

Design of a Real-Time Object Detection Prototype System With YOLOv3 (You Only Look Once)

Chichi Rizka Gunawan^{1*}, Nurdin², Fajriana²

¹Master Student Department of Information Technology, Universitas Malikussaleh, Aceh, Indonesia

²Department of Information Technology, Universitas Malikussaleh, Aceh, Indonesia

*Corresponding author E-mail: chichirizkagunawan@gmail.com

Manuscript received 15 April 2022; revised 1 May 2022; accepted 15 June 2022. Date of publication 25 July 2022

Abstract

Object detection is an activity that aims to gain an understanding of the classification, concept estimation, and location of objects in an image. As one of the fundamental computer vision problems, object detection can provide valuable information for the semantic understanding of images and videos and is associated with many applications, including image classification. Object detection has recently become one of the most exciting fields in computer vision. Detection of objects on this system using YOLOv3. The You Only Look Once (YOLO) method is one of the fastest and most accurate methods for object detection and is even capable of exceeding two times the capabilities of other algorithms. You Only Look Once, an object detection method, is very fast because a single neural network predicts bounded box and class probabilities directly from the whole image in an evaluation. In this study, the object under study is an object that is around the researcher (a random thing). System design using Unified Modeling Language (UML) diagrams, including use case diagrams, activity diagrams, and class diagrams. This system will be built using the python language. Python is a high-level programming language that can execute some multi-use instructions directly (interpretively) with the Object Oriented Programming method and also uses dynamic semantics to provide a level of syntax readability. As a high-level programming language, python can be learned easily because it has been equipped with automatic memory management, where the user must run through the Anaconda prompt and then continue using Jupyter Notebook. The purpose of this study was to determine the accuracy and performance of detecting random objects on YOLOv3. The result of object detection will display the name and bounding box with the percentage of accuracy. In this study, the system is also able to recognize objects when they object is stationary or moving.

Keywords: YOLO, YOLOv3, Python, Anaconda.

1. Introduction

With the development of the times, humans continue to develop knowledge and technology to help and ease their work. One area of research that is still developing is artificial intelligence or better known as Artificial Intelligence (AI) [1][2][3] [4].

Machine Learning is an approach in AI that is widely used to replace or imitate human behavior to solve problems or perform automation. As the name implies, machine learning tries to imitate how humans or intelligent creatures learn and generalize. The hallmark of machine learning is the existence of a training, learning, or training process. Therefore, machine learning requires data to learn, known as training data [5] [6] [7].

Object detection is the ability of a system to recognize objects that are in an image or video [8]. Then the object detection process begins with the file.bmp extension from the original image, then resizing, grayscale, and edge detection convolution [9]. As one of the fundamental computer vision problems, object detection can provide valuable information for the semantic understanding of images and videos and is associated with many applications, including image classification. Object detection has recently become one of the most exciting fields in computer vision [2] [10].

The You Only Look Once (YOLO) method is one of the fastest and most accurate methods for object detection and is even capable of exceeding 2 times the capabilities of other algorithms. You Only Look Once, an object detection method, is very fast because a single neural network predicts bounded box and class probabilities directly from the full image in an evaluation [11] [12]. However, it makes more localization errors and the training speed is relatively slow. This research will create a system to detect objects in real-time. This study aims to determine the accuracy and performance of this algorithm by utilizing surrounding object data. It is hoped that this research will be able to provide accuracy values and show better performance of object detection algorithms when applied [12] [13].



2. Literature Review

2.1. Object Detection

Object detection is an activity that aims to gain an understanding of the classification, concept estimation, and location of objects in an image. As one of the basic computer vision problems, object detection can provide valuable information for semantic understanding of images and videos, and is associated with many applications, including image classification [14].

2.2. Artificial Intelligence

Artificial Intelligence is a simulation of human intelligence that is modeled in a machine and programmed to be able to think like humans. Artificial intelligence is a technology that requires data to be used as knowledge so that the intelligence made can be even better so that it can continue to grow and learn from previous mistakes [3].

Artificial intelligence can do *self-correction* is because artificial intelligence designed to learn from the mistakes that have been experienced. Artificial intelligence is one of the following four factors, namely: acting humanly, thinking humanly, thinking rationally, and acting rationally [4].

2.3. Machine Learning

Machine learning can be defined as computer applications and mathematical algorithms adopted using learning that comes from data and produces predictions in the future. The learning process in question is an attempt to acquire intelligence through two stages, including *training* and *testing*. The field of *machine learning* deals with the question of how to build computer programs to improve automatically based on experience [15].

2.4. You Only Look Once

You Only Look Once (YOLO) is an algorithm for object detection based on *Convolutional Neural Network*. In the YOLO architecture, there are 24 *convolutional layers* that function to get features from the image. Then followed by 2 *connected layers* which function to predict probability and coordinates [12].

2.5. You Only Look Once v3

Single-stage architecture created is called the YOLO (You Only Look Once) method which produces fast inference time. The frame rate for a 448 x 448-pixel image is 45 fps (0.022 seconds per image) on the Titan X GPU while achieving advanced mAP (precision average). Yolov3 has several stages in classifying detection YOLOv3 feature extraction uses the darknet to predict the class and location of objects, after which YOLOv3 will classify objects according to their class [16] [17].

2.6. Python

Python is a high-level programming language that can execute a number of multi-use instructions directly (*interpretively*) with the *Object Oriented Programming method* and also uses dynamic semantics to provide a level of syntax readability. As a high-level programming language, python can be learned easily because it has been equipped with automatic memory management [18].

2.7. Tensorflow

Tensorflow is a software library or *library* that is *open source* or open, and free for machine learning. *Tensorflow* is used for many things but focuses more on training and inference of deep neural Tensorflow library is a library based on dataflow and programming [19] [20].

Tensorflow is a computational framework for building machine learning models. *Tensorflow* provides a variety of toolkits that allow you to build models at your preferred level of abstraction and run graphics on multiple hardware platforms, including CPU, GPU, and TPU [20].

3. Methods

3.1. Object Analysis

The detection carried out leads to random objects around the researcher. Light intensity is also taken into account.

3.2. System Overview

To obtain object information, a system will be built, where system will recognize the name of the object taken from the webcam, then the results will detect the name and accuracy of the random object. This system will be built using the python language where the user must run through the *Anaconda prompt* and then continue using *Jupyter Notebook*, after that when the system displays the camera screen, the user must scan the object so that the camera can capture the object and generate information on the object.

Next, there is a training stage where all datasets that are used as training data will be trained using the YOLO (You Only Look Once) method. All data will be recognized so that the system can detect objects accurately and accordingly.

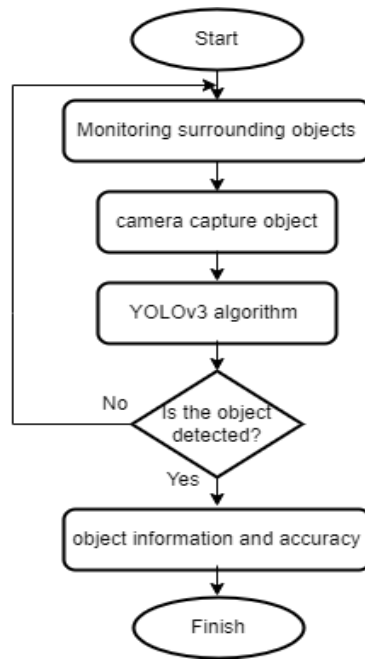


Fig 1. Flowchart of the Overall System

Figure 1 shows the system flow diagram in this study. Where the camera will monitor objects around. When the camera captures an object, the captured object will be processed using the YOLOv3 algorithm for identification. If detected by the camera, the object will be marked with a bounding box on the displayed display and the information and accuracy of the object will be known if the system cannot identify the detected object, the system will repeat the command to monitor the surrounding situation again.

4. Results and Discussion

In the application of machine learning, you can learn various forms of visual random objects from colors, shapes, textures, and images. Problems that are often encountered in object detection are the difficulty of detecting objects or non-objects in an image, and the high number of object variability as is the case with random objects that have various shapes, colors, and sizes that vary.

4.1. System Design

In this stage, the system design will be carried out using Unified Modeling Language (UML) diagrams including use case diagrams, activity diagrams, and class diagrams.

4.1.1. Use Case Diagram

Description of use cases on this system.

1. Scan Object is a feature that can be done by the user before getting the results of random object detection, the user must point a random object at the camera, and then it will be detected.
2. Detecting Object is a process where the object captured by the camera will be recognized to get the result in the form of the object name and its accuracy.
3. Viewing Object Detection Results is a feature where users can see directly the results of object detection that appear, complete with object names and percentage accuracy.

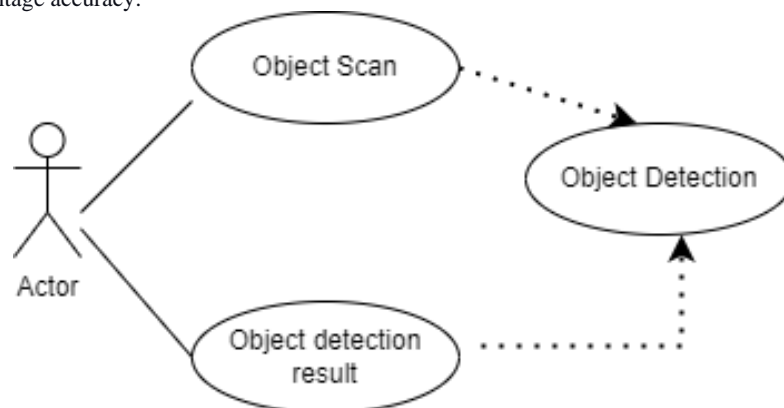


Fig 2. Use Case Diagram

4.1.2. Activity Diagram

A diagram that shows the activities of each function, which describes the workflow (workflow) of a system and can describe the menu activities that exist in the system.

1. Object Scan

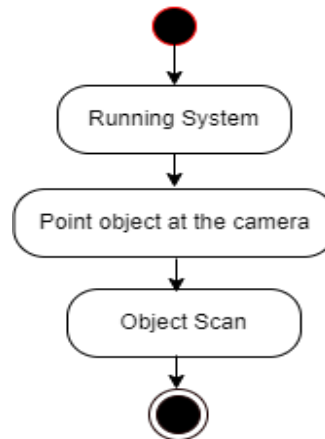


Fig 3. Activity Diagram of the Scan Object Process

The object scan process, where the first object detection system will be run then the user directs the object to the camera. Then the object scan process will run.

2. Object Detection

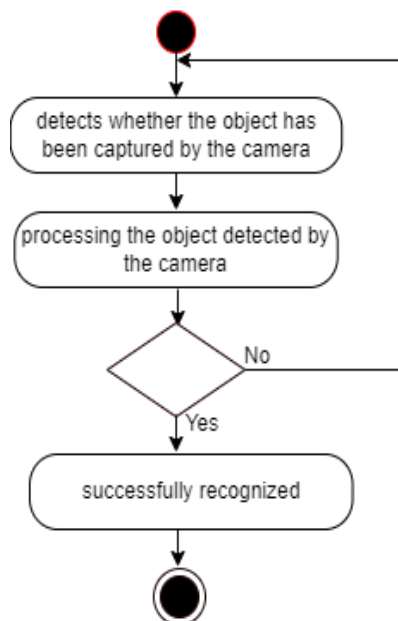


Fig 4. Activity Diagram of the Object Detection Process

The object detection process, the first process detects whether the object was successfully captured by the camera there are 2 conditions in this process, which is whether the system can recognize the object or not, if not, the system will return to the initial step, which is detecting the object to be captured by the camera.

3. Object Detection Results

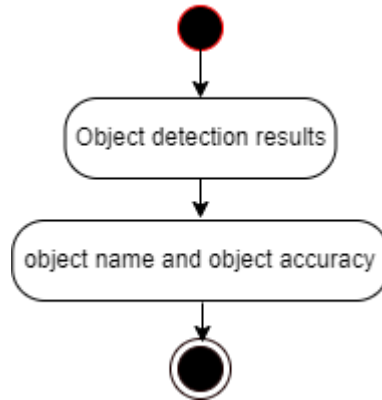


Fig 5. Activity Diagram of the Object Detection Result Process

The process of object detection results, where after the system obtains the results, the system will display or provide the name of the object and the percentage of object accuracy.

4. Managing Object Information

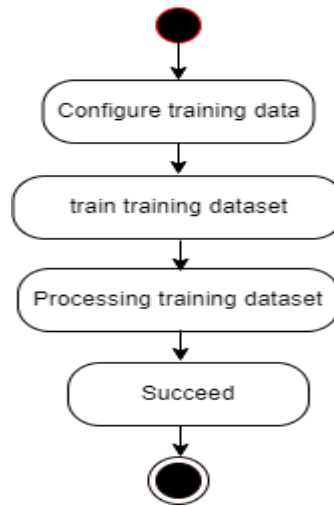


Fig 6. Activity Diagram of the Process of Managing Object Information

4.3.3 Class Diagram

Class diagrams describe the static structure of the classes in the system and describe the attributes, operations, and relationships between classes. Class diagrams help in visualizing the class structure of a system and are the most widely used type of diagram.

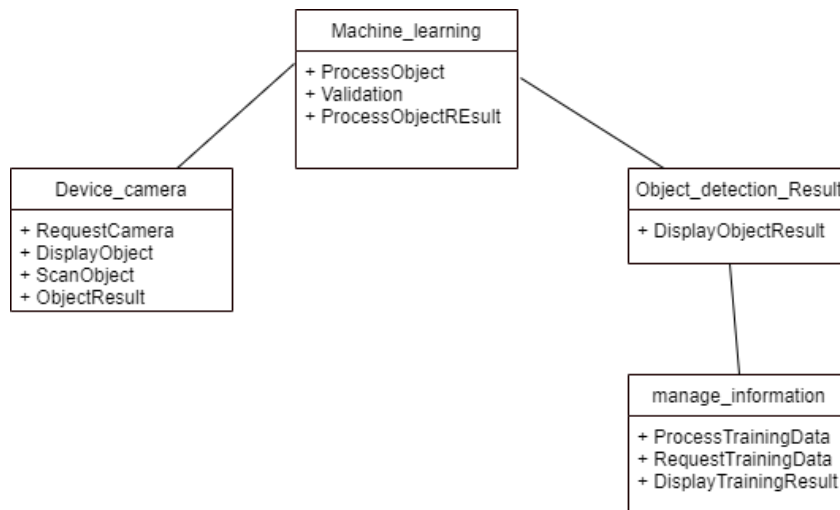


Fig 7. Class Diagram

5. Conclusion

It can be concluded that the system design has been carried out using *Unified Modeling Language* (UML) diagrams including use case diagrams, activity diagrams, and class diagrams. This system will be built using the python language where the user must run through the *Anaconda prompt* and then continue using *Jupyter Notebook*. The object under study is the object that is around the researcher. The result of object detection will display the name and *bounding box* with the percentage of accuracy. In this study, the system is also able to recognize objects when they object is stationary or moving.

References

- [1] C. Fajar, "Tax Law Enforcement Sebagai Upaya Optimalisasi Penerimaan Pajak Negara Ditinjau Melalui Tingkat Kepatuhan Wajib Pajak," *Ecodemica*, 2014.
- [2] M. Adjouadi and F. Candocia, "A Stereo Matching Paradigm Based on the Walsh Transformation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 12, 1994, doi: 10.1109/34.387486.
- [3] M. K. Md Dahlan, N. Abdullah, and A. I. H. Suhaimi, "A study on supporting factors of digital workplace diffusion in public sector," 2018, doi: 10.1007/978-981-13-1628-9_29.
- [4] R. A. T. M. Ranasinghe, M. B. Jaksa, F. Pooya Nejad, and Y. L. Kuo, "Genetic programming for predictions of effectiveness of rolling dynamic compaction with dynamic cone penetrometer test results," *J. Rock Mech. Geotech. Eng.*, vol. 11, no. 4, 2019, doi: 10.1016/j.jrmge.2018.10.007.
- [5] A. Asrianda, H. A. K. Aidilof, and Y. Pangestu, "Machine Learning for Detection of Palm Oil Leaf Disease Visually using Convolutional Neural Network Algorithm," *J. Informatics Telecommun. Eng.*, vol. 4, no. 2, pp. 286–293, 2021, doi: 10.31289/jite.v4i2.4185.
- [6] Q. Wang, D. Wu, G. Li, and W. Gao, "A virtual model architecture for engineering structures with Twin Extended Support Vector Regression (T-X-SVR) method," *Comput. Methods Appl. Mech. Eng.*, vol. 386, p. 114121, 2021, doi: <https://doi.org/10.1016/j.cma.2021.114121>.
- [7] L. Nguyen-Vu, J. Ahn, and S. Jung, "Android Fragmentation in Malware Detection," *Comput. Secur.*, vol. 87, 2019, doi: 10.1016/j.cose.2019.101573.
- [8] R. I. Tiyyar and D. H. Fudholi, "Kajian Pengaruh Dataset dan Bias Dataset terhadap Performa Akurasi Deteksi Objek," *Petir*, vol. 14, no. 2, pp. 258–268, 2021, doi: 10.33322/petir.v14i2.1350.
- [9] Nurdin, D. Hamdhana, and M. J. Setiawan, "Sistem Pendeteksi Pola Lafadz Allah Dan Muhammad Pada Citra Al-Qur'an Menggunakan Metode Peirce," *e-Journal Techsi Tek. Inf.*, vol. 9, no. 2, pp. 78–90, 2017.
- [10] Y. Mujiaty, D. Novayanti, H. Khotimah, and Hanifah, "PENGOLAHAN DATA AKUNTANSI MENGGUNAKAN SISTEM ZAHIR VERSI 5.1 PADA PT.MENARA WARNA INDONESIA JAKARTA," *AKRAB JUARA*, vol. 5, no. 2, 2020.
- [11] G. V. Shalini, M. K. Margret, M. J. S. Niraimathi, and S. Subashree, "Retraction: Social Distancing Analyzer Using Computer Vision and Deep Learning," *Journal of Physics: Conference Series*, vol. 1916, no. 1. 2021, doi: 10.1088/1742-6596/1916/1/012039.
- [12] L. Agustien *et al.*, "Real-time Deteksi Masker Berbasis Deep Learning menggunakan Algoritma CNN YOLOv3," vol. 8, no. 2, pp. 129–137, 2021.
- [13] J. R. Parker, *Algorithms for image processing and computer vision*. John Wiley & Sons, 2010.
- [14] F. Leite, A. Akcamete, B. Akinci, G. Atasoy, and S. Kiziltas, "Analysis of modeling effort and impact of different levels of detail in building information models," *Autom. Constr.*, 2011, doi: 10.1016/j.autcon.2010.11.027.
- [15] A. Roihan, P. A. Sunarya, and A. S. Rafika, "Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper," *IJCIT (Indonesian J. Comput. Inf. Technol.)*, vol. 5, no. 1, pp. 75–82, 2020, doi: 10.31294/ijcit.v5i1.7951.
- [16] Q. C. Mao, H. M. Sun, Y. B. Liu, and R. S. Jia, "Mini-YOLOv3: Real-Time Object Detector for Embedded Applications," *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2941547.
- [17] Z. Zheng, J. Zhao, and Y. Li, "Research on Detecting Bearing-Cover Defects Based on Improved YOLOv3," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3050484.
- [18] M. K. Rahmadhika, A. M. Thantawi, and U. P. I. Y. A. I, "Rancang Bangun Aplikasi Face Recognition Pada Pendekatan CRM Menggunakan Opencv Dan Algoritma Haarcascade," vol. 5, no. 7, pp. 109–118.
- [19] Y. Primatama, A. E. Rhamadani, F. D. Ramtomo, D. Cahya, and P. Buani, "MENGGUNAKAN PEMINDAI WAJAH BERBASIS ANDROID," pp. 59–65, 2018.
- [20] N. H. A. E and M. I. Zul, "Aplikasi Penerjemah Bahasa Isyarat Indonesia menjadi Suara berbasis Android menggunakan Tensorflow," vol. 7, no. 1, pp. 74–83, 2021.