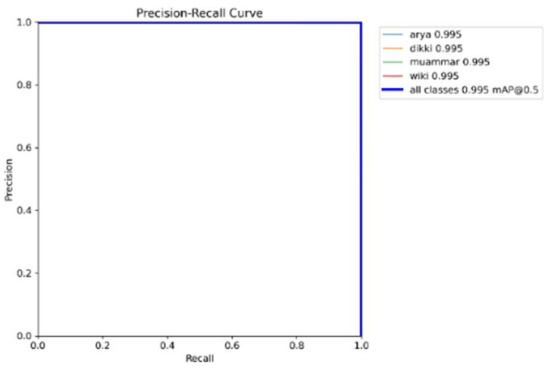

Fig. 5. Precision recall curve

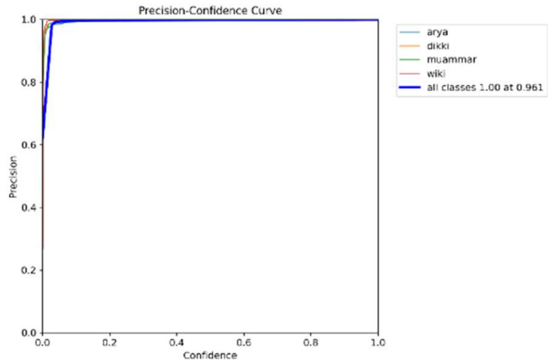

Fig. 6. Precision confidence curve

Figure 6 indicates the curves show that the first iteration begins at 0.6 and increases until the last iteration reaches 0.961 out of 1 in all classes; this data finding shows that the data is above than 80%, indicating that the positive prediction (TF) is correct. Based on the formula and calculation in equation (2), each class has a score of 1 or 100%.

$$\text{Precision} = \frac{\text{TP}}{(\text{TP}+\text{FP})} \tag{2}$$

Recall Confidence. According to Figure 7, the curve is consistently at 1.0 in the initial iteration, with a decrease in the recall value of 0.9 and a return to 1.0 in the final iteration. The data above demonstrates that the model can maintain the confidence value when the confidence value decreases.

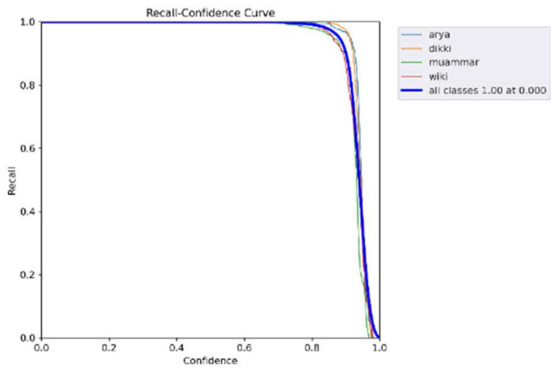

Fig. 7. Recall confidence curve

$$\text{Recall} = \frac{\text{TP}}{(\text{TP+FN})}$$
(3)

Recall is the measure of the proportion of true positive predictions among all real positive occurrences, it is also referred to as sensitivity or true positive rate.. It is calculated as the ratio of TP to the total of TP and false negatives (FN). It can be written using equation (3).

F1 Score. The F1 Score is a combination calculation of the precision and recall values, which are then referred to as measurement values. The result is referred to as the measurement value[25]. According to Figure 8, in the first iteration, the F1 score is 0.663, and in the successive iterations, the value peaks at 1 until the final iteration; this condition is positive because the best score is at 1, and this data demonstrates that the custom model has good precision and recall. To find out the calculation of the F1 value, it can be calculated with equation (4) below.

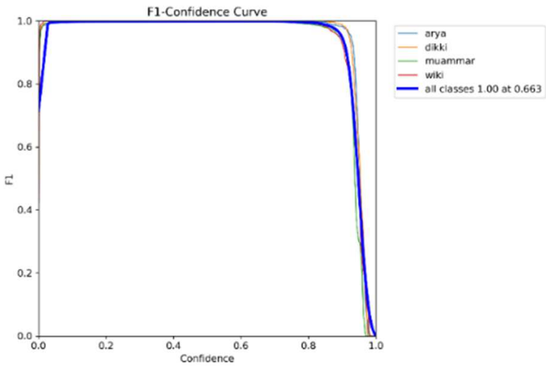

Fig. 8. F1 Confidence curve

$$\text{F1 Score} = 2\ \frac{(\text{Precision}*\text{Recall})}{(\text{Precision}+\text{Recall})}$$
(4)

Mean Average Precision. The Precision-Recall AUC (Area Under Curve) is computed using Average Precision (AP). A total of 804 images were used in the test data for the Average Precision testing. Every object has an AP value; the lowest is 99%, and the highest is 100%. The following equation (5) can be used to get the Average Precision value.

Table 5. Training result

Epochs	Precision	Recall	mAP@0.5	mAP@0.5:0.95

90	0.9991	0.99954	0.99449	0.89365
91	0.99909	0.99954	0.99449	0.89401
92	0.99911	0.99953	0.9945	0.89384
93	0.99911	0.99952	0.99449	0.89373
94	0.99913	0.99952	0.99449	0.89374
95	0.99913	0.99951	0.99449	0.89443
96	0.99913	0.99951	0.99449	0.89514
97	0.99914	0.9995	0.99449	0.89533
98	0.99914	0.9995	0.9945	0.89569
99	0.99914	0.9995	0.9945	0.89538

$$\text{Average Precision} = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1\}} \rho\text{interp}(r) \tag{5}$$

$$\rho\text{interp}(r) = \max_{\tilde{r}:\tilde{r} \geq r} \rho \tag{6}$$

$$\text{mAP@}\alpha = \frac{1}{n} \sum_{k=1}^{k=n} \text{AP}_k \tag{7}$$

APk = the AP of class k

n = the number of classes

The average precision value of each class is calculated using equation (5), where the above method is an 11 point interpolated precision method. Based on the necessary IoU threshold, the Mean Average Precision (mAP) value can be found by calculating the average precision in each class. To calculate the mean average precision, use equation (6). At the 100th epoch at table 5, the model can achieve an mAP value of 0.9945 with an IoU threshold of 0.5, and 0.89538 with an IoU threshold of 0.5:0.95.

Intersection over Union (IoU). This formula calculates the correlations between ground truth bounding boxes and prediction bounding boxes to determine the accuracy of prediction bounding boxes. The IoU metric produces a value between 0 and 1, but in this test, IoU testing will be performed visually in the image using test data on the model. to calculate IoU using equation (5) below.

$$\text{IoU} = \frac{area\ (B_{Prediction} \cap B_{Ground\ Truth})}{area\ (B_{Prediction} \cup B_{Ground\ Truth})} \tag{8}$$

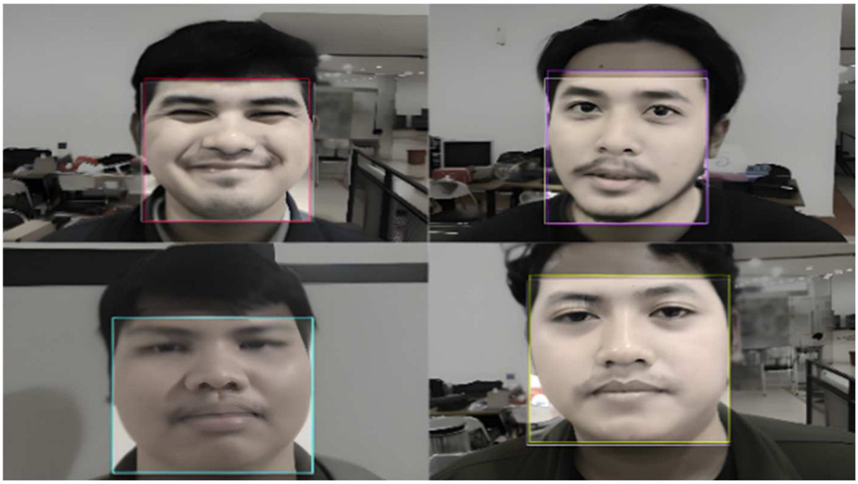

Fig. 9. Comparison between ground truth and prediction bounding box

Figure 9 compares ground truth and prediction bounding boxes, the ground truth bounding box is represented by a white bounding box image, and the prediction bounding box is a colored bounding box, The comparison between ground truth and prediction bounding boxes. is not significant. This means that the IoU value in this model is expected to be higher than 90%. The image results show that the model created with 8,099 datasets and 100 epochs in the training process can perform good detection, as evidenced by the image above, where the model can predict each object accurately.

3.2 Performance Testing

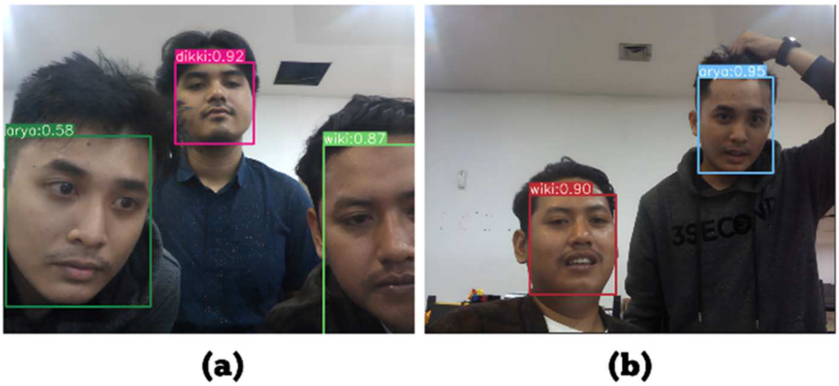

Fig. 10. Real-time object detection on service robot

Real-time object detection in Figure 10 a,b shows that the bounding box is exactly in the previously annotated area; changes in the object's position have no effect on the bounding box; and the bounding box remains within the predetermined face area. The confidence value of the object is greater than 80%, depending on the distance between the camera and the detected object. The files that have been converted into TensorRT format are loaded onto the robot, and real-time detection experiments are carried out via webcam. This process tests several aspects, including the bounding box, detection accuracy, and speed.

Table 6. Evaluation on validation data

Class	Images	P	R	mAP@0.5	mAP@0.5:0.95
arya	1599	0.991	1	0.994	0.894
dikki		0.999	0.998	0.995	0.889
muammar		0.996	1	0.995	0.858
wiki		0.991	0.995	0.995	0.838

Table 6 represents the validation results for the model that uses a total of 1.599 images to evaluate each class. Each class receives an average score of 0.9 for precision, recall, and mAP with a threshold of 0.50, while the average score for the mAP test results with a treshold of 0.5 to 0.95 is 0.8.

4 Conclusion

Researchers conducted a variety of experiments for this study. Although human faces have similarities, they are difficult to distinguish. As a result, it would be extremely difficult to identify human faces from names. This study successfully developed a specialized face recognition model. With the YOLOv5 architecture, the model can produce mAP@0.5 values of 0.99 and mAP@0.5:0.95 values of 0.89 in training performance, and in model evaluation on each class, the model produces average values on precision, recall, and mAP@0.5 values of 0.99 and 0.88 on each object. This model can also adapt the model created with 8,093 digital images to recognize a person's face based on their name. In real-time object detection, the robot can detect objects with a confidence level above 85% on each object detected at a detection speed of 9-11 Fps. This confirms that the customize dataset, which was created with 8,099 images overall—roughly 2000 images for each class—can be used to train the model over 100 epochs and be used appropriately on the robot. Testing real-time detection objects on the robot that the robot can recognize each user with a confidence level above 65%.

References

1. S. Lika, Y. Pernando, and A. Kurniawan, "Lightweight Deep Learning for Object De-tection on Mobile Device," Bulletin of Informatics and Data Science, vol. 2, no. 2, p. 106, Nov. 2023, doi: 10.61944/bids.v2i2.82.
2. F. M. Sinaga, Gunawan, S. Winardi, H. Kurniawan, W. S. Lestari, and K. M. Tarigan, "Object Detection in E-Commerce Using YOLO in Real Time," Teknika, vol. 13, no. 1, pp. 145–154, Mar. 2024, doi: 10.34148/teknika.v13i1.773.
3. K. S. S. Reddy, G. Ramesh, J. Praveen, P. Surekha, and A. Sharma, "A Real-time Automated System for Object Detection and Facial Recognition," in E3S Web of Conferences, EDP Sciences, Oct. 2023. doi: 10.1051/e3sconf/202343001076.
4. P. Chotikunnan, T. Puttasakul, R. Chotikunnan, B. Panomruttanarug, M. Sangworasil, and A. Srisiriwat, "Evaluation of Single and Dual Image Object Detection through Image Segmentation Using ResNet18 in Robotic Vision Applications," Journal of Robotics and Control (JRC), vol. 4, no. 3, pp. 263–277, May 2023, doi: 10.18196/jrc.v4i3.17932.
5. H. Jurnal, A. Fathurohman FKIP, and P. Fisika, "JURNAL INFORMATIKA DAN TEKNOLOGI KOMPUTER MACHINE LEARNING UNTUK PENDIDIKAN: MENGAPA DAN BAGAIMANA," vol. 1, no. 3, pp. 57–62, 2021.
6. U. Arshad, "Object Detection in Last Decade - A Survey," Scientific Journal of In-formatics, vol. 8, no. 1, pp. 60–70, May 2021, doi: 10.15294/sji.v8i1.28956.
7. J. Zhang, Z. Chen, G. Yan, Y. Wang, and B. Hu, "Faster and Lightweight: An Im-proved YOLOv5 Object Detector for Remote Sensing Images," Remote Sens (Basel), vol. 15, no. 20, Oct. 2023, doi: 10.3390/rs15204974.
8. A. R. Sindar Sinaga, "Optical Systems and Digital Image Acquisition," 2018.
9. I. Purwita Sary, E. Ucok Armin, S. Andromeda, E. Engineering, and U. Singaperbang-sa Karawang, "Performance Comparison of YOLOv5 and YOLOv8 Architectures in Human Detection Using Aerial Images," Ultima Computing : Jurnal Sistem Komput-er, vol. 15, no. 1, 2023.
10. P. A. Cahyani, M. Mardiana, P. B. Wintoro, and M. A. Muhammad, "Sistem Perhi-tungan Kendaraan Menggunakan Algortima YOLOv5 dan DeepSORT," Jurnal Teknik Informatika dan Sistem Informasi, vol. 10, no. 1, May 2024, doi: 10.28932/jutisi.v10i1.7519.
11. V. R. Oktavia, A. W. S. B. Johan, W. G. Ananta, F. Refiandi, and M. K. Karim, "De-tection of Motorcycle Headlights Using YOLOv5 and HSV," Teknika, vol. 12, no. 3, pp. 189–197, Oct. 2023, doi: 10.34148/teknika.v12i3.682.
12. N. I. Nife and M. Chtourou, "Improved Detection and Tracking of Objects Based on a Modified Deep Learning Model (YOLOv5)," International Journal of Interactive Mobile Technologies, vol. 17, no. 21, pp. 145–160, 2023, doi: 10.3991/ijim.v17i21.45201.
13. "Object Detection and Image Annotation Using Deep Learning," International Re-search Journal of Modernization in Engineering Technology and Science, Jun. 2023, doi: 10.56726/irjmets41723.
14. T. S. Wang, G. T. Kim, M. Kim, and J. Jang, "Contrast Enhancement-Based Prepro-cessing Process to Improve Deep Learning Object Task Performance and Results," Applied Sciences (Switzerland), vol. 13, no. 19, Oct. 2023, doi: 10.3390/app131910760.
15. K. Gautam and D. N. Agarwal, "Anomaly Detection in Video Surveillance using Ob-ject Detection," 2022.
16. A. Y. Alin, K. Kusrini, and K. A. Yuana, "The Effect of Data Augmentation in Deep Learning with Drone Object Detection," IJCCS (Indonesian Journal of Computing and Cybernetics Systems), vol. 17, no. 3, p. 237, Jul. 2023, doi: 10.22146/ijccs.84785.

17. D. H. Kim, “A Study on an Effective Teaching of AI using Google Colab-Based DCGAN Deep Learning Model Building for Music Data Analysis and Genre Classifi-cation,” International Journal of Recent Technology and Engineering (IJRTE), vol. 11, no. 6, pp. 13–25, Mar. 2023, doi: 10.35940/ijrte.E7351.0311623.
18. W. Vallejo, C. Díaz-Uribe, and C. Fajardo, “Google Colab and Virtual Simulations: Practical e-Learning Tools to Support the Teaching of Thermodynamics and to In-troduce Coding to Students,” ACS Omega, vol. 7, no. 8, pp. 7421–7429, Mar. 2022, doi: 10.1021/acsomega.2c00362.
19. W. Rahmaniar and A. Hernawan, “Real-time human detection using deep learning on embedded platforms: A review,” Journal of Robotics and Control (JRC), vol. 2, no. 6. Department of Agribusiness, Universitas Muhammadiyah Yogyakarta, pp. 462-468Y, Nov. 01, 2021. doi: 10.18196/jrc.26123.
20. U. Azmi, S. Soehardjoepri, R. Prihandoko, and I. Asif, “Application of Artificial Neural Network in Predicting Direct Economic Losses Due to Earthquake,” Jurnal Varian, vol. 7, no. 1, pp. 37–46, Oct. 2023, doi: 10.30812/varian.v7i1.2326.
21. M. S. Binetti, C. Massarelli, and V. F. Uricchio, “Machine Learning in Geosciences: A Review of Complex Environmental Monitoring Applications,” Mach Learn Knowl Extr, vol. 6, no. 2, pp. 1263–1280, Jun. 2024, doi: 10.3390/make6020059.
22. C. Mishra and D. L. Gupta, “Deep Machine Learning and Neural Networks: An Over-view,” IAES International Journal of Artificial Intelligence (IJ-AI), vol. 6, no. 2, p. 66, Jun. 2017, doi: 10.11591/ijai.v6.i2.pp66-73.
23. S. A. Hicks et al., “On evaluation metrics for medical applications of artificial intelli-gence,” Sci Rep, vol. 12, no. 1, Dec. 2022, doi: 10.1038/s41598-022-09954-8.
24. Y. Zhang, T. Zuo, L. Fang, J. Li, and Z. Xing, “An Improved MAHAKIL Over-sampling Method for Imbalanced Dataset Classification,” IEEE Access, vol. 9, pp. 16030–16040, 2021, doi: 10.1109/ACCESS.2020.3047741.
25. S. Clara et al., Implementasi Seleksi Fitur Pada Algoritma Klasifikasi Machine Learn-ing Untuk Prediksi Penghasilan Pada Adult Income Dataset. 2021.