Real-time Students' Safety Helmet-wearing Detection Based on Convolutional Neural Network

1st Abdi Suryadinata Telaga
Department of Building Construction Engineering
Astra Polytechnic
Jakarta, Indonesia
abdi.telaga@polytechnic.astra.ac.id

2nd Elora Manuella Amei
Department of Informatics Management
Astra Polytechnic
Jakarta, Indonesia
elora.manuella@polytechnic.astra.ac.id

3rd Rifqih Syarial Anwar
Department of Informatics Management
Astra Polytechnic
Jakarta, Indonesia
qihsyahrial@gmail.com

4th Henkhi Krismayanto
Department of Building Construction Engineering
Astra Polytechnic
Jakarta, Indonesia
henkhi.krimayanto@polytechnic.astra.ac.id

***Abstract*—Students often ignore safety helmet-wearing in practice areas with high accident risk. Therefore, the students are exposed to objects falling from above or a hard collision on the head. The accident could be fatal for the student.This study aims to detect students' safety helmets in the practicearea. The method used in this study is Deep Learning with the Convolutional Neural Network YOLO (You Only Look Once) algorithm. The methodology in this research consists of data collection, pre-processing, training, and testing. Firstly, images of students wearing and not wearing helmets from multiple angles were captured during data collection. Then, in the pre-processing stage, the object was labelled using LabelImg. Therefore, the number of annotations that wear helmets is 708, and those that do not are 794. The training stage uses transfer learning darknet-53 YOLOv3-tiny, which is run on Google Collaboratory. At the detection stage, the object recognition process is carried out using the CNN (Convolutional Neural Network) method and the detection process using YOLOv3-tiny. Finally, in the testing phase, the ten-fold-cross validation model for CNN with epoch 50 and batch size 32 was carried out to see the algorithm's performance from the study. The results show that the accuracy is 0.8, the precision is 0.83, the recall is 0.8, and the F1 score is 0.8.**

Keywords— Safety Helmet, Deep Learning, CNN, YOLOv3-tiny.

I. INTRODUCTION

A helmet is a head protector that protects someone's head from impact so they do not get injured. One of the activities of students who wear helmets is during engineeringpractice. Civil engineering students must wear safetyequipment when carrying out construction work. The studentsmust wear safety helmets all the time. However, some students often do not wear safety helmets during practice, even though the behavior is risky and can be fatal if an unwanted incident happens. Lecturers cannot always monitor the students' obedience to wearing helmets. Therefore, to improve students' discipline in wearing a safety helmet, it is necessary to develop real-time safety helmet-wearing detection.

Researchers have been researching the best method to detect safety helmet-wearing. Hayat et al. (2022) stated that YOLO-based architectures could be applied in real-time safety helmet detection [1]. Zhang et al. (2021) utilize k-means and YOLOv5 to develop DWCA-YOLOv5 in detecting safety helmet wearing of the construction workers, and the method has 96,5% accuracy [2]. Zhang et al. (2022) developed SCM-YOLOv4 to improve the accuracy of safety helmet detection in workshop [3]. The technique has an accuracy of 93.19%. Moreover, Cheng et al. (2022) applied SAS-YOLOv3-Tiny to detect multi-scale safety helmets. The accuracy of the method is 80.3%. While the accuracy is lower than YOLOv3 and YOLOv4, YOLOv3-Tiny has a lower processing cost [4]. Further, Deng et al. (2022) developed mini and lightweight YOLOv3 (ML-YOLOv3) to detect safety helmets and found that ML-YOLOv3 has better calculation cost and detection effect than YOLOv5 [5].

Based on the previous research, YOLOv4 and YOLOV5 have better accuracy, but the calculation cost is higher than YOLOv3 tiny. Therefore, considering the practical situation in practice areas where the network is slow and lacks a powerful graphic processor, the YOLOv3 tiny is a suitable algorithm for accurate real-time detection. Further, to evaluate the performance of the algorithm model that has been designed, the model training stage is carried out by implementing the 10-fold cross-validation method.

II. METHODS

The research begins with the data collection stage to identify the use of safety helmets in images and videos using one of the deep learning methods, yolov3. The dataset for training data in this study is a picture of people wearing safety helmets and those not. The amount oftraining data is 448 images consisting of 209 images of people wearing helmets and 239 images of people not wearing helmets.

A. Methodology Flow Chart

The methodology of the system used in this research is shown in the flow chart depicted in Fig. 1. The initial process is to collect data, label images and then conduct training with transfer learning using the YOLOv3-tiny pre-trained model to produce new weights. Furthermore, unknown weights are used to detect students who wear safety helmets and those who do not appear in the trial image.

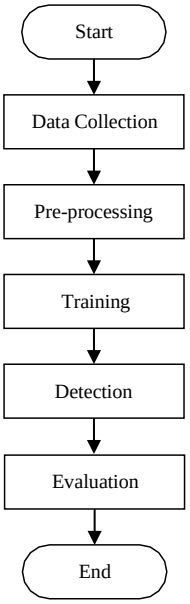

Fig. 1. Methodology flow chart.

B. Pre-Processing

After the dataset is collected, the pre-processing stage is carried out to label the object [7]. Object labelling is divided intotwo classes: wearing helmets and not wearing helmets. The labelling process uses software, namely LabelImg by Tzutalin. After labelling, the annotation results are stored in the YOLO (You Only Look Once) annotation format, which contains <x>, <y>, <width>, and <height>, which are the bounding box prediction values for each detected object in each image. The results of the labelling process of the 448 pictures obtained the number of annotations that wore helmets is 708, and those that did not wear helmets is 794. After the labelling process, extracting data from the image data set and bounding box .txt into a zip was necessary. A dataset must be connected to Google Drive during the training process using Google Collab, so the extracted dataset must be uploaded to Google Drive first—pre-Processing data flowchart depicted in Fig. 2.

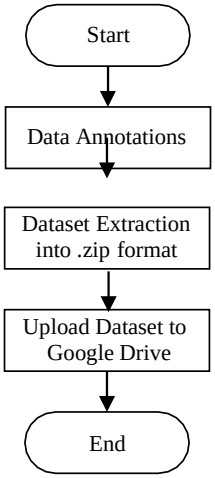

Fig. 2. Pre-Processing data flow chart.

C. Training

The annotated dataset is then used for the training process. In the training process (Fig. 3), the Neural Network will be trained torecognise a pattern in images and annotations to perform object recognition to be considered in the making. Decisions or predictions. The training process uses the darknet-53 architecture as the load model and YOLOv3-tiny as the load weight by using transfer learning to get new weights, namely by using pre-trained model weights trained to recognize new objects [6]. The pre-trained model is the YOLOv3-tiny weight that runs on Google Colaboratory, epoch and batch sizes.

The first step is connecting Google Colab to Google Drive. Next, download the configuration file from the darknet to GitHub and enable GPU, OpenCV and cuDNN. The next step is to configure the darknet with YOLOv3-tiny and then extract the images from the .zip to carry out the training process so that it will generate new weights stored in Google Drive. This training process was carried out for 3 hours.

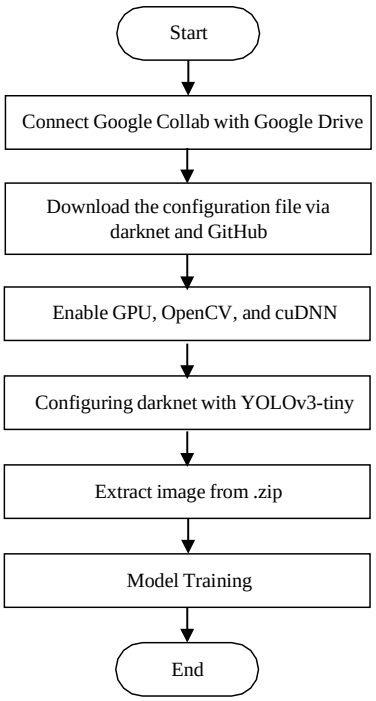

Fig. 3. Training process flow chart.

TABLE I. CONFIGURATION ON DARKNET

Configuration	Description
Load Model	*Darknet-53*
Load Weight	YOLOv3-*tiny*
GPU	Yes
OpenCV	Yes
cuDNN	Yes

TABLE II. CONFIGURATION ON YOLOV3-TINY

Configuration	Description
Batch	64
Subdivisions	32
Max_batches	4000
Width	416
Height	416
Classes	2
Filters	21

The max_batches value is the iteration limit for training. The training process will stop automatically when the iteration reaches a maximum value of 4000. The number of batches determines the number of images to be processed before the network weight is updated; the smaller the batch value in the training process, the faster the training process. However, this is inversely proportional to the level of accuracy that will be produced because the higher thebatch value in the training process, the more features the system will learn. Next, the subdivision processes batch values into small parts or mini-batches. In this study, the batch value is 64, and the subdivision value is32, so the batch value will be divided by 32 to produce a valueof 2, which means that the training process is carried out for two images for each mini batch. This process will occur 32 times until the training process for one batch is complete. Then, the system will switch to the next batch, which also hasa value of 64. At the same time, the class is the number of types that will be predicted.

The max_batches value is obtained from equation 1.

$$\text{max\_batches} = \text{number of classes} \times 2000 \quad (1)$$

The filter value is obtained from equation 2.

$$\text{filters} = (\text{number of classes} + 5) \times 3 \quad (2)$$

D. Detection

At the detection stage, YOLO uses image input with a network size with a multiple of 32. The larger the network size, the more accurately the computer predicts the detection obtained. Still, if the network size provided is more significant, then the computational process on the computer will be slower. Conversely, if the given network size is smaller, the computational process will be faster, but with the consequence that the detection accuracy will be less good. In this study, the network size of 416 × 416.

Furthermore, Feature Extraction is performed using darknet-53. Darknet53 has 53 convolutional layers. The convolutional layer is the first layer in the CNN structure which is the core of CNN and where most computational processes occur. In the feature extraction process, all convolutional layers in Darknet53 use the Leaky Relu activation function except for the last layer, which uses the Linear activation function.

YOLOv3 provides prediction results in 6 values of bounding box coordinates (tx, ty, tw, th), confidence and class probability. In predicting bounding box classes, YOLOv3 uses a multi-label classification. YOLOv3indicates the class score of each bounding box using logistic regression to avoid bounding boxes that overlap with other bounding boxes. If a class score has a high value exceeding confidence, then the class is given to the detected object. The detection process flow chart *is* depicted in Fig. 4.

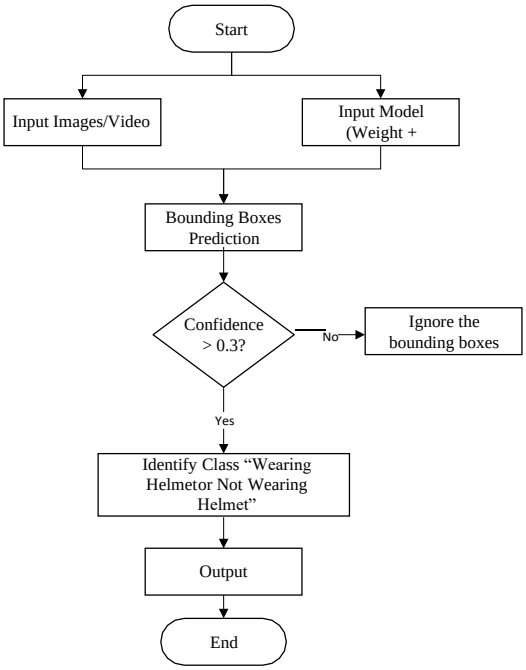

Fig. 4. Detection process flow chart.

E. Evaluation

K-fold cross-validation is one of the statistical methods implemented to evaluate the performance of the model or algorithm that has been designed [8]. The model will be trained using training data and validated k-fold times (Fig. 5). In this study, 10-fold cross-validation was used to evaluate the model's performance. In 10-fold cross-validation, the data is divided into 10-fold of the same size where 9-fold will be used as training data and 1-fold will be used as validation data.

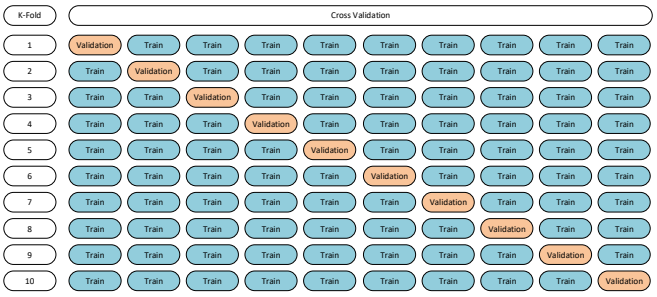

Fig. 5. Illustration of 10-Fold Cross Validation Process.

F. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is a deep learning method to detect, identify, and classify image data. The CNN model is widely used in image processing because of the high level of accuracy that can be achieved [9]. CNN consists of two main parts, namely feature extraction and classification. Moreover, CNN is an architecture composed of several trainable stages. Input and output at each stage have a feature map consisting of several arrays. There are three layers of each stage. The first stage is the convolution stage; at this stage, the number of kernels with a specific size used is determined by the number of features to be produced. The next stage is the activation function at this stage using a linear rectifier unit or ReLu. The last or third stage is the Pooling

layer: At this stage, all the resulting information is summarized by a convolution to reduce dimensions. The process results from these stages are usually called the fully connected layer, typically used to classify an object or a particular problem.

G. *You Only Look Once (YOLO)*

YOLO is a detection system based on the Convolutional Neural Network. The YOLO architecture consists of 24 convolutional layers that obtain features from the image, followed by two connected layers that predict probabilities andcoordinates.

Detect objects using YOLO as follows:

1) Resize the dimensions of the input image to 448 x 448.
2) Running a single convolutional network on the image.
3) Perform a threshold on the detection results based on the confidence value obtained by the model.

YOLO detects the model as a regression, divides the image into grids, and predicts the bounding boxes and confidence in those bounding boxes and probability classes.

H. *Darknet*

The study uses YOLOv3, so the feature extractor is Darknet53 which uses 53 layers. In the feature extraction process, all convolutional layers on Darknet53 use the Leaky Relu activation function except for the last layer, which uses the Linear activation function.

I. *Epoch*

The epoch is a hyperparameter that determines how often the deep learning algorithm works across the entire dataset, both forward and backwards. One epoch is reached when all batches have been successfully passed through the neural network once.

J. *Performance Measurement*

Four measurements are used to measure the ability of the built-in classification system, namely accuracy, precision, recall and F1 score. Accuracy is the ratio of the amount of data correctly predicted to the total amount of data. Precision is the value of the system's accuracy between the information provided to show the negative class or positive class data accurately. A recall is a value that indicates the success of finding the correct information about negative class data or positive text content. The F1 score is the harmonic average between precision and recall [10].

III. Results and Discussion

A. *Results*

This testing stage is the stage to determine how accurate a model that has gone through the training stage is in detecting objects. In this study, testing was carried out using test data that had been prepared. The expected output is the formation of a bounding box that surrounds objects detected wearing safety helmets and those not wearing safety helmets. It displays a confidence score along with the number of students caught.

Testing is done with several scenarios as follows:

1) Performance testing in an outdoor practice area

In this trial, objects were tested in an outdoor practice area with more than one object with several types of object positions to the camera. Object shooting is done from above the object.

Fig. 6. Performance testing in an outdoor practice area.

Fig. 6 shows the result of system implementation in outdoor practice areas. From the results presented, it can be concluded that the system can perform real-time detection well with a confidence value that will change according to movement, distance, lighting, and image sharpness when captured by the system. From the trial results, the highest confidence value was 99%, and the lowest was 23%.

2) Performance testing in an indoor practice area

In this trial, objects were tested in an indoor practice area with more than one object with several types of object positions to the camera.

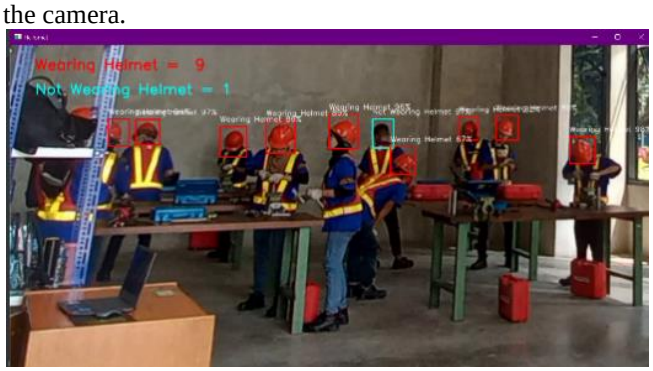

Fig. 7. Performance testing in an indoor practice area.

Fig. 7 shows the result of system implementation in the indoor practice area. From the figure presented, it can be concluded that the system can perform real-time detection properly with a confidence value that will change according to movement, lighting and image sharpness when captured by the system. From the trial results, the highest confidence value is 99%, and the lowest is 67%.

3) Performance testing on standard condition objects

Objects with normal conditions are objects that wear helmets and those that do not. Fig. 8 shows the result of system implementation on the standard state. Objects wearing helmetsproduce a confidence value of 99%. Things that do not wearhelmets have a confidence value of 97%.

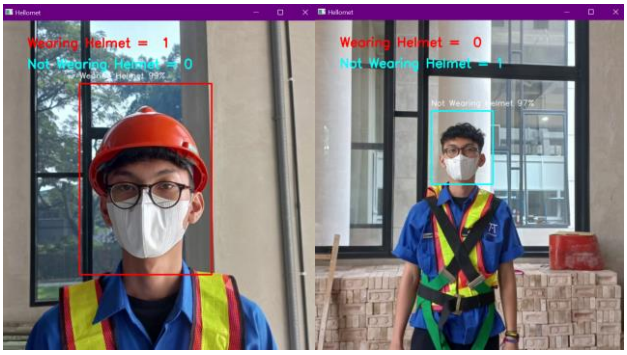

Fig. 8. Performance testing in an indoor practice area.

B. *Model Evaluation*

The performance of the CNN model for the classification of helmet use must be evaluated to see how well it predicts the test data. The study uses the 10-fold cross-validation method. The first scenario is tested to get the optimal number of epochs for the model in the training stage to obtain the best performance. Epoch is one round when the entire dataset has passed the training stage in the Neural Network. This study tested several epoch values, including 10, 20, 30, 40, and 50. During the epoch test, the batch size was 32, and the Adam optimizer was used [11]. The results of testing the precision, accuracy, recall and F1 score values in each epoch follow.

TABLE III. RESULT VALUES IN EACH EPOCH

Epoch	Accuracy	Precision	Recall	F1 Score
10	0.7	0.75	0.7	0.71
20	0.77	0.78	0.77	0.78
30	0.77	0.8	0.77	0.77
40	0.79	0.8	0.79	0.8
50	0.8	0.83	0.8	0.8

Based on the test results shown in TABLE III, it can be seen that as the epoch value increases, the accuracy, precision, recall and F1 score also increase. The best performance in theCNN model is obtained when the epoch value is 50. The performance is indicated by the highest accuracy, precision, recall and F1 scores compared to other epochs.

IV. CONCLUSIONS

The use of safety helmets was successfully identified using the YOLOv3 tiny. The system is indicated to be successful in predicting based on each image of wearing a helmet or not. Furthermore, the number of epochs in system learning affects the accuracy obtained and how long it takes to analyze the data. Fifty epochs show the best accuracy. The more data the system learns, the better it can test the data and the longer it takes to process it. The selected algorithms, methods and tools significantly impact the accuracy obtained. With these results, the system can help supervisors to ensure students wear safety helmets in the practice area with optimal results. Authors and Affiliations.

ACKNOWLEDGMENT

The authors thank XL Axiata Future Leaders for the research collaboration with Politeknik Astra.

REFERENCES

[1] A. Hayat and F. Morgado-Dias, "Deep Learning-Based Automatic Safety Helmet Detection System for Construction Safety," *Applied Sciences*, vol. 12, no. 16, p. 8268, Aug. 2022, doi: 10.3390/app12168268.

[2] Z. Jin *et al.*, "DWCA-YOLOv5: An Improve Single Shot Detector for Safety Helmet Detection," *J Sens*, vol. 2021, pp. 1–12, Oct. 2021, doi: 10.1155/2021/4746516.

[3] B. Zhang, C.-F. Sun, S.-Q. Fang, Y.-H. Zhao, and S. Su, "Workshop Safety Helmet Wearing Detection Model Based on SCM-YOLO," *Sensors*, vol. 22, no. 17, p. 6702, Sep. 2022, doi: 10.3390/s22176702.

[4] X. He, R. Cheng, Z. Zheng, and Z. Wang, "Small Object Detection in Traffic Scenes Based on YOLO-MXANet," *Sensors*, vol. 21, no. 21, p. 7422, Nov. 2021, doi: 10.3390/s21217422.

[5] L. Deng, H. Li, H. Liu, and J. Gu, "A lightweight YOLOv3 algorithm used for safety helmet detection," *Sci Rep*, vol. 12, no. 1, p. 10981, Jun. 2022, doi: 10.1038/s41598-022-15272-w.

[6] L. Yang, G. Chen, and W. Ci, "Multiclass objects detection algorithm using DarkNet-53 and DenseNet for intelligent vehicles," *EURASIP J Adv Signal Process*, vol. 2023, no. 1, p. 85, Aug. 2023, doi: 10.1186/s13634-023-01045-8.

[7] J.-H. Lo, L.-K. Lin, and C.-C. Hung, "Real-Time Personal Protective Equipment Compliance Detection Based on Deep Learning Algorithm," *Sustainability*, vol. 15, no. 1, p. 391, Dec. 2022, doi: 10.3390/su15010391.

[8] I. Campero-Jurado, S. Márquez-Sánchez, J. Quintanar-Gómez, S. Rodríguez, and J. Corchado, "Smart Helmet 5.0 for Industrial Internet of Things Using Artificial Intelligence," *Sensors*, vol. 20, no. 21, p. 6241, Nov. 2020, doi: 10.3390/s20216241.

[9] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A Convolutional Neural Network for Modelling Sentences," Apr. 2014.

[10] Q. An, Y. Xu, J. Yu, M. Tang, T. Liu, and F. Xu, "Research on Safety Helmet Detection Algorithm Based on Improved YOLOv5s," *Sensors*, vol. 23, no. 13, p. 5824, Jun. 2023, doi: 10.3390/s23135824.

[11] M. S. Al Reshan *et al.*, "Detection of Pneumonia from Chest X-ray Images Utilizing MobileNet Model," *Healthcare*, vol. 11, no. 11, p. 1561, May 2023, doi: 10.3390/healthcare11111561.