# Implementation of YOLOv5 Architecture for Clothing Detection Systems

Naula Qisty Modjo
*Dept. of Electrical Engineering*
*Sam Ratulangi University*
Manado, Indonesia
naulamodjoo@gmail.com

Jane Litouw
*Dept. of Electrical Engineering*
*Sam Ratulangi University*
Manado, Indonesia
jane_litouw@unsrat.ac.id

Febriyanti Ludja
*Magister of Informatics*
*Sam Ratulangi University*
Manado, Indonesia
febriyantiludja124@gmail.com

***Abstract*— Object detection is one of the key areas in computer vision that plays a crucial role in processing and analyzing visual data. In various applications such as clothing recognition, object detection is instrumental in identifying and localizing objects in image or videos. This research utilizes one of the Convolutional Neural Network (CNN) architectures, YOLOv5n, to develop an effective framework for clothing detection. The objective is to enchance the performance of YOLOv5n in terms of accuracy while ensuring applicability to low-cost devices. Additionally, a new clothing dataset is curated for this purpose. The study leverages CPU-based cameras for real-time object detection. By modifying the SPPF module within the YOLOv5n architecture, high accuracy is achieved with parameters totaling 7,086,779, mAP of 0,685, and GFLOPS of 8,3.***

**Keywords—** ***Clothing Detection; CNN; Real-time; YOLOv5n***

## I. INTRODUCTION

Clothing, or the appearance of clothing, is a fundamental aspect of daily human life and personal expression [1]. Technological advancements have played a significant role in enhancing the efficiency and convenience of various human activities. In the context of e-commerce, the integration of intelligent methods to assist with clothing combination can streamline the process of selecting and organizing garments, tasks that are often time-consuming and influenced by diverse user preferences. Apparel detection can enhance the online shopping experience by enabling consumers to visualize how garments would appear on their bodies before purchase. To facilitate the integration of such functionality in e-commerce platforms, accurate clothing detection is required at an early stage. To achieve this, deep learning techniques are employed to ensure reliable and precise detection results [2].

Deep learning is a modern machine learning approach that mimics the structure and function of neural networks in the human brain. One such architecture is the artificial neural network, which simulates the behavior of biological neurons to perform prediction tasks at the output layer [3]. Deep learning methods are capable of effectively extracting relevant features and high-level representations from data, making them particularly well-suited for accurate clothing detection and classification tasks [4], [5].

The deep learning method employed in this study is the Convolutional Neural Network (CNN), a class of neural networks widely used for processing image data. Among the various CNN-based architectures, this work utilizes YOLOv5n (You Only Look Once version 5n) [6], which is known for its efficiency and real-time object detection capabilities. This research aims to enhance the YOLOv5n architecture to achieve high performance in clothing detection, measured using mean Average Precision (mAP) [7]. The model is designed to be efficient enough for deployment on low-cost devices. In addition, a new clothing dataset is created, consisting of specific categories such as short-sleeved shirts, long-sleeved shirts, long pants, shorts, sweaters, long skirts, short skirts, long dresses, short dresses, ties, hats, and vests.

The Convolutional Neural Network (CNN) is a deep learning algorithm capable of learning from both new and existing datasets by optimizing millions of parameters. It processes 2D image inputs through convolutional layers, where filters are applied to extract important features, ultimately producing the desired output [8], [9].

One of the most widely recognized deep neural network architectures is the Convolutional Neural Network (CNN), named for its use of a linear mathematical operation known as convolution, performed between input data and learnable filters. A typical CNN consists of several key layers, including convolutional layers, non-linear activation layers, pooling layers, and fully connected layers. While convolutional and fully connected layers contain learnable parameters, pooling and non-linear activation layers do not. CNNs are well known for their outstanding performance in addressing a wide range of machine learning and computer vision problems [8].

The CNN architecture for image classification begins with an input image associated with a set of target classes. The image is first processed through convolutional layers to extract relevant features, followed by activation functions to introduce non-linearity [10]. The resulting feature maps are then passed through pooling layers to reduce dimensionality and computational complexity. Finally, the output is fed into fully connected layers to produce the final class predictions [11].

YOLO is an object detection framework based on CNN. This architecture processes the entire image in a single pass, simultaneously predicting the object class and bounding box location [12]. This approach uses a single forward propagation, enabling real-time object detection by combining classification and localization tasks into a single unified network [7].

Object detection is analyzing a digital image to identify and determine the presence of specific objects. This process involves various techniques that focus on extracting and

Received: 20 June 2025, Revision: 15 September 2025, Accepted: 19 November 2026
International Journal of Informatics, Data, and Emerging Applications, Vol 1, No 1, Januariy – June, 2026, pp. 8-12
available at: https://ejournal.unsrat.ac.id/v3/index.php/idea/

interpreting the features of objects within the image or video [13], [14]. The primary goal of object detection is to distinguish and localize the target objects (foreground) from the surrounding context (background) in an image or video sequence [15].

Object detection is an advanced computer vision technique that aims to recognize and label objects within images, videos, and real-time footage. Object detection models are trained on annotated visual datasets, enabling them to accurately identify and classify objects in new, unseen data. The overall process can be simplified into two main steps: receiving a visual input and generating a labeled visual output [16].

At the core of object detection lies the bounding box, which delineates the spatial extent of the detected object within an image. Each bounding box is typically associated with a class label that describes the identified object, such as clothing, people, vehicles, and so on. In scenes containing multiple objects, bounding boxes may overlap to accurately capture the presence of several instances, depending on the model's learned representation and prior knowledge of the object categories [16].

## II. METHODOLOGY

### A. *Experiment*

The research on designing an object detection system using YOLOv5 begins with the following steps:

1. Collecting relevant reference data and information related to the proposed system model.
2. Preparing the necessary hardware and software tools to develop the YOLOv5 architecture.
3. Creating a new dataset tailored to the application.
4. Designing and implementing the training program.
5. Fine-tuning the YOLOv5 architecture to optimize accuracy and efficiency.
6. Deploying the model by testing the tuned architecture on low-cost devices.
7. Analyzing the optimization in terms of frames per second (FPS) or delay, as well as evaluating the detection accuracy of the developed model.
8. Preparing the final research report.

### B. *System Architecture*

In developing an object detection system using YOLOv5, a clear and well-defined conceptual framework is essential to ensure that the results are in line with the desired objectives. This framework serves as a strategic guide for designing an optimal system, outlining the necessary steps, and determining the supporting components required for effective implementation. The dataset used in this study was specifically designed to support the training and evaluation of object detection models. The data was obtained from various digital media, including mobile device recordings and publicly available online video platforms such as YouTube and Pinterest. Each video was extracted into individual frames, then further processed to generate a dataset of clothing images.

Overall, 3,000 labeled images with a minimum resolution of 367×652 pixels and a maximum resolution of 6000×3376 pixels were collected, covering a range of quality from low to high. This dataset was divided into 600 images for training and 2,400 images for testing, with the appropriate division applied to the label files. To minimize learning bias, the number of samples in each class was balanced. This dataset plays a critical role in ensuring that the training data is clean, relevant, and representative, thereby improving the model's detection performance. The dataset creation process began with data collection, which involved several classes as shown in Figure 1. This was followed by annotation and label assignment according to predetermined categories. After dataset construction, image preprocessing is performed prior to training. The training process is conducted on the Ubuntu operating system using a CNN architecture, and the model weights are extracted upon completion. The model that achieves the highest accuracy and efficiency is then deployed on a low-cost device for real-time inference using a webcam. Figure 2 illustrates two main stages: the training stage and the testing stage. In the training stage, the machine learning process mimics human learning by using examples to recognize patterns and respond to related cases. The first step involves introducing the dataset, specifically, the preprocessed *General Clothing* dataset, which ensures that the image data used for training is clean, relevant, and capable of producing optimal results. During this stage, an efficient YOLOv5 architecture is also designed and implemented to enhance the performance of the object detection model.

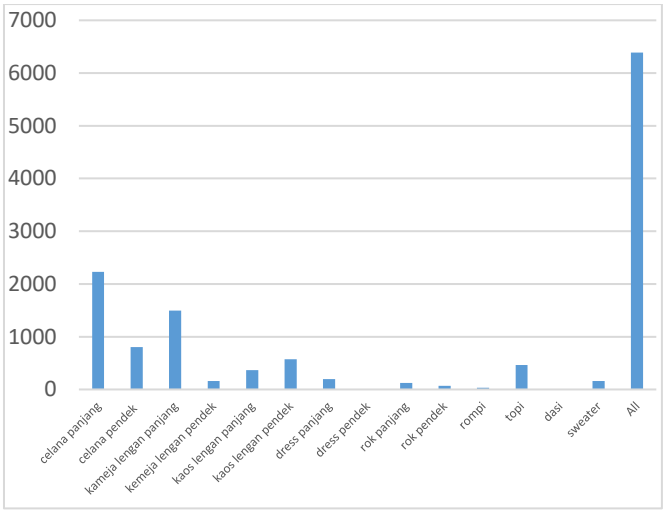


Figure 1. Number of Images Created for the Dataset

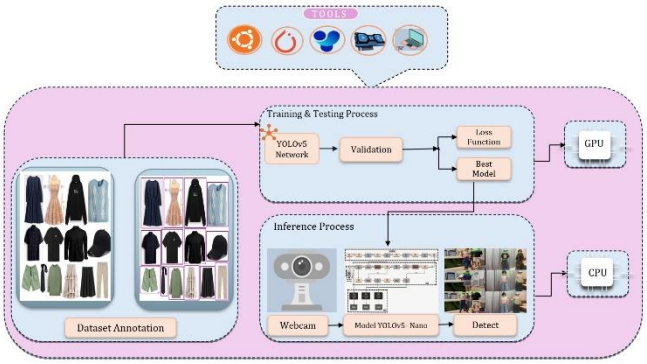


Figure 2 illustrates the learning and testing processes involved in the system workflow.

The research began with the compilation of a dataset, which involved collecting clothing images and annotating them with bounding boxes and category labels. This

annotation step was essential to provide the model with clear information about the target objects. The annotated dataset was subsequently used for training and testing. The data were fed into the YOLOv5 network and trained using specific parameters, including the number of epochs, batch size, image size, and learning rate. During training, validation was performed to evaluate the model's performance, while a loss function was employed to calculate the prediction error rate. The best-performing model from this process was saved for subsequent stages. To ensure computational efficiency, the

After obtaining the best model, the research proceeded to the inference stage. Test data was collected via webcam and processed using the YOLOv5-Nano model. The model generated detections in the form of bounding box locations and corresponding clothing classes in the image. This inference process can be run on a CPU, enabling evaluation in real-world conditions using commonly available computing devices. The workflow was implemented on the Ubuntu operating system, with PyTorch as the deep learning framework and Ultralytics providing the official YOLOv5 implementation. Model training was performed on a GPU-based PC, while inference was performed on a CPU-based laptop.

Table 1 Speed test obtained using a laptop during the model's inference process.

| MODEL | PARAMETER | GFLOPS | mAP:050 | mAP05:95 | FPS |
|---|---|---|---|---|---|
| YOLOv5n | 1,778,107 | 4,3 | 0,75 | 0,567 | 31,10 |
| YOLOv5n | 1,778,107 | 4,3 | 0, 761 | 0,59 | 31,35 |
| YOLOv5n | 1,778,107 | 4,3 | 0,797 | 0,646 | 31,80 |
| YOLOv5n | 1,778,107 | 4,3 | 0,82 | 0,669 | 31,14 |
| **YOLOv5n-modify** | **7,086,779** | **8,4** | **0,824** | **0,685** | **25,90** |

In this experiment, the system was configured with Ubuntu 22.04.3 as the operating system, 32 GB of RAM, and an Intel Core i7-13700KF CPU. To accelerate model training and inference, a 24 GB NVIDIA RTX 4090 GPU was utilized as the graphics processing unit.The proposed network was implemented using PyTorch and leveraged CUDA 12.2 to enable efficient GPU acceleration. The training process was conducted over 200 epochs to ensure sufficient iterations for effective object detection. During training and evaluation, the input image size was set to 960 × 960 pixels, with a learning rate of 0.001 and a batch size of 64. A loss function was employed to quantify the difference between the predicted bounding boxes and object classes and the ground truth annotations. The model's performance was evaluated on an iPhone 14 Pro connected to a deployment PC with 6 GB of RAM during the testing and deployment stages. This setup enabled the assessment of the model's compatibility and efficiency in real-world scenarios with limited computational resources. Model performance was measured using the mean average precision (mAP), calculated based on the Intersection over Union (IoU) thresholds of 0.5 and the range 0.5:0.95.

## III. RESULTS AND DISCUSSION

Clothing detection performance is improved by modifying the Spatial Pyramid Pooling-Faster (SPPF) module. This change expands the receptive field by applying sequential pooling using a 5×5 kernel, followed by additional convolution operations after concatenation. A 7×7 convolution is first applied to capture broader contextual information, while 5×5 and 3×3 convolutions are subsequently used to refine and integrate features from different pooling scales. This improvement strengthens the model's feature extraction capabilities, resulting in higher detection accuracy compared to the base YOLOv5n model.

Table 1 presents a comparison between the original and modified versions of the YOLOv5n object detection model. It includes results for the original model using input image sizes of 320×320, 460×460, 640×640, and 960×960 pixels, as well as the modified model evaluated with 960×960 images. The mean Average Precision (mAP) metric, which is widely used to assess object detection performance, is reported; higher mAP values indicate better detection accuracy. Here, mAP is measured at IoU threshold values of 0.5 and 0.95. Models with fewer parameters are generally preferred, as they tend to be lighter and more computationally efficient. The number of epochs, representing the total iterations over the entire dataset during training, is also reported.

The last column displays the detection speed, measured in FPS, showing how many images are processed per second when the model is applied in an application. Versions trained with smaller input image sizes achieve higher values, indicating faster object detection performance.

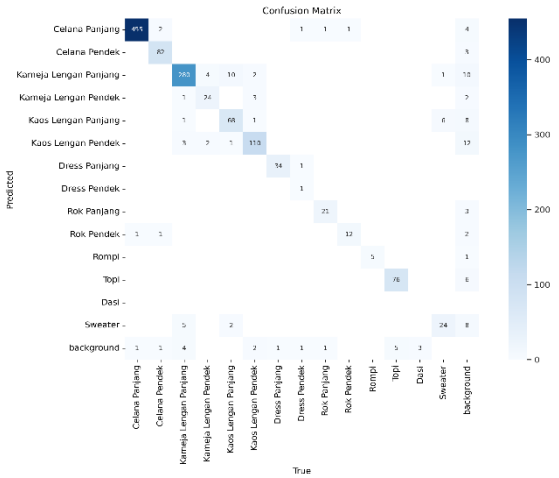


Figure 3. Confusion matrices on evaluation dataset.

Figure 4. Visualization of the model's validation results before the modification process was applied.

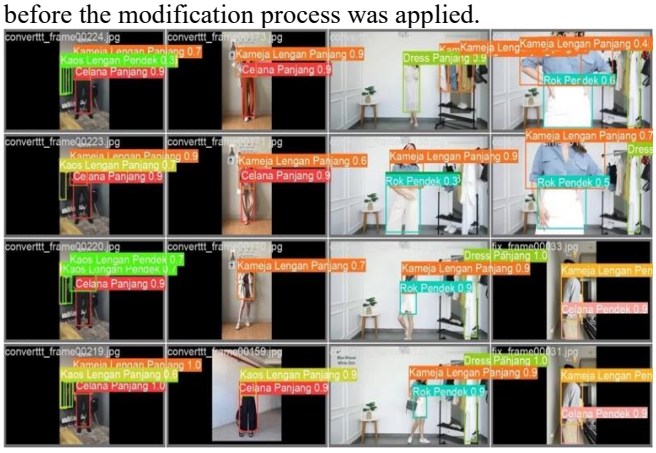

Figure 5. Visualization of the model's validation results following the modification process.

Figure 6. Results of performance testing in real-time implementations conducted on a laptop device.

The confusion matrix in Figure 3 shows that the model achieves high accuracy in several categories, including 455 correct predictions for long pants, 110 for short-sleeved shirts, and 76 for hats. However, classification errors were observed in visually similar classes; for example, long-sleeved shirts were often misclassified as collared long-sleeved shirts, and sweaters were sometimes misclassified as backgrounds. These findings indicate that the model performs well on categories with distinct visual features but faces challenges in distinguishing clothing types with overlapping characteristics, highlighting the importance of a balanced dataset and improved feature extraction strategies.

Figures 4 and 5 illustrate the prediction results before and after modification of the YOLOv5n architecture. Figure 4 shows the validation results of the original, unmodified YOLOv5n model, which exhibits a relatively low mAP. Consequently, some objects in the images are either undetected or misclassified. In contrast, Figure 5 presents the validation results of the modified YOLOv5n architecture, which achieves a higher mAP, resulting in more accurate detections with correct class predictions across all images. Figure 6 visually demonstrates the detection results overlaid on images of general environments to analyze model performance. The accuracy of the modified YOLOv5n model was tested on a CPU by evaluating its ability to detect humans. During CPU-based testing, the model's performance is influenced by the computational efficiency of the CPU. Additionally, detected objects are accurately localized and highlighted with color-coded bounding boxes corresponding to their respective classes, emphasizing significant features. The proposed model in this study demonstrates improved accuracy in object identification.

The proposed system exhibits several shortcomings, such as incorrect object detection, which may indicate potential errors in the detection process or difficulties in object identification. Additionally, the system struggles with detecting overlapping objects, likely due to the model's limitations in understanding contextual information or accurately localizing objects when there is a high degree of similarity among learned features.

## IV. CONCLUSIONS AND FUTURE WORK

### *A. Conclusion*

Based on research and discussion regarding the implementation of the YOLOv5 architecture for clothing detection systems, several conclusions can be drawn. A new dataset consisting of 14 object classes was created, and this architecture was implemented to optimize the clothing detection process. The most suitable architecture identified was YOLOv5n-Modify. However, this system still has several limitations, including the occurrence of double bounding boxes and potential classification errors due to the existence of several classes with similar visual features. This modification achieved the highest detection accuracy, with a mAP@0.5 of 0.824 and a mAP@0.5:0.95 of 0.685, compared to the original YOLOv5n, which only achieved a maximum mAP@0.5 of 0.82 and a mAP@0.5 of 0.669. This improvement shows that the modification enhances the model's ability to detect clothing objects more accurately. However, the modified architecture has more parameters, namely 7,086,779, and higher computational costs of 8.4 GFLOPs, resulting in a slower inference speed of 25.90 FPS compared to more than 31 FPS in the original model. These results show that the proposed modification can improve detection accuracy, in line with the research objective of enhancing the performance of the nano variant. However, the

trade-off between accuracy and real-time speed highlights the need for further optimization to maintain efficiency when applied to low-cost devices.

### B. Future Work

This research still has several aspects that require further evaluation to enhance its quality in future work. For future researchers, it is recommended that when constructing a dataset, the number of images per class should be balanced to ensure that the model can learn all classes effectively. Additionally, the architecture can be further developed to achieve higher accuracy while reducing computational complexity. Moreover, this system has the potential to be expanded into a fully functional application for practical deployment.

## ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the AIVISION research team for their invaluable support and expertise in computer vision and deep learning. Their provision of computational resources and insightful feedback significantly contributed to the experimental work, methodological improvements, and the overall quality of this manuscript.

## REFERENCES

[1] A. Medina, J. I. Méndez, P. Ponce, T. Peffer, A. Meier, and A. Molina, 'Using deep learning in real-time for clothing classification with connected thermostats', *Energies*, vol. 15, no. 5, p. 1811, 2022.

[2] X. Han, D. Zheng, and D. Wang, 'An Enhanced Clothing Detection Model for E-Commerce Applications', in *2024 39th Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2024, pp. 1181–1186.

[3] A. Shah *et al.*, 'A comprehensive study on skin cancer detection using artificial neural network (ANN) and convolutional neural network (CNN)', *Clinical eHealth*, vol. 6, pp. 76–84, 2023.

[4] M. C. Wujaya and L. W. Santoso, "Klasifikasi Pakaian Berdasarkan Gambar Menggunakan Metode YOLOv3 dan CNN," *J. Infra*, vol. 9, no. 1, Art. no. 1, Apr. 2021.

[5] H. T. Nguyen, K. K. Nguyen, P. T.-N.-Diem, and T. T.-Dien, 'Clothing Detection and Classification with Fine-Tuned YOLO-Based Models', in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 2023, pp. 127–132.

[6] J. Suttanuruk, S. Jomnonkwao, V. Ratanavaraha, and S. Kanjanawattana, "Convolutional Neural Network for Overcrowded Public Transportation Pickup Truck Detection," *Comput. Mater. Contin.*, vol. 74, no. 3, pp. 5573–5588, 2022, doi: 10.32604/cmc.2023.033900.

[7] L. Yipeng and W. Junwu, 'Personal protective equipment detection for construction workers: a novel dataset and enhanced YOLOv5 approach', *IEEE Access*, vol. 12, pp. 47338–47358, 2024.

[8] K. Azmi, S. Defit, and S. Sumijan, "Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat," *J. UNITEK*, vol. 16, no. 1, Art. no. 1, Jun. 2023, doi: 10.52072/unitek.v16i1.504.

[9] Y. Liu, H. Pu, and D.-W. Sun, 'Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing complex food matrices', *Trends in Food Science & Technology*, vol. 113, pp. 193–204, 2021.

[10] Y. Wang, Y. Li, Y. Song, and X. Rong, 'The influence of the activation function in a convolution neural network model of facial expression recognition', *Applied Sciences*, vol. 10, no. 5, p. 1897, 2020.

[11] A. Purnomo and H. Tjandrasa, "Improved Deep Learning Architecture With Batch Normalization For Eeg Signal Processing," *J. Ilm. Teknol. Inf.*, vol. 19, no. 1, pp. 19–27, Jan. 2021, doi: 10.12962/j24068535.v19i1.a1023.

[12] J. Redmon and A. Angelova, 'Real-time grasp detection using convolutional neural networks', in *2015 IEEE international conference on robotics and automation (ICRA)*, 2015, pp. 1316–1322.

[13] S.-H. Tsang, "Brief Review: YOLOv5 for Object Detection," Medium. Accessed: Apr. 28, 2024. [Online]. Available: https://sh-tsang.medium.com/brief-review-yolov5-for-object-detection 84cc6c6a0e3a.

[14] S. Rani, D. Ghai, and S. Kumar, 'Object detection and recognition using contour based edge detection and fast R-CNN', *Multimedia Tools and Applications*, vol. 81, no. 29, pp. 42183–42207, 2022.

[15] "Object detection," *Wikipedia*. Nov. 26, 2023. Accessed: Apr. 28, 2024.[Online].Available:https://en.wikipedia.org/w/index.php?title=Object_detection&oldid=1186971351.

[16] R. Lantang, A. Jacobus, and S. Sompie, "Pemanfaatan Image Hashing Pada Klasifikasi Penyakit Kulit Terhadap Citra yang Terduplikasi," *Jurnal Teknik Informatika*, vol. 19, no. 02, Mar. 2024, doi: https://doi.org/10.35793/jti.v19i02.53408.

[17] S. Moitra and S. Biswas, "Object Detection in Images: A Survey," *Int. J. Sci. Res. IJSR*, vol. 12, no. 4, pp. 10–29, Apr. 2023, doi: 10.21275/SR23330184650.

[18] M. D. A. Hasan, T. Bhargav, V. Sandeep, V. S. Reddy, and R. Ajay, 'Image classification using convolutional neural networks', *International Journal of Mechanical Engineering Research and Technology*, vol. 16, no. 2, pp. 173–181, 2024.

[19] C. Yang, W. Tian, and L. Zhang, "An Improved Target Detection Algorithm model for Garment image Detection," in *2022 21st International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, Oct. 2022, pp. 99–102. doi: 10.1109/DCABES57229.2022.00014.

[20] F. M. Qotrunnada and P. H. Utomo, "Metode Convolutional Neural Network untuk Klasifikasi Wajah Bermasker," *PRISMA Pros. Semin. Nas. Mat.*, vol. 5, pp. 799–807, Feb. 2022.

[21] "Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper | Roihan | IJCIT (Indonesian Journal on Computer and Information Technology)." Accessed: Apr. 26, 2024. [Online]. Available:https://ejournal.bsi.ac.id/ejurnal/index.php/ijcit/article/view/7951/0.

[22] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, p. 53, Mar. 2021, doi: 10.1186/s40537-021-00444-8.

[23] B. A. Septyanto, S. A. Wibowo, and C. Setianingsih, "Implementasi Face Recognition Berbasis Deep Neural Network Sebagai Sistem Kendali Pada Quadcopter," *EProceedings Eng.*, vol. 9, no. 6, Art. no. 6, 2022, Accessed: Apr. 28, 2024. [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/18960.

[24] "PENERAPAN ARTIFICIAL NEURAL NETWORK (ANN) DALAM MEMPREDIKSI KAPASITAS DUKUNG FONDASI TIANG | Dananjaya | Matriks Teknik Sipil." Accessed: Apr. 26, 2024. [Online].Available:https://jurnal.uns.ac.id/matriks/article/view/65034.

[25] "Deep Learning dan Penerapannya dalam Pembelajaran | JIIP - Jurnal Ilmiah Ilmu Pendidikan", Accessed: Apr. 26, 2024. [Online]. Available:https://jiip.stkipyapisdompu.ac.id/jiip/index.php/JIIP/article/view/805.