ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

FACIAL EXPRESSION DETECTION SYSTEM FOR STUDENTS IN CLASSROOM LEARNING PROCESS USING YOLOV7

Alifya Nuraisyar Aglaia[1], Mukhlishah Afdhaliyah[2], Fhatiah Adiba[3*], Andi Baso Kaswar[4],Muhammad Fajar B[5], Dyah Darma Andayani[6], Muhammad Yahya[7]

[1,2,3,4,5,6] Department of Informatics and Computer Engineering, Makassar State University, Indonesia
[7] Department of Vocational Engineering Education, Makassar State University, Indonesia

Email: alifyaaglaia02@gmail.com[1] , lisaafdhaliyah@gmail.com[2] , adibafhatiah@unm.ac.id[3], a.baso.kaswar@unm.ac.id[4], fajarb@unm.ac.id[5] , dyahdarma@unm.ac.id[6] , m.yahya@unm.ac.id[7]

Abstract

The utilization of technology in education is not only about using hardware or software, but also how technology can facilitate effective learning experiences. However, in the learning process there is a problem for teachers to know the level of student attention in the classroom to the material presented, so that the teacher does not know accurately the concentration of students during the learning process until it has an impact on the teacher's learning methods that are not in accordance with the characteristics of students. The purpose of this research is to detect students' facial expressions in the classroom learning process using yolov7. The implementation of several architectural models on CNN consists of several proposed methods, namely data collection, data augmentation, data annotation, split dataset, training, and model evaluation. System testing is done by measuring accuracy and comparing with other methods, namely CNN, CNN MobileNet, CNN EfficientNet-B0 and YoloV7. The test results show the average accuracy of CNN 80%, CNN MobileNet 93%, CNN EfficientNet-B0 31% and YoloV7 96%. Based on these results, it can be concluded that the YoloV7 method can detect student concentration effectively and efficiently compared to CNN, CNN MobileNet, and CNN EfficientNet-B0.

Keywords: Facial Expressions, YOLOv7, CNN, CNN MobileNet, CNN EfficientNet-B0

Received: 08-07-2024 | **Revised:** 09-04-2024 | **Accepted:** 10-20-2024
DOI: https://doi.org/10.23887/janapati.v13i3.83978

INTRODUCTION

Currently, technology has developed very rapidly, one of which is monitoring and detection using AI (Artificial intelligence), has been widely used in various fields. In recognizing and analyzing student behavior in the classroom as an evaluation material to measure the effectiveness of learning.[1]. Object detection technology and artificial intelligence have shown rapid development in recent years. One technique that has recently received wide attention is YOLO (You Only Look Once). [2]. An object detection algorithm that enables real-time object recognition with a high degree of accuracy. The development of AI has entered a new era by using deep learning techniques. [3] This technology has been applied in various fields, starting from security surveillance to facial recognition in smartphone applications and websites. With the development of AI technology [4] and deep learning [5] provide new research methods in detecting student activities [6]. Meanwhile, in education, there is an increasing demand for technological innovations to enhance the learning experience. The utilization of technology in education is not only about the use of hardware or software, but also about how technology can facilitate a more effective learning experience.

Based on this, there are several previous studies related to the problem of recognizing students' activities in classroom videos and analyzing their knowledge and performance points during teaching sessions using the enhanced Asynchronous Interaction Aggregation (AIA) network method, which combines Multi-scale Temporal Attention (MSTa) and Multi-scale Channel Spatial Attention (MsCSa) modules and YOLOV7 with 84.8% accuracy.[3]. Furthermore, research on the improvement of the Yolo-v4 model and the incorporation of CSP DarkNet53 comparison results show that the face mask recognition mAP can reach 98.3% and a high frame rate of 54.57 FPS. [7]. Furthermore,

research using Yolo V8 resulted in an increase in detection accuracy with an average of 7.7% compared to the base model, The performance of the UAV-YoLo V8 model is superior to other mainstream models because it offers higher detection speed accuracy.[8].

In addition, previous studies using Yolo V3 had problems in terms of limited computing power and excessive power consumption by embedded and mobile devices using Mixed YOLO V3-Lite with accuracy results of 13.68%, 8.48%, and 12.67%, respectively. The proposed Mixed YOLO V3-Lite network achieves a good balance between detection accuracy and speed, making it suitable for devices that have limited resources. This network model also shows improved performance in terms of mAP (average precision) compared to other experiments, with a decrease in the number of computational layers. [9].

This research proposes a new framework that combines facial landmarks and YoloV5 architecture to improve drowsiness detection. Current methods for detecting driver drowsiness often lack accuracy in capturing subtle aspects such as facial expressions, eye movement patterns, micro head movements, changes in blink frequency, and variations in steering control behavior. This study produced accuracy results of 95.5% and 96.4% on the UTA benchmark and custom datasets, respectively. This represents a significant improvement of 3.2% compared to existing techniques. The integration of facial landmarks and the Yolo V5 architecture allows the system to capture subtle indicators of drowsiness by observing small changes in facial expressions, providing valuable insights into the driver's level of alertness." [10]. [10].

Then the previous research was the development of a python application using Yolo V3 for social distance detection and digital image analysis and counting the number of people and detecting their proximity using the Yolo Tiny V3 algorithm with 76.316% accuracy for detecting human objects and 94.444% accuracy in detecting warning (unsafe) distances between people. [11]. However, in the current learning process, there is a problem for teachers to know the level of student attention in the classroom so that teachers do not know accurately the concentration of students during the learning process to have an impact on teaching methods that are not in accordance with the characteristics of students and the above research still has shortcomings related to turnover power, accuracy and speed of detection, behavior detection and accuracy in digital detection and analysis.

Therefore, in this research it is proposed to examine the student facial expression detection system in the classroom learning process using Yolov7 to improve and make comparisons with previous research. This research is proposed using CNN comparison with Yolov7 because of the increase in detection speed accuracy which is higher than other versions. There are 5 stages of the proposed research, namely data collection, data preprocessing and augmentation, data annotation, training, model evaluation. The proposed method is an innovative model that offers higher detection accuracy and speed.[12]. The proposed method is an innovative model that offers higher detection accuracy and speed, which can make a significant contribution to students' facial expressions that can enhance the interaction between lecturers and students and improve learning effectiveness.

LITERATURE STUDY

A. YOLOv7

You Only Look Once (YOLO) is an algorithm used for object detection. [13]. YOLO was first introduced in 2016 and then the improvements proposed in this study to add mechanisms to YOLOv7 are introduced in detail. YOLOv7 is an object detection system that has a fairly complex architectural structure.[14] and the fastest and most accurate detection for computer vision.[15]. One of the main advantages of YOLO is its execution speed.[16]. Figure 1 shows a diagram of the network structure of yolov7 [17][18]. The preprocessing method in the YoloV7 model integrates techniques from YoloV5, including the use of Mosaic data augmentation which is effective for detecting small objects. [19]. In terms of architecture, an extended ELAN (E-ELAN) based on ELAN is introduced. Cardinality expansion, randomization, and merging techniques are applied to continuously improve the learning capability of the network without disturbing its original gradient path.[20]. To expand the channels and cardinalities in the computing blocks, group convolution is used. Different groups of computational blocks are directed to learn more diverse features.

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

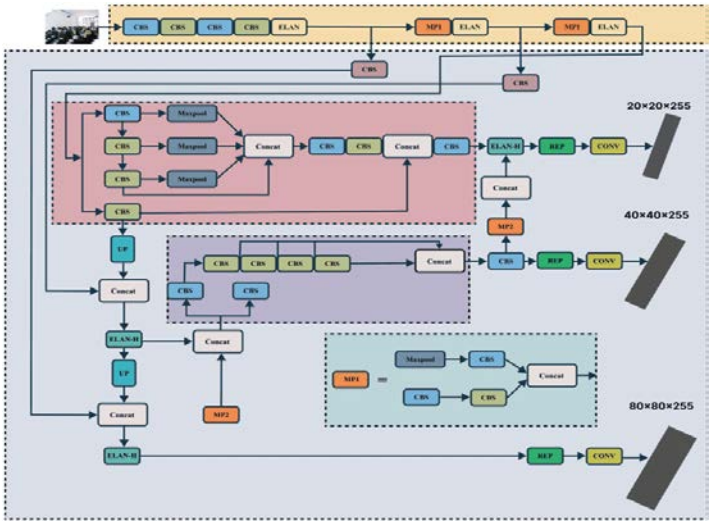


Figure 1. YOLOv7 Architecture

B. CNN

Convolutional Neural Network (CNN) is one of the development models of neural networks commonly used in object detection in images and is the most successful model in the field of image processing. [21], [22]. CNN is one component or part of a neural network that is used to process data in the form of images. [23] such as image processing tasks[24] image classification[25] object recognition, and image segmentation[26]. The input image is processed by the CNN layer and information is extracted from the received image to enable subsequent image recognition and classification. A CNN consists of three main layers namely convolution layer, union layer, and fully connected layer. **The** architecture of a CNN is shown in Figure 2**.**

The convolution layer is the first layer and is the core of the CNN architecture. This layer performs a convolution process of filters in the form of a matrix of about 3x3 in size to extract important features from the input image. In addition, stride length and padding also play an important role in this layer. Stride is a parameter that determines the amount of filter shift in the matrix. Padding or zero padding is a technique used to preserve the original size of the input image. The pooling layer performs matrix reduction by adopting some or a group of features generated in the convolutional layer. The adopted features are processed until they have a value that represents the value of the selected group or part. After passing through the pooling layer, there is the last layer of the CNN architecture, which is the fully connected layer. At this level, classification is performed based on the input received. Before classification, the features created in the previous layer must be reshaped or flattened into vectors so that they can be used as input in the fully connected layer. Finally, there are activation functions such as softmax and sigmoid to classify based on the highest category value.[27].

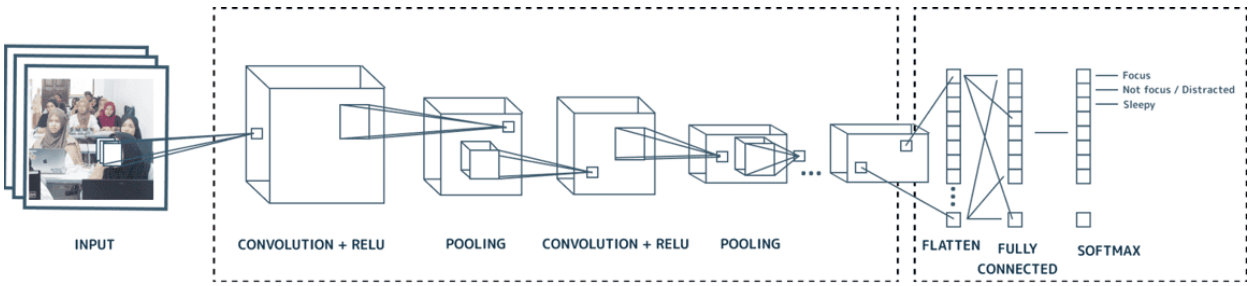


Figure 2. CNN Architecture

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

C. CNN MobileNet

MobileNet is an architecture that is able to reduce the size of the model by dividing the layer into two parts, namely standard convolution and depth convolution. [28]. MobileNet has a smaller weight size and faster computation time in the training process, so it can be easily implemented to meet the needs of mobile and embedded applications.[29]. Compared with traditional convolutional networks of the same depth, this method significantly reduces the number of parameters, resulting in a lighter artificial neural network. MobileNet is built using depth-separable convolution layers, consisting of a depth convolution layer and a point convolution layer. If these two types of convolutions are computed separately, MobileNet consists of 28 layers [30]. By adjusting the width multiplier hyperparameters, the number of parameters in a conventional MobileNet can be reduced to 4.2 million. The input image has a pixel size.

D. CNN EfficientNet-B0

EfficientNet-B0 is one of the models in the EfficientNet family specifically designed for computer vision applications. It uses intelligent scalability strategies to process visual data more efficiently. Developed by Google's research team, EfficientNet-B0 stands out for its outstanding combination of high adaptability and computational efficiency. The model works by identifying various visual patterns and uses deep learning techniques to recognize important features that distinguish objects. Despite being considered a basic variant in the EfficientNet family, EfficientNet-B0 is capable of accomplishing detection tasks with good accuracy. One of the main advantages of EfficientNet-B0 is its efficiency in use.[20]. The architecture of the EfficientNet-B0 CNN is shown in Figure 4.

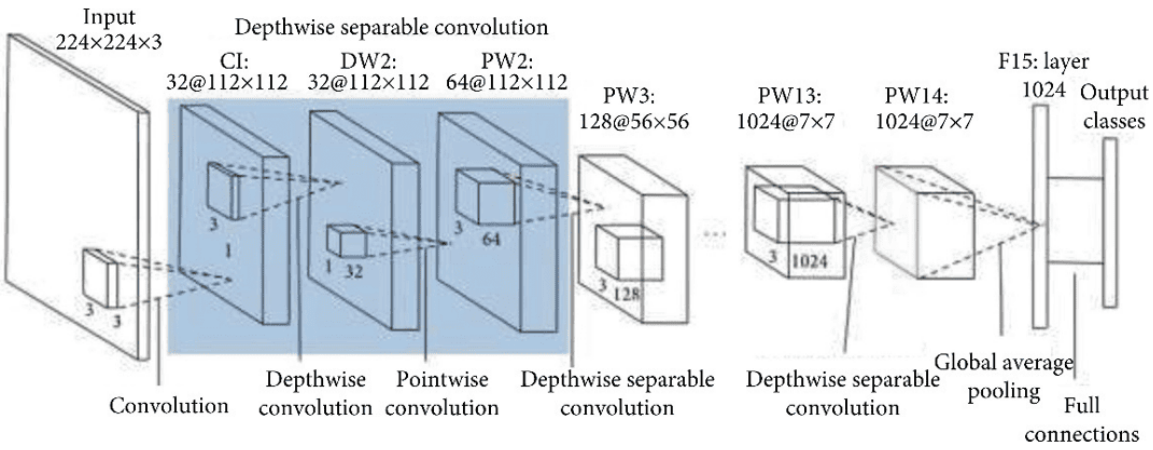


Figure 3. CNN Mobile Network Architecture

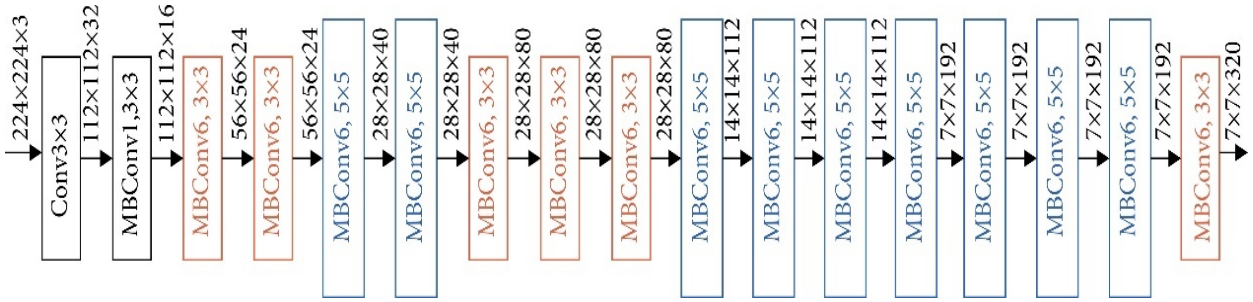


Figure 4. EfficientNet-B0 Architecture

.

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

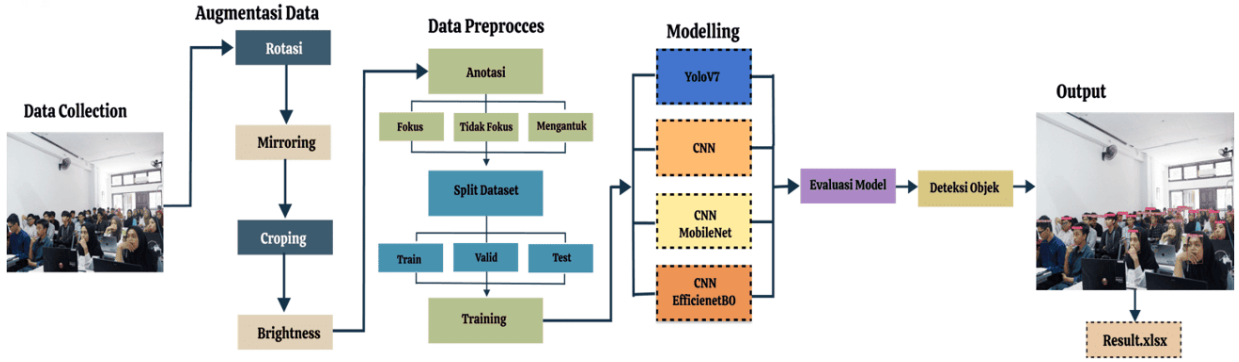


Figure 5. Proposed Research Flow

METHODS

In this study, the proposed method involves a series of steps, namely the process of data collection, data addition, data annotation, such as rotation, mirroring, and brightness change. This is followed by the annotation stage, split dataset, and then training. After the model successfully detects objects, the detection results are saved into an excel file

Data Collection

In this study, data was collected from two main sources, namely direct data collection and data available on the website Drowsiness_ dataset (kaggle.com) . The total images used amounted to 3,000 images, consisting of three categories: 1,000 images for sleepy conditions, 1,000 images for focused conditions, and 1,000 images for unfocused conditions. The data were taken directly using a Fujifilm X-A5 camera on the campus of the Department of Informatics and Computer Engineering. Taking pictures using a camera in a position in front of students with room conditions that are carrying out the learning process. The dataset collection time was carried out on March 19 - April 15, 2024. While the data taken from Kaggle amounted to 460 images, and the rest were obtained by adding data to achieve balance. Tssshe data was divided into three parts: 10% is used for testing, 80% for model training, and 10% for validation. The dataset retrieved from Kaggle can be quoted from the description of the data source available on the platform. Figure 6 shows an example of the dataset used in this study.

Augmentation

The augmentation process is very useful for increasing the amount of image data, so that the resulting model does not experience overfitting.[31]The data that the author has collected has many variations based on focus that is not focused on the object, and sleepy. Based on this, the data obtained will be carried out in the augmentation stage with the aim of increasing the diversity of training data so that the model can perform better learning and change the image so as to get more data from the original data. In the augmentation stage, the author performs cropping, rotation, mirroring, and changes the brightness and mirroring of the image. Figure 7 shows the result of dataset augmentation.

\

(a) (b) (c)

Figure 6. Data set: (a) Drowsy; (b) Focused; (c) Unfocused

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

(a) (b) (c) (d) (e)

Figure 7. Augmentation: (a) original image; (b) image rotation; (c) brightness; (d) cropping; (e) mirroring.

(a) (b) (c)

Figure 8. Dataset Annotations: (a) Focused; (b) Out of focus; (c) Sleepy

Annotation

In this study, after the data has been collected and augmented, the next stage is data annotation. This process involves classifying the objects in the image into three classes with predefined indices, namely index 0 for "sleepy", index 1 for "focused", and index 2 for "unfocused". Data annotation was done manually by the author, who observed each image and determined the appropriate label based on the state of the object seen in the image. For example, it is categorized as sleepy if the hand covers the mouth like yawning, it is categorized as focused if the eye gaze is facing forward, then it is categorized as unfocused if the eye gaze is downcast and the eyes are closed. This annotation process is important to prepare the dataset needed to train and test the classification model, thus ensuring the model can correctly identify the condition of the subject in the image. Figure 8 shows an example of dataset annotation.

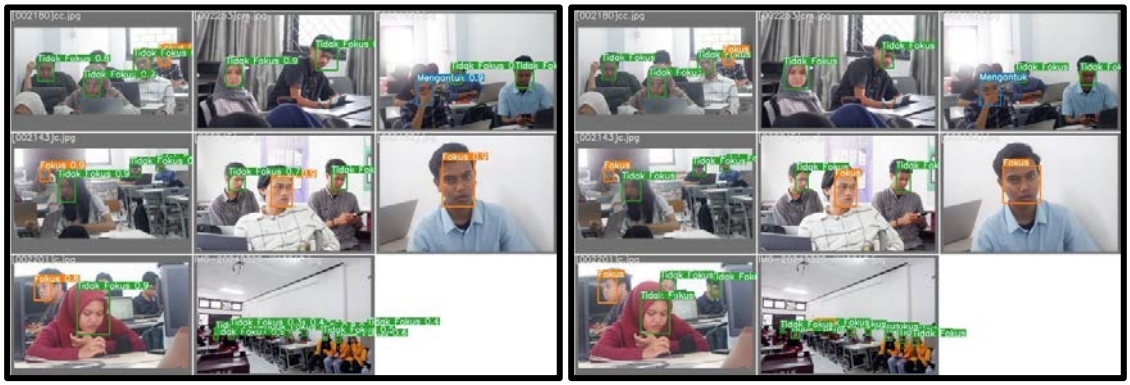
Model Making

In the model building stage, the author builds a model that is able to classify images based on the condition of the object into 3 classes involving model architecture selection, parameter configuration, and model training using datasets that have been collected, augmented, and annotated. In this case the author creates a YoloV7 model and CNN method with Mobilenet and EfficientNet-B0 architectures, the model that has been created will be tested and evaluated to ensure its performance in accordance with the research objectives. The main goal of this stage is to produce a model that can classify images with high accuracy according to predetermined categories.

Model Training

Model training in this study uses 80% training data 10% test and 10% for validation. The model testing process uses a built-in Nvidia GForce RTX 3050 6 GB GPU, Cuda 12.1 is installed to be used to maximize performance and make model training faster than using a CPU. The model training process can be seen in Figure 9.

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024



(a) (b)

Figure 9. Testing: (a) label data; (b) object detection result

Table 1. Method Comparison Results

Methods	Accuracy (%)	Precission (%)	Recall (%)	Epoch
YOLOv7	96	99	99	100
CNN	80	89	89	100
CNN MobileNet	93	90	98	100
CNN Efficient-NetB0	31	10	33	100

Results and Discussion

Accuracy is the most important thing in object classification [32]. This research compares several methods as a comparison, namely Yolov7, CNN, CNN MobileNet, CNN Efficient-NetB0. The author starts this research by dividing the data into test data, training data, and validation data. The dataset consists of three classes namely sleepy, focused, and unfocused. Training data covers 80% of the total dataset, test data and training data each covers 10% of the dataset.

Based on Table 1, the results of the method comparison between YOLOv7, CNN, CNN MobileNet, and CNN Efficient-NetB0. YOLOv7 shows excellent performance with a high accuracy of 96% and almost perfect precision and recall values of 99%. This shows that this model is very effective in detecting and classifying objects, with very minimal errors.

CNN performs quite well with 80% accuracy. Although the precission and recall values are higher at 89%, indicating that this model is better at ensuring the detected objects are correct (precission) and in detecting almost all objects (recall). However, compared to YOLOv7, CNN still loses in terms of overall accuracy.

MobileNet CNN shows quite good results with an accuracy of 93%, which is higher than the basic CNN. The accuracy is slightly lower at 90% compared to the very high recall of 98%, which shows that the model is very good at detecting all the objects present although some detections may be incorrect.

The Efficient-Net B0 CNN showed very poor performance with an accuracy of only 31% and a very low precision of 10%. Recall was also low at 33%, indicating that the model failed to detect and classify objects correctly. Although EfficientNet is known to perform well in image classification tasks in some scenarios, these results suggest that the configuration or training of this model may not be optimal or there are issues in the training process.

From this data, it can be concluded that YOLOv7 is the best model in terms of object detection and classification with very high accuracy, precission, and recall. MobileNet CNN also shows good results and can be an efficient alternative. Basic CNN performs quite well but is still below YOLOv7 and MobileNet. Efficient-Net B0 CNN, despite being a sophisticated model in theory, performed very poorly in this experiment, which may be due to training or implementation issues.

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

Table 2. Speed Comparison of Methods

Image Name (.jpg)	Model Detection Results			
	CNN	**EfficientNet-B0**	**MobileNet**	**YoloV7**
Mng1	Drowsiness	Drowsiness	Out of Focus	Drowsiness
Mng2	Focus	Drowsiness	Out of Focus	Drowsiness
Fks1	Drowsiness	Drowsiness	Drowsiness	Focus
Fks2	Focus	Drowsiness	Drowsiness	Focus
Tdf1	Out of Focus	Drowsiness	Out of Focus	Out of Focus
Tdf2	Drowsiness	Drowsiness	Out of Focus	Out of Focus
Total Detection Time	**2.570 seconds**	**5.030 seconds**	**15.936 seconds**	**2.375 seconds**

Based on the detection results shown in Table 2, the speed of each model in detecting images shows significant variation. The detection results of the YoloV7 model are said to be perfect because the model is able to detect objects correctly, YoloV7 as the fastest model with a total detection time of 2.375 seconds to detect images. This YoloV7 model is able to process data very efficiently and provide fast results and makes it ideal for real-time applications that require fast and accurate detection. CNN is also quite fast with a detection time of 2.570 seconds, only slightly slower than YoloV7. The CNN detection results showed 3 correct predictions and 3 incorrect predictions. CNNs are generally efficient in handling image processing tasks, as convolutional networks are able to extract features from images quickly. However, standard CNNs may not be as optimal as YoloV7 in terms of real-time detection due to Yolo's more speed-focused architecture. Meanwhile, EfficientNet-B0 shows 2 correct detections in the sleepy class and 4 incorrect detections. This is because the model is less capable in detecting objects and has a slower detection time of 5.030 seconds, although the model is optimized for parameter efficiency and higher accuracy, its more complex architecture causes longer inference time. Then MobileNet shows similar results to the EfficientNet-B0 model. The detection results of the MobileNet model showed 2 correct object predictions and 4 incorrect object predictions. This model, designed for devices with low computing power, recorded the longest detection time of 15.936 seconds, making it less suitable for applications that require high speed. From this comparison, the YoloV7 Model was the most superior in speed, followed by CNN, then EfficientNet-B0, and finally MobileNet.

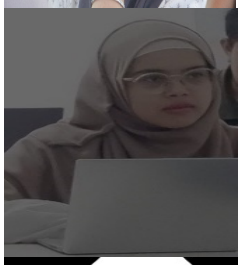
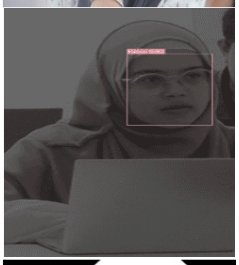
Table 3 displays the results of the YOLOv7 model evaluation, based on the results of the model evaluation on YOLOv7 to detect student facial expressions in table 1, the average accuracy result is 96%, where this model manages to get the best accuracy results in the classes Sleepy 100%, Unfocused 97%, and Drowsy 97%. Detection using the YOLOv7 model detects objects using 6 stages with different conditions namely drowsiness, focus, out of focus, brightness level, and image rotation. The detection results are shown in table 3.

Table 3. Evaluation of the YOLOv7 Model

Epoch	Class	Accuracy (%)	Precision (%)	Remember (%)	mAP (%)	Total Data
100	Drowsiness	100	100	99	99	1000
100	Focus	91	99	95	98	1000
100	Out of Focus	97	100	93	98	1000

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

Table 4. YOLOv7 Detection Results

Original Image	Image Detection	Detection Result	
		Belief Value	**Results**
First try			
		0.39	Sleepiness Detected
		0.90	Detected Focus
		0.74	Detected Out of Focus
Second try			
		0.81	Sleepiness Detected
		0.82	Detected Focus
		0.85	Detected Out of Focus

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

Based on the results from table 4, it can be obtained that there is success in testing the YOLOv7 model in detecting these objects. In the sleepy class, the first experiment obtained a Confidence Value of 0.39 due to the position of the object from the side and low image quality. Such as poor resolution, insufficient lighting that reduces the ability of the model to detect objects and produce a lower Confidence Value then in the second experiment the confidence value has increased from 0.39 to 0.81 due to image quality as well as poses and facial expressions in the second trial image object expressions are clearer, and better image quality so that the model's ability to detect gets an increase in Confidence Value. The focus class in the second experiment decreased from 0.90 to 0.82 due to the quality of image lighting and image clarity. The unfocused class in the second experiment experienced an increase in confidence value from 0.74 to 0.85 the same factor in the sleepy class.

Figure 10 shows the training graph of the YOLOv7 model through the training process up to 100 epochs. From epoch 0 to 100, the graph in Figure 10 shows a significant performance improvement of the object detection model. At the beginning of the training, from epoch 0 to about 10, there is a decrease in Box loss, Objectness loss, and Classification loss, which indicates that the model quickly learns to improve the prediction of the bounding box, distinguish between the object and the background, and classify the object correctly. After that, the loss decrease becomes slower but still shows a downward trend until the 100th epoch, indicating continuous improvement in model performance.

On the validation data, the same pattern is also seen. Validation Box loss, Objectness loss, and Classification loss all decrease significantly at the beginning of training and then decrease gradually but consistently, indicating that the model not only learns well on training data but can also generalize on validation data.

Precision and Recall also increased rapidly at the start of training, especially
up to about epoch 30, then increases more slowly but steadily up to epoch 100. This shows that the model is getting better at identifying the correct object without generating many false positives as well as detecting all instances of the object in the image.

In addition, the metrics mAP@0.5 and mAP@0.5:0.95 show significant improvement from epoch 0 to 30, followed by a more gradual but consistent improvement up to epoch 100. This indicates that the model maintains good performance in detecting and classifying objects at various IoU thresholds.

Overall, the graphs show that the developed object detection model experienced significant performance improvements during the training and validation processes. The steady loss reduction and improvement in precision, recall, and mAP indicate that the model successfully learns from the training data and is able to generalize well to the validation data, which indicates an effective approach and model architecture for detecting students' facial expressions in the classroom learning process.

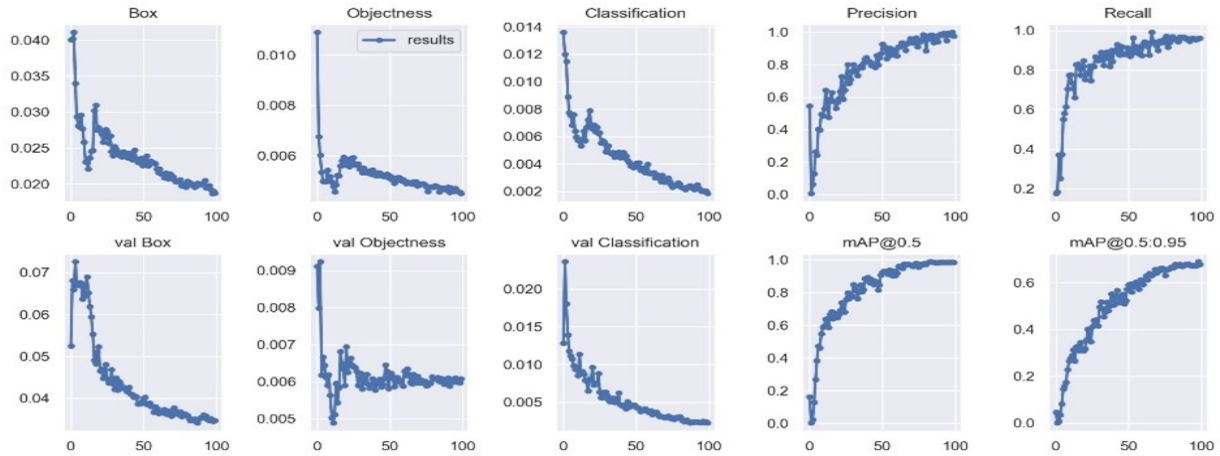


Figure 10. YOLOv7 Model Training Graph

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

Table 5. CNN Model Evaluation

Number of Epochs	Time	Accuracy (%)	Loss (%)	Val_Accuracy (%)	Val_Loss (%)
Epoch 10/100	22 seconds 516ms/step	46	32	65	93
Epoch 30/100	12 seconds 358ms/step	100	0,82	80	86
Epoch 50/100	11 seconds 345ms/step	100	0,015	79	10
Epoch 99/100	11 seconds 337ms/step	100	100	80	15

Table 5 shows the evaluation results of the CNN mode. Based on the training results of the model in this study, we can see the development of the model's performance over 100 epochs. At Epoch 10, the model achieved a training accuracy of 46% and a validation accuracy of 65%, with loss values of 32% and 93%, respectively. This shows that in the early stages, the model is still learning from the training data.

At Epoch 30, there was a significant improvement where the training accuracy reached 100% and the training loss value dropped dramatically to
0.82%. The validation accuracy also increased to 80%, although the validation loss value was still quite high at 86%. Furthermore, at Epoch 50, the model still maintains 100% training accuracy with a decrease in training loss value to 0.015%. However, the validation accuracy slightly decreased to 79% with a significant decrease in validation loss value to 10%. At Epoch 100, the training accuracy is still 100%, and the validation accuracy slightly increases to 80%, but the validation loss value increases to 15%.

Overall, the model showed excellent learning ability on the training data, indicated by the training accuracy reaching 100% after the 30th epoch. However, fluctuations in accuracy and validation loss values indicate a symptom of overfitting, where the model recognizes the training data well but is less able to generalize to new data. Figure 11 shows the training graph of the CNN model, Based on the attached accuracy and model loss graphs, we can see how the model performance evolves during the training process. On the accuracy graph, we can see that the training accuracy increases rapidly and reaches 100% within the first few epochs. Meanwhile, the testing accuracy increases to about 80% within the first 20 epochs and then tends to stabilize with small fluctuations around that value until the end of training.

In the loss graph, it can be seen that the training loss value decreases drastically to near zero within the first few epochs. However, the testing loss value decreases more slowly and stabilizes around a higher value compared to the training loss. At the end of the training, the testing loss shows a slight increase.

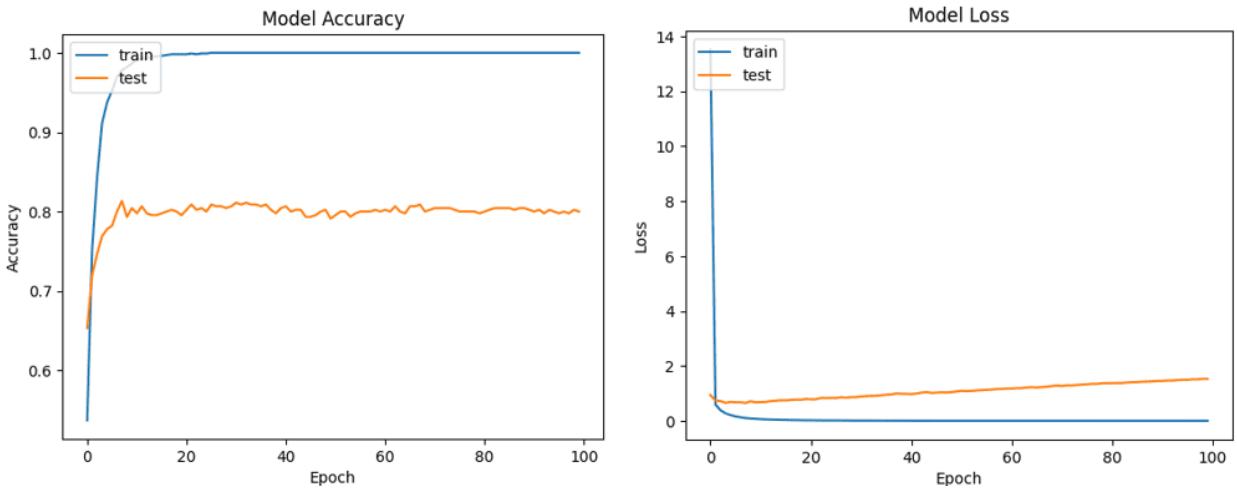


Figure 11. CNN Training Graph

This analysis shows that the model is overfitting. Overfitting occurs when the model is too good at recognizing patterns in the training data but less able to generalize the pattern to the testing data or new data. This can be seen from the significant difference between accuracy and loss in training and testing data.

The training results of the MobileNet CNN model in Table 6 show the development of the model performance at several different epoch counts, which shows the model evaluation results with average accuracy (total average accuracy). At the first epoch, the accuracy was recorded at 88.15% with a loss value of 0.3940. At this point, the model has achieved a fairly good level of accuracy, and the resulting loss value is also quite low, which indicates that the model has learned significant patterns in the data. Furthermore, at the 30th epoch, the accuracy slightly increased to 88.26% with a loss value of 0.3997. Despite the small increase in accuracy, the loss value of the model increased slightly, but was still within the acceptable range. At the 50th epoch, the accuracy again increased to 88.46% with a loss value that slightly decreased to 0.3969. This shows that the model continues to improve its performance over time. Finally, at the 100th epoch, the accuracy peaked at 89.09% with the loss value decreasing to 0.4063. Although there is a slight decrease in accuracy on the validation data, the low loss value shows that the model is good enough at predicting the data. It should be noted that the time required for each epoch also increases as the number of epochs increases, reaching 68 seconds at the 100th epoch.

Figure 12 shows the model performance metrics over 100 epochs for training and validation, including accuracy, loss, AUC, precision, and F1 score. The training and validation accuracies increase sharply at first and then stabilize around 0.88 to 0.90. Loss decreased dramatically in the first 20 epochs and then stabilized at a very low value, indicating good convergence. The AUC, which measures the overall performance of the classification model, increased rapidly and stabilized around 0.95, indicating the model's excellent ability to discriminate between classes. Precision shows a similar pattern of improvement, stabilizing around 0.85 to 0.88, meaning the model has a consistent level of precision. F1-score, which combines precision and recall, also stabilized in the range of 0.80 to 0.82 after the initial increase. Overall, this graph shows that the model achieves optimal performance within the first 20 to 30 epochs and then maintains stable performance.

Table 6. Evaluation of MobileNet CNN Architecture Model

Number of Epochs	Time	Accuracy (%)	Loss (%)	Val_Accuracy (%)	Val_Loss (%)
Epoch 10/100	58 seconds 1 second/step	88	39	87	42
Epoch 30/100	56 seconds 1 second/step	88	39	87	42
Epoch 50/100	49 seconds 1 second/step	88	39	88	40
Epoch 99/100	68 seconds 2 seconds/step	89	40	88	39

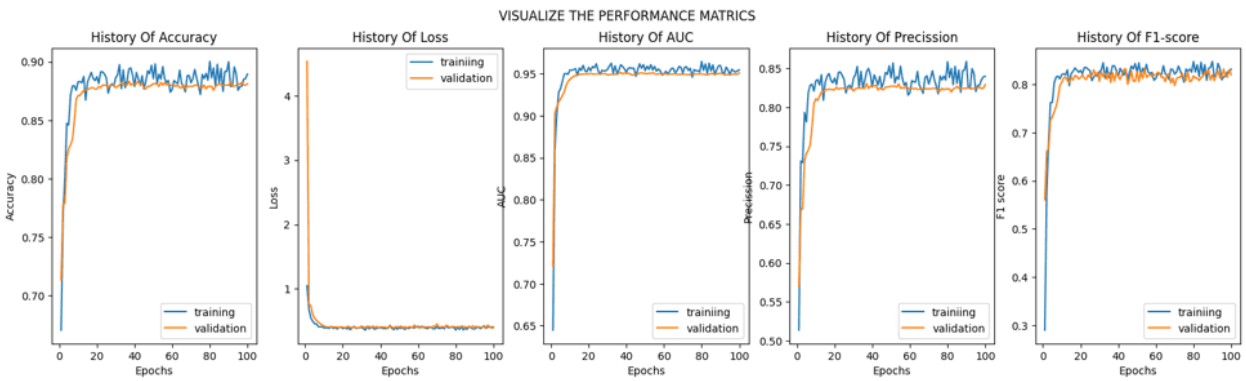


Figure 12. CNN Mobile Network Architecture Graph

Table 7. Evaluation of EfficientNet-B0 CNN Architecture Model

Number of Epochs	Time	Accuracy (%)	Loss (%)	Val_Accuracy (%)	Val_Loss (%)
Epoch 10/100	62 seconds 740ms/step	40	-0.02	54	-2.57
Epoch 30/100	11s 452ms/step	50	-6.78	51	-13.78
Epoch 50/100	22s 552ms/step	49	-22.37	54	-34.43
Epoch 99/100	11s 441ms/step	52	-90.98	52	-133.75

Table 7 shows the test results of the EfficientNet B0 CNN model which shows the model evaluation results with average accuracy (total average accuracy). In the first epoch, the accuracy was recorded at 40.63% with an unusual loss value of -0.0221. This indicates an error in measurement or calculation. However, in the third epoch there was a significant increase in accuracy to 50.19%, but the loss value showed a very high negative number of -6.7887. There may be errors in measurement or calculation that cause the loss value to be unreasonable. At the 5th epoch, although the accuracy decreased slightly to 49.17%, the very low loss value (-22.3739) still decreased significantly. This may indicate a deficiency in model training. Finally, at the 10th epoch, the accuracy slightly increased to 52.36%, but the loss value again showed a very high negative number of -90.9895. This indicates that there is a serious problem in training this model, possibly related to the configuration or pre-processing of the data. Thus, the test results of the EfficientNet B0 CNN model in this table indicate problems that need to be further reviewed to ensure the reliability and accuracy of the model.

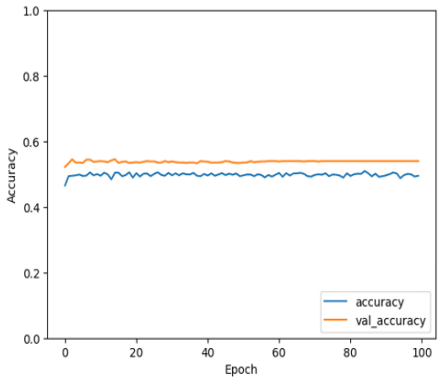


Figure 13. Graph of EfficientNet-B0 CNN Architecture Model

Figure 13 shows the accuracy and validation accuracy (val_accuracy) graphs for 100 epochs of model training. It can be seen that the training accuracy tends to stabilize around a relatively low value, while the validation accuracy is also stable but slightly higher than the training accuracy. This indicates that the model has difficulty improving its performance, possibly due to underfitting, where the model cannot capture the patterns in the training data well. This stable but low performance indicates that the model still needs to be improved.

Conclusion

In this research, a system is made that is able to detect student facial expressions. This system can detect objects, namely student faces that are sleepy, focused and unfocused. The training and comparison process uses 4 methods namely YOLOv7, CNN, CNN MobileNet, CNN EfficientNet-B0 to find the best model and provide the best accuracy, precission, and recall values. From the results of the study, YOLOv7 showed excellent performance with high accuracy of 96% and almost perfect precission and recall values of 99% where this model managed to get the best accuracy results in Sleepy 100%, Out of Focus 97%, and Focus 91% classes. This study concludes that the YOLOv7 method is the best method to use in implementation, including in the field of education and security surveillance, because this method provides superior performance in terms of accuracy and speed of detection. Future research suggestions could include the integration of existing CCTV in classrooms to study interactions between students, teaching and learning effectiveness.

Referensi

[1] Z. Shou, M. Yan, H. Wen, J. Liu, J. Mo, dan H. Zhang, "Penelitian tentang Metode Pengenalan Perilaku Tindakan Siswa Berdasarkan Gambar Deret Waktu di Kelas," *Ilmu Terapan (Swiss)*, vol. 13, no. 18, Sep. 2023, doi: 10.3390/app131810426.

[2] D. Sadykova, D. Pernebayeva, M. Bagheri, dan A. James, "IN-YOLO: Deteksi Waktu Nyata Isolator Tegangan Tinggi di Luar Ruangan Menggunakan Pencitraan UAV," *IEEE Transactions on Power Delivery*, vol. 35, no. 3, pp. 1599-1601, 2020, doi: 10.1109/TPWRD.2019.2944741.

[3] Z. Wang, L. Li, C. Zeng, dan J. Yao, "Pengenalan Perilaku Belajar Siswa yang Menggabungkan Augmentasi Data dengan Representasi Fitur Pembelajaran di Ruang Kelas Pintar," *Sensor*, vol. 23, no. 19, 2023, doi: 10.3390 / s23198190.

[4] R. Vaishya, M. Javaid, I. Haleem, dan A. Haleem, "Aplikasi AI untuk pandemi Covid-19," no. Januari, 2020.

[5] L. Liu *dkk.*, "Pembelajaran Mendalam untuk Deteksi Objek Generik: A Survey," *Int J Comput Vis*, vol. 128, no. 2, hal. 261-318, 2020, doi: 10.1007/s11263-019-01247-4.

[6] X. Ning, "Pengenalan Perilaku Mahasiswa Berdasarkan Peningkatan Algoritma Deep Learning," *Jurnal Internasional Teknologi Pembelajaran dan Pengajaran Berbasis Web*, vol. 18, no. 2, pp. 1-16, 2023, doi: 10.4018/ijwltt.320647.

[7] Z. Ding, J. Guo, J. Liu, dan H. Zhu, "Algoritma pendeteksi pemakaian masker berdasarkan YOLOv7 yang ditingkatkan," *ACM International Conference Proceeding Series*, pp. 165-173, 2023, doi: 10.1145/3614008.3614032.

[8] G. Wang, Y. Chen, P. An, H. Hong, J. Hu, dan T. Huang, "UAV-YOLOv8: Model Pendeteksian Objek Kecil Berdasarkan YOLOv8 yang Disempurnakan untuk Skenario Fotografi Udara UAV," *Sensor*, vol. 23, no. 16, 2023, doi: 10.3390/s23167190.

[9] H. Zhao *dkk.*, "Mixed YOLOv3-LITE: Metode deteksi objek waktu nyata yang ringan," *Sensors (Swiss)*, vol. 20, no. 7, 2020, doi: 10.3390/s20071861.

[10] M. Arava dan D. Meena Sundaram, "Jurnal Internasional SISTEM CERDAS DAN APLIKASI DALAM REKAYASA Meningkatkan Deteksi Kantuk Pengemudi: Perpaduan Landmark Wajah dan Arsitektur YOLOv5 yang Dimodifikasi," *Makalah Penelitian Asli Jurnal Internasional Sistem Cerdas dan Aplikasi dalam Rekayasa IJISAE*, vol. 2024, no. 11s, pp. 437-449, 2024.

[11] Y. Yunefri, Sutejo, Y. E. Fadrial, K. Anggraini, M. Ramadhani, dan R. Hardianto, "Implementasi Pendeteksian Objek dengan Algoritma You Only Look Once pada Waktu Tatap Muka yang Terbatas di Masa Pandemi," *Jurnal Rekayasa Terapan dan Ilmu Pengetahuan Teknologi*, vol. 4, no. 1, hal. 400-404, 2022, doi: 10.37385/jaets.v4i1.1161.

[12] I. A. Putra, "Analisis performa arsitektur model you only look once (yolo) versi 7 dalam melakukan segmentasi jenis virus dari citra mikroskop skripsi," 2023.

[13]L. Susanti, N. K. Daulay, and B. Intan, "Sistem Absensi Mahasiswa Berbasis Pengenalan Wajah Menggunakan Algoritma YOLOv5," *JURIKOM (Jurnal Riset Komputer)*, vol. 10, no. 2, p. 640, Apr. 2023, doi: 10.30865/jurikom.v10i2.6032.

[14] Y. Liu *dkk.*, "Metode Deteksi Batang untuk Robot Pemanen Buah Camellia oleifera Berdasarkan YOLOv7 yang Disempurnakan," *Forests*, vol. 14, no. 7, Jul. 2023, doi: 10.3390/f14071453.

[15] C.-Y. Wang, H.-Y. M. Liao, and I.-H. Yeh, "Merancang Strategi Perancangan Jaringan Melalui Analisis Jalur Gradien," Nov. 2022, [Online]. Available: http://arxiv.org/abs/2211.04800

[16] P. Jiang, D. Ergu, F. Liu, Y. Cai, dan B. Ma, "Tinjauan Perkembangan Algoritma Yolo," dalam *Procedia Computer Science*, Elsevier B.V., 2021, hlm. 1066-1073. doi: 10.1016 / j.procs.2022.01.135.

[17] "Ditarik kembali: Deteksi COVID-19 Berdasarkan Pemindaian CT Scan Paru Menggunakan Teknik Deep Learning," *Comput Math Methods Med*, vol. 2023, hlm. 1-1, Oktober 2023, doi: 10.1155/2023/9840132.

[18] K. Jiang *dkk.*, "Algoritma Deteksi Objek YOLOv7 yang Ditingkatkan dengan Mekanisme Perhatian untuk Estimasi Jumlah Bebek Rami," *Agriculture (Swiss)*, vol. 12, no. 10, Oct. 2022, doi: 10.3390/agriculture12101659.

[19] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies menetapkan state-of-the-art baru untuk pendeteksi objek waktu nyata," Juli 2022, [Online]. Tersedia: http://arxiv.org/abs/2207.02696

[20] D. Hindarto, "Analisis Akurasi Model: Membandingkan Deteksi Gulma pada Tanaman Kedelai dengan EfficientNet-B0, B1, dan B2," *Jurnal Teknologi Informasi dan Komunikasi)*, vol. 7, no. 4, p. 2023, 2023, doi: 10.35870/jti.

[21] O. A. Barro, M. Himdi, dan O. Lafond, "Radiasi Antena Patch yang Dapat Dikonfigurasi Ulang Menggunakan Efek Perisai Faraday Plasma," *IEEE*

Antennas Wirel Propag Lett, vol. 15, hal. 726-729, 2016, doi: 10.1109/LAWP.2015.2470525.

[22] D. Sebagai *dkk.*, "TUGAS AKHIR."

[23] N. Adhayanti, T. Nugroho, and R. Susiloatmadja, "SISTEM PENDETEKSIAN WAJAH BERMASUK SECARA REAL TIME MENGGUNAKAN METODE CNN," *JUIT*, vol. 2, no. 1.

[24] F. Hafifah, S. Rahman, and S. Asih, "Klasifikasi Jenis Kendaraan Pada Jalan Raya Menggunakan Metode Convolutional Neural Networks (CNN)," vol. 2, no. 5, pp. 292-301, 2021, [Online]. Available: https://ejurnal.seminar-id.com/index.php/tin

[25] D. Hananta Firdaus, B. Imran, L. Darmawan Bakti, and E. Suryadi, "KLASIFIKASI PENYAKIT KATARAK PADA MATA MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) BERBASIS WEB," 2022.

[26] D. Jha, MA Riegler, D. Johansen, P. Halvorsen, dan HD Johansen, "DoubleU-Net: Jaringan saraf konvolusi dalam untuk segmentasi citra medis," dalam *Prosiding - Simposium IEEE tentang Sistem Medis Berbasis Komputer*, Institut Insinyur Listrik dan Elektronik Inc, Jul. 2020, hlm. 558-564. doi: 10.1109 / CBMS49503.2020.00111.

[27] F. Denta Sukma and R. Mukhaiyar, "Alat Pendeteksi Ekspresi Wajah pada Pengendara Berbasis Image Processing," *JTEIN: Jurnal Teknik Elektro Indonesia*, vol. 3, no. 2, pp. 364-373, 2022.

[28] I. B. Venkateswarlu, J. Kakarla, dan S. Prakash, "Deteksi masker wajah menggunakan MobileNet dan blok penyatuan global," dalam *Konferensi IEEE ke-4 tentang Teknologi Informasi dan Komunikasi, CICT 2020*, Institute of Electrical and Electronics Engineers Inc.

[29] W. Wang, Y. Hu, T. Zou, H. Liu, J. Wang, dan X. Wang, "Pendekatan Klasifikasi Gambar Baru melalui Model MobileNet yang Ditingkatkan dengan Perluasan Bidang Reseptif Lokal di Lapisan Dangkal," *Comput Intell Neurosci*, vol. 2020, 2020, doi: 10.1155 / 2020 / 8817849.

[30] A. Fuadi *dkk.*, "PERBANDINGAN ARSITEKTUR MOBILENET DAN NASNETMOBILE UNTUK KLASIFIKASI PENYAKIT PADA CITRA DAUN KENTANG."

[31] K. Kusrini *dkk.*, "Penambahan data untuk klasifikasi hama otomatis di perkebunan mangga," *Comput Electron Agric*, vol. 179, Dec. 2020, doi: 10.1016/j.compag.2020.105842.

[32] S. Rahman and H. Dafitri, "Pengembangan Convolutional Neural Network untuk Klasifikasi Ketersediaan Ruang Parkir," Online.