J-COSINE (Journal of Computer Science and Informatics Engineering)
Vol. 10, No. 1, June 2026 E-ISSN:2541-0806
Accredited Sinta-4 by RISTEKDIKTI Decree No. 79/E/KPT/2023 P-ISSN:2540-8895

Fish Image Classification Based on Feature Extraction Using Convolutional Autoencoder and CNN Classifier

Destia Suhada*, Qalbi Ala Dinika, Muhamad Firdaus, I Gede Pasek Suta Wijaya
Master of Information Technology, Mataram University
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

Email: destiasuhada96@gmail.com, qalbialadinika@gmail.com, nakamurasan95@gmail.com, gpsutawijaya@unram.ac.id

Corresponding Author

Abstract **Automatic fish species classification is crucial for supporting fisheries management, particularly in coastal regions such as Lombok. This study proposes an artificial intelligence-based approach combining a Convolutional Autoencoder (CAE) as a feature extractor and a Convolutional Neural Network (CNN) as a classifier. The dataset consisted of 9,000 colored images from nine fish species, which were converted to grayscale. Experimental results show that the CAE+CNN model achieved a classification accuracy of 95.94% within 0.87 seconds for 1,800 images. In comparison, the pure CNN model achieved a higher accuracy of 99.05% and faster performance, but with a much larger model size and complexity. The CAE+CNN approach is considered more lightweight and efficient, making it suitable for deployment on resource-constrained devices. This technology has strong potential for local implementation to support the digital transformation of the fisheries sector and assist fishing communities in automatically identifying fish species, improving efficiency, and reducing dependency on manual classification by experts.**

***Key words*: Deep Learning, Feature Extraction, Convolutional Autoencoder, CNN, Fish Classification.**

I. Introduction

Indonesia is the largest archipelagic country in the world, with a coastline of over 81,000 km, giving it vast marine potential [1]. This potential is crucial for coastal communities, especially those relying on the fisheries sector, such as in Lombok, West Nusa Tenggara, where most of the population works as fishermen [2], [3]. However, the utilization of fishery resources still faces challenges, including the limited use of technology for fish species identification and classification [4].

Accurate fish classification is essential for fisheries management, ecological monitoring, and seafood marketing [4]. Currently, classification is still performed manually by experts or experienced fishermen, which is time-consuming and prone to errors, especially for visually similar species [5]. With advances in computer technology, deep learning-based computer vision offers a promising solution to automate fish image classification [6].

This study develops and evaluates a deep learning-based classification model using a publicly available fish image dataset. The model combines a Convolutional Autoencoder (CAE) as a feature extractor [7] and a Convolutional Neural Network (CNN) as a classifier [8]. CAE extracts key feature representations in a compact and informative form [9], preserving local spatial information while reducing input dimensions and noise [10]. These features are then processed by the CNN for accurate classification through convolutional filters, mimicking the human visual cortex [8].

A similar approach was applied by Boukhlifa and Chibani (2024) in classifying tomato plant diseases, showing that combining CAE and CNN can reduce the number of convolutional layers while maintaining performance [11].

The results of this study are expected to serve as a foundation for developing a broader and more applicable fish classification system, particularly in coastal areas such as Lombok. Although the dataset used is not from Indonesia, the approach can be adapted to local fish datasets to assist fishermen, fish markets, and marine researchers, supporting the national initiative for digital transformation in the marine and fisheries sector [12].

II. Theoretical Background & Related Work

A. *Theoretical Background*

A.1. Fish Classification in the fisheries sector

Fish classification plays an important role in Indonesia's fisheries sector, which has around 8,500 species, or 45% of the world's total fish species [4]. Accurate identification supports sustainability, cultivation, and trade [13] but manual methods still rely on experts and limited resources [4]. As a solution, digital image processing and algorithms such as CNNs enable automatic and efficient fish identification to support ecosystem management and fisheries development [14].

A.2. Use of CNN for image classification

CNN excels in supervised classification [14] by extracting local features through convolution and forming abstract representations in deeper layers [15]. Without the need for manual feature design, CNN is suitable for complex images such as ornamental fish [9] and can achieve up to 84% accuracy in koi fish classification [6].

A.3. CAE as a feature extractor

CAE is an unsupervised model combining convolutional and pooling layers in an encoder-decoder structure to extract and reconstruct image features [16], [17]. Unlike standard autoencoders, CAE preserves local spatial information, making it effective for high-dimensional and noisy images by filtering noise and generating stable features [10]. It has been widely applied in fields such as medical imaging [17] and geology [18] Due to its ability to produce informative feature representations.

A.4. Combination of CAE and CNN for classification

The combination of CAE and CNN enables efficient feature extraction and accurate classification [11]. This approach reduces the complexity of CNN without compromising accuracy, enhanced by transfer learning and dropout to prevent overfitting [9]. Its effectiveness has been proven in plant disease classification [11] and medical image classification [17], especially when labelled data is limited [18], [19].

B. Related Work

B.1. State of The Art (SOTA)

TABLE I. SOTA (STATE OF THE ART)

Source	Domain / Object	Method	Accuracy	Params/ Key Results
CAE & CNN Applications Across Domains				
[9]	Mineral Signature Classification (Geoinformation)	CAE & CNN	91%	No labels needed
[7]	Fault Diagnostics (Sensor Data)	4 CAE + CNN	No accuracy stated	Robust to noise & complex faults
[11]	Plant Disease (PlantVillage)	CAE & CNN	99.53%	0.3M Params, Efficient on limited devices
[19]	Transformer Condition Image	CAE & CNN & DCNN	96.86%	Generalizes well, removes saturation effects
[20]	Leukemia Detection (ALL cells)	CAE & CNN (VGG16 + SVM)	99.92%	Modular and Efficient Design
Fish Image Classification with Lightweight and Accurate Models				
Source	Domain / Object	Method	Accuracy	Params
[21]	Underwater Fish Detection	SCM YOLO	86.7%	0.86M
[22]	Fish Classification (Aldabra, F4K)	Dense Net + KD (Aldabra)	86.78%	7M
		Mobile Net + KD (F4K)	96.64%	2M
[23]	Fish Species Classification	FD_Net	95.29%	32M
		YOLOv5	87.93%	58M
		Faster-RCNN	87.93%	63M

B.2. SOTA Table Summary

The combination of CAE and CNN has been proven effective in various fields, such as medical diagnosis [20], plant disease detection [11], transformer condition classification [20], and mineral classification [9], with advantages in unsupervised feature extraction, noise reduction [7], [19], as well as high accuracy up to 99.9%[20], high efficiency [11], and robustness to noise [7].

In the field of fish image classification, research has focused more on lightweight CNN architectures such as SCMYOLO [21], MobileNet with knowledge distillation [22], and FD_Net [23], which achieves accuracy up to 96.64% with small parameters. However, CAE-based approaches are still rarely applied.

This study fills this gap by adopting the CAE-CNN method to improve the efficiency and accuracy of fish classification, especially to support automatic systems that are useful for coastal communities, such as in Lombok.

III. RESEARCH METHODOLOGY

This study aims to develop a fish image classification model based on feature extraction using CAE, followed by CNN as a classifier. The general methodology flow can be seen in Fig. 1.

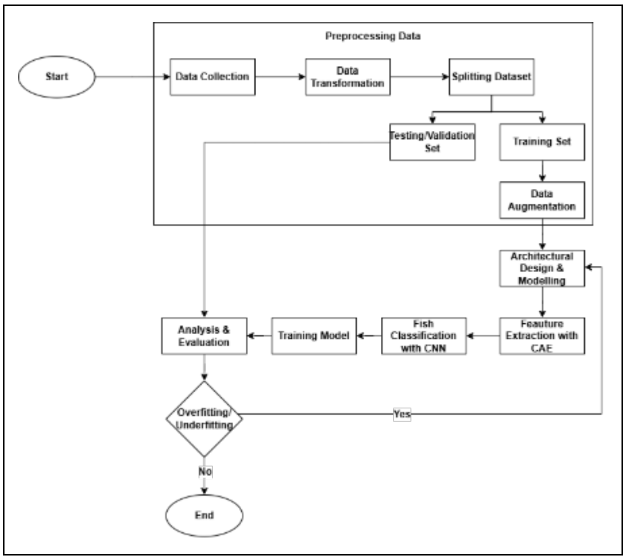


Fig. 1. Methodology Flow

A. Data Collection

The dataset used in this study comes from the Kaggle platform via the link: https://www.kaggle.com/datasets/crowww/a-large-scale-fish-dataset

This dataset consists of 9,000 colour images (RGB) of nine different fish species, each class containing 1,000 images grouped in separate folders based on their type. The images were taken using two digital cameras, namely the Kodak Easyshare Z650 (2832 × 2128 pixels) and the Samsung ST60 (1024 × 768 pixels), then resized to 590 × 445 pixels without damaging the original aspect ratio.

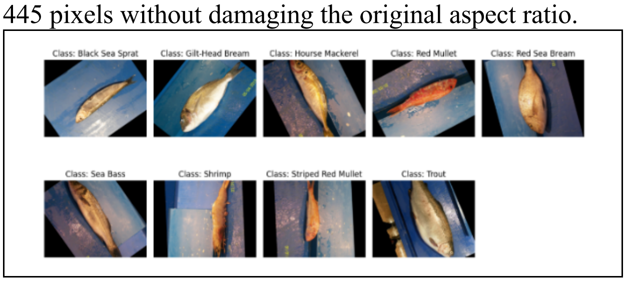


Fig. 2. One Image from Each Class of a Large-scale Dataset

This dataset has high quality and neat structure, making it suitable for image classification tasks. Several previous studies have shown its good performance, such as a simple CNN with an accuracy of 99.31% [24], Quantum-CNN with accuracy of 98.74% [25], and a TensorFlow-based CNN with 99.1% [26]In this study, the dataset is used to evaluate the effectiveness of combining CAE as a feature extractor and CNN as a classifier in automatic identification of fish species.

B. Data Preprocessing

The preprocessing step prepares the dataset so that it can be optimally used in the training process of the fish image classification.

B.1. Data Transformation

Data transformation is performed to convert the raw images into a format that suits the needs of the model [27]. This process includes the following steps.

- **Image resizing**: All images are resized to 64 × 64 pixels using the resize() function of skimage. transform to suit the architecture of the CAE and CNN models. This size was chosen based on a study of Babad Lombok image reconstruction using CAE, which showed a high accuracy of 94.57% [28].
- **Grayscale conversion**: Images are converted to grayscale format using the as_gray=True parameter in the read () function of skimage.io. The pixel values are then normalised to the range [0–1] by dividing them by 255 to simplify the visual representation and speed up the training process.

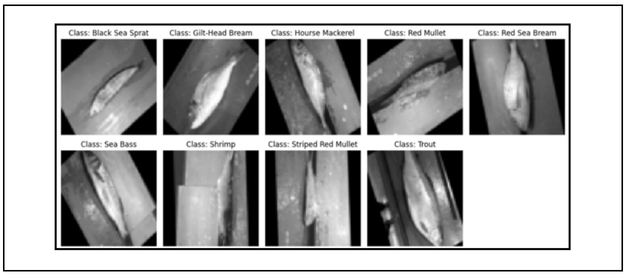


Fig. 3. Fish Dataset in Grayscale Form

This transformation is done directly during data reading, where the processed images are stored in a NumPy array of dimensions (data_count, 64, 64, 1). This step helps ensure that all data has a consistent format and is ready to be used in the classification model training process.

B.2. Splitting Dataset

The dataset is divided into two main parts, the training and testing sets. This division is done randomly using the train_test_split function from the Scikit-learn library, with the stratify parameter to ensure a balanced class distribution between the two subsets. This is important so that the test data is genuinely independent of the training data and can provide an objective picture of model accuracy.

TABLE II. Proportion of Dataset Distribution

Dataset Types	Proportion	Number of Images
Training Set	80%	7,200
Testing/Validation Set	20%	1,800

This step is part of the pre-processing to prepare the dataset before it is used in training the fish image classification model using the CAE and CNN approaches.

B.3. Data Augmentation

In this study, the data augmentation process is carried out using the ImageDataGenerator class from the Keras library. The primary purpose of this augmentation is to increase data diversity, strengthen the model's generalisation ability, and minimise the risk of overfitting [29]. The augmentation parameters used are as follows.

TABLE III. AUGMENTATION PARAMETERS

Parameter	Mark
Rotation	Up to 15 Degrees
Horizontal Translation	Up to 10%
Vertical Translation	Up to 10%
Zoom	Up to 10%
Flipping Horizontal	Enabled

C. Architectural Design and Modelling

C.1. Feature Extraction Using CAE

The feature extraction process is performed using the CAE architecture. An autoencoder is a type of neural network that is trained to reconstruct its input through two main components: an encoder and a decoder. The encoder transforms the input image into a low-dimensional feature representation (feature vector), while the decoder reconstructs the original image from that representation [30].

TABLE IV. CAE FEATURE EXTRACTION DETAILS

Layer Name	Description
Input Layer	Accepts 64×64 pixel image input with one channel (grayscale).
Conv2D-1	32 3×3 filters, ReLU. Captures basic features.
MaxPooling2D-1	Downsampling to 32×32.
Conv2D-2	64 3×3 filters, ReLU. Captures deeper features.
MaxPooling2D-2	Downsampling to 16×16.
Conv2D-3	128 3×3 filters, ReLU. Captures complex features.
MaxPooling2D-3	Downsampling to 8×8.
GlobalMaxPooling	Flatten spatial features (8×8×128 to 128).
Dense (encoded)	128 neurons, latent feature representation.
Reshape	Reshape to (1×1×128).
UpSampling2D-1	Upsampling to 8×8.
Conv2D-4	128 3×3 filters, ReLU.
UpSampling2D-2	Upsampling to 16×16.
Conv2D-5	64 3×3 filters, ReLU.
UpSampling2D-3	Upsampling to 32×32.
Conv2D-6	32 3×3 filters, ReLU.
UpSampling2D-4	Upsampling to 64×64.
Conv2D Output	1 3×3 filter, sigmoid, reconstructed result.

After the training process, only the encoder part is used to extract features from the dataset. The encoded feature representation is then input to the CNN classification model.

C.2. CNN Architecture and Model Design for Classification with CAE Input

The classification model in this study uses a CNN architecture combined with latent features extracted from the CAE model. CNN functions as a classifier to map the 128-dimensional feature representation generated by the CAE encoder into nine classes of fish species. The CNN architecture is efficient and straightforward, consisting of several fully connected layers (dense layers) with dropout to avoid overfitting. The following are the details of the CNN model architecture used:

TABLE V. CNN ARCHITECTURE DETAILS

Layer Name	Description
Input Layer	Receives a feature vector from CAE of 128 dimensions.
Dense-1	Fully connected layer with 256 neurons and ReLU activation.
Dropout	Dropout regularisation of 0.5 to reduce overfitting.
Dense-2	Fully connected layer with 128 neurons and ReLU activation.
Output Layer	The final layer comprises nine neurons (number of classes) and a softmax activation for classification.

This CNN model is compiled with the following configuration:

- **Optimiser**: Adam is used to speed up the model training process and helps find convergence points efficiently through adaptive learning rate adjustment [31].
- **Loss Function**: Categorical Crossentropy, used because the model performs multi-class classification of 9 types of fish species. This loss function measures how much error there is between the predicted label and the actual label [32].
- **Evaluation Metrics**: Accuracy monitors model performance during the training process.

C.3. CNN Without CAE Model

A CNN model without CAE was created to compare the combined use of CAE + CNN with CNN alone. In this model, CNN directly extracts features without an autoencoder.

TABLE VI. CNN FEATURE EXTRACTION DETAILS

Layer Name	Description
Input Layer	Accepts 64×64 pixel image input with one channel (grayscale).
Conv2D-1	32 3×3 filters, ReLU. Captures basic features.
MaxPooling2D-1	Downsampling to 32×32.
Conv2D-2	64 3×3 filters, ReLU. Captures deeper features.
MaxPooling2D-2	Reduce the spatial dimensions to 16×16.
Conv2D-3	128 3×3 filters, ReLU. Captures complex features.
MaxPooling2D-3	Reduces the spatial dimension to 8×8.
Flatten	Converts the multi-dimensional output into a one-dimensional vector for further processing.
Dense	Fully connected layer with 256 neurons and ReLU activation.
Dropout	Dropout 0.5 to prevent overfitting.
Output Dense	The final layer comprises nine neurons (number of classes) and a softmax activation for classification.

The model is compiled using the Adam optimizer with the categorical_crossentropy loss function and accuracy metric.

D. Model Training

During CAE training, the model was trained for 100 epochs with a batch size of 64. Next, the CNN was trained using a 128-dimensional feature representation generated

E-ISSN:2541-0806
P-ISSN:2540-8895

from the CAE encoder. The training process was carried out on the training data, while validation was carried out using the test data to evaluate the classification performance on fish images.

E. Model Evaluation

Evaluation is performed using test data to measure the model's classification performance on previously unseen fish images. Evaluation methods include:

- **Accuracy**: Measures the proportion of correct predictions to the total predictions [33].
- **Confusion Matrix**: Displays the distribution of correct and incorrect predictions per class, helping to analyze the strengths and weaknesses of the model [34].
- **Precision, Recall, and F1-Score**: Provides a more in-depth evaluation per class [35].

F. Visualization of Results

Training and testing results are visualized to understand the overall performance of the model [36], including:

- **Accuracy Graph**: Compares the accuracy of training and test data during training (epochs) to monitor overfitting/underfitting.
- **Sample Prediction**: Displays model predictions on test images to directly evaluate classification accuracy.

IV. Results and Discussion

A. Model Parameters

A.1. Combined CAE + CNN Model and Its Parameters

TABLE VII. CAE + CNN Model Parameters

Parameter	CAE	CNN
Trainable Parameter	349.313 (1.33MB)	67.081 (786 KB)
Optimizer Parameter	698.628 (2.67MB)	134.164 (524KB)
Non-Trainable Parameter	0	0
Total Parameter	1.047.941 (4MB)	201.245 (786.12KB)
CAE+CNN Parameter	**1.249.186 (4.99MB)**	

The total parameters for the combined CAE and CNN model are 1,249,186 or around 4.99 MB, making this model relatively light but still effective and accurate in performing classification. Most of the parameters are in the CAE section which functions as a feature extractor, so that the computational burden on CNN can be minimized because it only focuses on the classification process of the extracted features. This efficient architecture allows the model to be implemented in real-time classification systems and is very suitable for use on devices with limited resources.

A.2. CNN Model and Its Parameters

The CNN model used in this study is a single architecture that combines feature extraction and end-to-end classification processes to detect image patterns in fish datasets. The following are the details of the model and its parameters.

TABLE VIII. CNN Model Parameters

Parameter	CNN
Trainable Parameter	2.192.393 (8.36 MB)
Optimizer Parameter	4.384.788 (16.73 MB)
Non-Trainable Parameter	0
Total Parameter	6.577.181 (25.09 MB)

The total model parameters reach 6,577,181, with 2,192,393 trainable parameters, indicating high complexity that supports accurate classification performance.

B. Results with CAE + CNN Combination

B.1. CAE + CNN Accuracy and Loss Graph

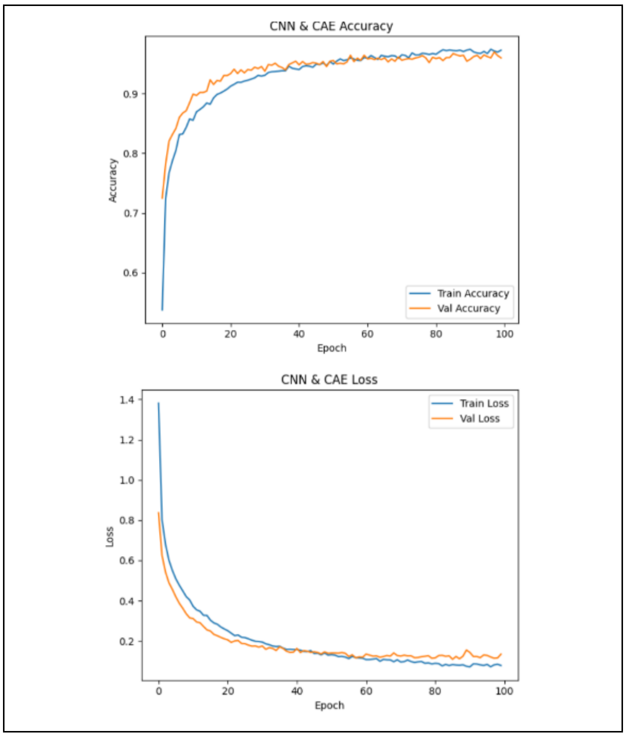


Fig. 4. CAE + CNN Accuracy and Loss Graph

Based on the graph and training results, the combined CAE and CNN model achieved an accuracy of 0.9744 and a loss of 0.0746 on the training data, as well as a validation accuracy of 0.9594 and a val_loss of 0.1345 on the 100th epoch. The accuracy graph shows a consistent increase approaching 1.0, indicating the model's ability to effectively learn important features. The small difference in accuracy between the training and validation data indicates a low risk of overfitting. Meanwhile, the loss graph decreases sharply during the first 30 epochs and is relatively stable thereafter. Overall, the model shows stable and efficient performance, thanks to the role of CAE in extracting important features as well as the optimal architecture design and training parameters.

B.2. CAE + CNN Confusion Matrix

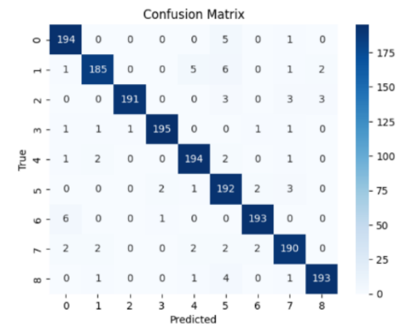


Fig. 5. CAE + CNN Confusion Matrix

The confusion matrix above shows that the combination of CAE and CNN models has excellent classification performance, as seen from the dominance of values on the main diagonal, indicating that most predictions match the original labels. Classes 0, 3, 4, 6, and 8 have almost perfect accuracy, while small errors occur in classes 1 and 2, which are sometimes mistakenly predicted as other classes. However, the number of errors is very small compared to the correct predictions. This shows that the model is able to recognize patterns accurately and has good generalization, although there is a slight influence from visual similarities between species.

B.3. CAE + CNN Classification Report

TABLE IX. CAE + CNN CLASSIFICATION REPORT

Class	Precision	Recall	F1-Score	Support
Black Sea Sprat	0.95	0.97	0.96	200
Gilt-Head Bream	0.97	0.93	0.95	200
Horseshoe Mackerel	0.99	0.95	0.97	200
Red Mullet	0.98	0.97	0.98	200
Red Sea Bream	0.96	0.97	0.96	200
Sea Bass	0.90	0.96	0.93	200
Shrimp	0.97	0.96	0.97	200
Striped Red Mullet	0.95	0.95	0.95	200
Trout	0.97	0.96	0.97	200

TABLE X. SUMMARY OF THE CAE + CNN CLASSIFICATION REPORT

Accuracy			0.96	1800
Macro Avg	0.96	0.96	0.96	1800
Weighted Avg	0.96	0.96	0.96	1800

The evaluation results of the combination of CAE and CNN models show excellent performance in fish species classification, with an overall accuracy reaching 96% of 1800 test data. The average value of precision, recall, and F1-score both macro and weighted is at 0.96, indicating consistent performance between classes. Almost all classes show high and balanced performance, such as Red Mullet and Shrimp with F1-scores approaching perfect (0.97). Although there is a slight variation between classes, the performance remains excellent.

B.4. CAE + CNN Prediction Visualization

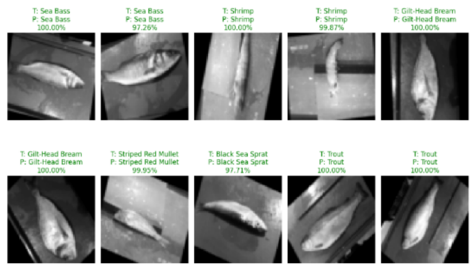


Fig. 6. CAE + CNN Prediction Visualization

The prediction visualization above shows that the CAE + CNN models successfully classify fish images very accurately, indicated by the predicted labels (P) that match the actual labels (T) and a high level of confidence, mostly above 97%. All images in this example have correct predictions, marked with green text, reflecting the model's ability to recognize the visual characteristics of fish species consistently and reliably.

B.5. CAE + CNN Time Consuming

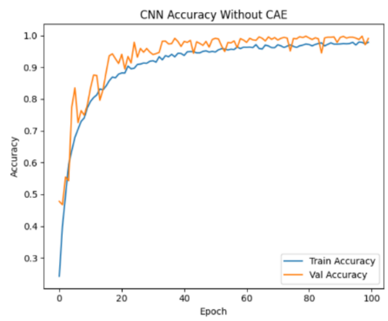
TABLE XI. CAE + CNN TESTING TIME RESULT

	Time (S)	Sample	Accuracy
Total Time for Testing	0.8799	1800	95.94%
Time for Single Image Prediction	0.7175	1	100%

The CAE + CNN model achieves 95.94% accuracy on 1800 test samples in 0.88 seconds, while single image prediction takes 0.72 seconds with 100% accuracy. This shows that batch prediction is much more efficient, making the model accurate and practical for real-world use.

C. Results Using Only CNN

C.1. CNN Accuracy and Loss Graph



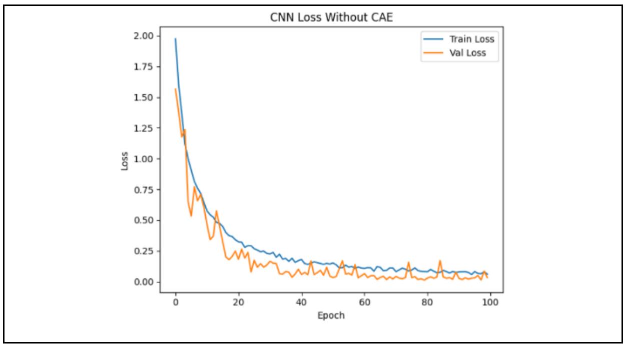


Fig. 7. CNN Accuracy and Loss Graph

Based on the graph and data, the CNN model without CAE achieved an accuracy of 0.9734 and a loss of 0.0720 on the training data, as well as a val_accuracy of 0.9906 and a val_loss of 0.0323 on the validation data at the 100th epoch. The accuracy graph shows a rapid and stable increase, approaching the maximum value, indicating that the model recognizes patterns well. The small difference in accuracy between the training and validation data indicates minimal overfitting. The loss graph also drops sharply in the first 40 epochs and remains low until the end, although there is a slight fluctuation. Overall, the CNN without CAE shows high and stable performance, proving that the optimal architecture and parameters are sufficient for accurate classification without additional features from CAE.

C.2. CNN Confusion Matrix

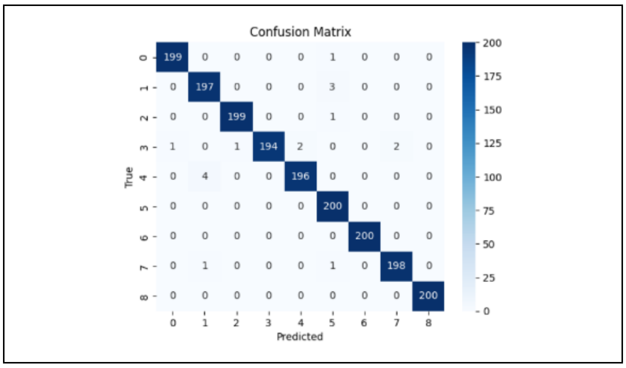


Fig. 8. CNN Confusion Matrix

The confusion matrix above shows that the CNN without the CAE model has excellent classification performance, with values dominating on the main diagonal indicating that the predictions match the original labels. Classes 5, 6, and 8 were successfully classified with perfect accuracy, while the other classes only experienced minor errors, such as classes 3 and 4 being swapped by a small amount. These errors are likely due to visual similarity between classes, but overall, the model shows high accuracy and good generalization in multi-class classification.

C.3. CNN Classification Report

TABLE XII. CNN CLASSIFICATION REPORT

Class	Precision	Recall	F1-Score	Support
Black Sea Sprat	0.99	0.99	0.99	200
Gilt-Head Bream	0.98	0.98	0.98	200
Horseshoe Mackerel	0.99	0.99	0.99	200
Red Mullet	1.00	0.97	0.98	200
Red Sea Bream	0.99	0.98	0.98	200
Sea Bass	0.97	1.00	0.99	200
Shrimp	1.00	1.00	1.00	200
Striped Red Mullet	0.99	0.99	0.99	200
Trout	1.00	1.00	1.00	200

TABLE XIII. SUMMARY OF THE CNN CLASSIFICATION REPORT

Accuracy			0.99	1800
Macro Avg	0.99	0.99	0.99	1800
Weighted Avg	0.99	0.99	0.99	1800

The evaluation results of the CNN model without CAE showed very high classification performance in nine fish classes, with an overall accuracy reaching 0.99 from 1800 test data. The average precision, recall, and F1-score values , both macro and weighted, were also 0.99, reflecting consistent performance across all classes. Some species, such as Shrimp and Trout, obtained perfect F1 scores (1.00), while other species, such as Red Mullet and Red Sea Bream, were close to perfect (0.98–0.99). These results prove that the pure CNN model is able to classify fish images accurately and reliably, and provide a strong foundation for further development of classification models.

C.4. CNN Prediction Visualization

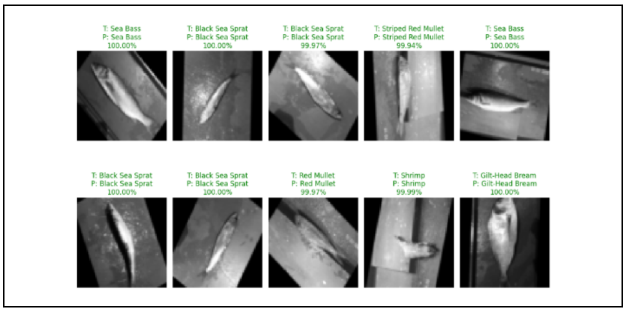


Fig. 9. CNN Prediction Visualization

The prediction visualization above shows that the CAE + CNN models successfully classify fish images very accurately, indicated by the predicted labels (P) that match the actual labels (T) and a high level of confidence, mostly above 99%. All images in this example have correct predictions, marked with green text, reflecting the model's ability to recognize the visual characteristics of fish species consistently and reliably.

C.5. CNN Time Consuming

TABLE XIV. CNN TESTING TIME RESULT

	Time (S)	Sample	Accuracy
Total Time for Testing	0.8282	1800	99.05%
Time for Single Image Prediction	0.6699	1	100%

The table shows the performance of the CNN model during testing. It took 0.8282 seconds for 1800 samples and an accuracy of 99.05% to predict a single image, which took 0.6699 seconds and an accuracy of 100%. The parallel process makes the average time per sample more efficient in a batch. This shows that CNN is also accurate and efficient enough to be applied in real scenarios.

D. Comparative Analysis

D.1. Parameters

Based on TABLE VII. And TABLE VIII. the comparison between the combined CAE + CNN model and the pure CNN model shows a striking difference in complexity and size. The CAE + CNN model has a total of 1,249,186 parameters (of which 67,081 are trainable) with a size of only about 4.99 MB, thanks to its simple architecture that utilizes CAE as a feature extractor, allowing CNN to focus only on classification. This makes it efficient and suitable for limited devices such as embedded systems or real-time applications. In contrast, the pure CNN model has 6,577,181 parameters (2,192,393 trainable) with a size of about 25 MB and a more complex architecture, including several Conv2D, MaxPooling2D, Flatten, and two Dense and one Dropout layers, which allow direct processing on raw images to fully preserve visual features. Thus, if high accuracy is a priority, then pure CNN is superior, but if efficiency and use on limited devices are more important, then CAE + CNN is a better choice.

D.2. Accuracy and Loss Graph Analysis

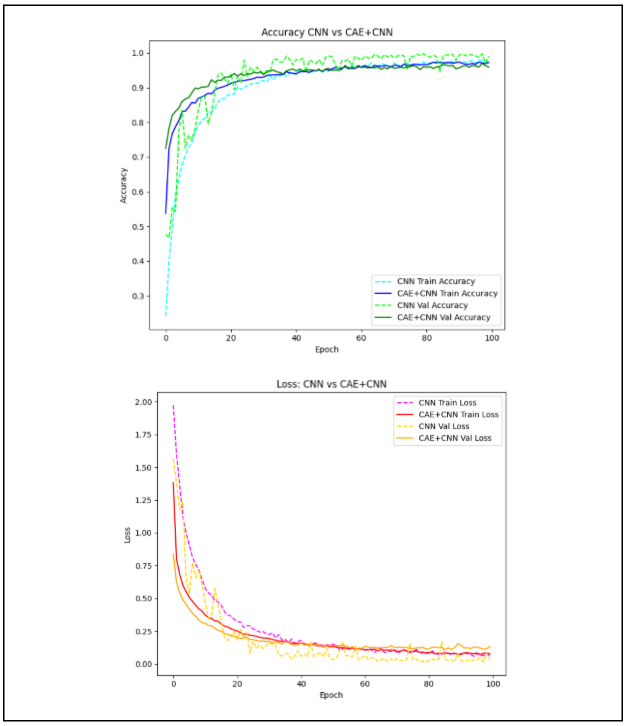


Fig. 10. Accuracy and Loss Graph Analysis

- Accuracy Graph Analysis

The combined CAE + CNN model shows more stable and consistent accuracy on training and validation data than pure CNN. The CNN validation accuracy line (dashed lime line) appears to fluctuate, indicating potential overfitting. In contrast, the CAE + CNN validation accuracy (green line) increases more smoothly and also can approaches 1.00, indicating that the features from CAE help CNN recognize patterns better.

- Loss Graph Analysis

The loss graph shows the advantages of stability and learning efficiency in the CAE + CNN model. The validation loss of CNN (gold dashed line) is higher and unstable, while CAE + CNN (orange line) decreases consistently and is lower. In the training phase, the loss of CNN (magenta line) decreases slowly from a high initial value, while CAE + CNN (red line) shows a fast and stable decrease from the beginning. This indicates that CAE successfully extracts relevant features and accelerates model convergence.

D.3. Confusion Matrix

Based on Fig. 5. and Fig. 8. The pure CNN confusion matrix shows very high classification performance, with most of the correct predictions on the main diagonal. There are very few misclassifications (1–2 cases per class), and some classes have no errors (200/200), indicating that the CNN effectively distinguishes features between classes. In contrast, in the CAE + CNN model, the number of correct predictions on the main diagonal is slightly lower, with a broader spread of errors. Although CAE makes training more stable, over-compressed representations can obscure important features and reduce the CNN's sharpness in distinguishing classes.

D.4. Classification Report

Based on TABLE X. TABLE XIII. The CNN model shows excellent classification performance with 0.99 accuracy, precision, recall, and F1-score (macro and weighted), indicating consistent and accurate results across all classes. Some classes, like Shrimp and Trout, even achieved perfect scores (1.00), meaning no misclassification.

In comparison, the CAE + CNN model performed slightly lower, with an overall accuracy of 0.96. Some classes show a slight drop to around 0.90.

Overall, CNN performs better due to direct access to full image features. The CAE+CNN model, while still strong, may lose some important visual information during encoding, although it offers more stable training behavior.

D.5. Time Consuming

Based on TABLE XI. and TABLE XIV. the pure CNN model shows superior performance in terms of both speed and accuracy compared to the combined CAE + CNN model. For testing 1800 samples, pure CNN only takes 0.8282 seconds, slightly faster than CAE + CNN, which takes 0.8799 seconds, and in unit prediction, CNN is also more efficient (0.6699 vs 0.7175 seconds). This is because

CAE + CNN requires an additional process in the form of feature extraction using an autoencoder, which, although lightweight, still adds time because it must encode and decode the input first before classification. In terms of accuracy, pure CNN is also superior with 99.05%, compared to 95.94% in CAE + CNN, although both recorded 100% accuracy in predicting one image. Overall, pure CNN is suitable for use if you prioritize high speed and accuracy, while CAE + CNN is more suitable for systems with modular architecture requirements, feature efficiency, and resistance to noise.

D.6. Summary of Comparative Analysis with Previous Research for Fish Image Classification

TABLE XV. COMPARATIVE ANALYSIS WITH PREVIOUS RESEARCH

Source	Model	Accuracy	Params
[21]	SCM YOLO	86.7%	0.86M
[22]	Dense Net + KD (Aldabra)	86.78%	7M
	Mobile Net + KD (F4K)	96.64%	2M
[23]	FD_Net	95.29%	32M
	YOLOv5	87.93%	58M
	Faster- RCNN	87.93%	63M
Our CNN	Customized CNN	99.05%	6.5M
Our CAE+CNN	**Customized CAE+CNN**	**95.94%**	**1.2M**

Based on TABLE XV. the Customized CAE+CNN model in this study offers an ideal balance between accuracy and efficiency, with an accuracy of 95.94% and only 1.2 million parameters. Although the Customized CNN model in this study produces higher accuracy (99.05%) with 6.5M parameters, the CAE+CNN approach remains superior in terms of efficiency, especially for deployment on limited devices. Compared to SCM YOLO, which has fewer parameters (0.86M) but much lower accuracy (86.7%) [21], CAE+CNN provides significant accuracy improvements with little additional complexity. In addition, compared to other models such as FD_Net (95.29, 32M) [23] and MobileNet + KD (96.64%, 2M) [22] as distillation. By leveraging the capabilities of CAE in unsupervised feature extraction and CNN as a classifier, this approach demonstrates its effectiveness as a lightweight, accurate, and suitable fish image classification solution for edge or mobile applications.

V. CONCLUSION AND RECOMMENDATIONS

This study successfully developed a fish classification model using a combination of CAE as a feature extractor and CNN as a classifier. The CAE+CNN model achieved 95.94% accuracy with a testing time of 0.87 seconds for 1,800 images, offering a lightweight and stable solution for resource-constrained devices. In comparison, the pure CNN model achieved a higher accuracy of 99.05% and faster performance, but with a much larger model size and complexity. Thus, while CNN is ideal for high-accuracy scenarios, the CAE+CNN model is more suitable for real-time applications on edge devices. Compared to other models such as SCM YOLO, FD_Net, or MobileNet + KD, the CAE+CNN model offers a competitive balance, providing high accuracy with fewer parameters and without relying on sophisticated compression or distillation techniques. This approach shows promising potential to support digital transformation in the fisheries sector, especially in helping fishermen and coastal communities identify fish species automatically.

Future research should use local fish image datasets from Indonesia, especially Lombok, to make the model more contextual and accurate for real-world applications. Additionally, optimizing CAE and CNN architectures using techniques such as pruning or lightweight models (e.g., MobileNet) is necessary for improved efficiency. The developed model can also be integrated into mobile applications or hardware systems for direct use by fishermen in the field.

REFERENCES

[1] M. Filemon Gultom, F. Hassan Rizqullah, M. Iqbal, H. Rahma Cahyadi, and M. Syahrin, “Analisis Yuridis Terhadap Strategi Geopolitik Di Indonesia Sebagai Negara Kepulauan Dalam Menghadapi Tentang Sengketa Wilayah Perbatasan,” *Adil Indonesia Journal*, vol. 5, no. 2, pp. 136–145, 2024.

[2] M. G. Putri, M. A. Rosyadi, and R. Rahmawati, “Strategi Adaptasi Nelayan Menghadapi Perubahan Iklim Masa Pandemi (Studi Kasus Nelayan Desa Tanjung, Lombok Utara),” *Seminar Nasional Sosiologi (SeNSosio Unram)*, vol. 3, pp. 60–78, 2022.

[3] M. Yusuf and A. Rakhman, “Analisis Strategi Penghidupan Dan Program Peningkatan Kesejahteraan Rumahtangga Nelayan Di Kecamatan Pringgabaya Kabupaten Lombok Timur,” *Agrimansion*, vol. 24, no. 3, pp. 725–736, Dec. 2023.

[4] A. Azis, “Identifikasi Jenis Ikan Menggunakan Model Hybrid Deep Learning Dan Algoritma Klasifikasi,” *SEBATIK*, vol. 24, no. 2, pp. 201–206, 2020, doi: 10.46984/sebatik.v24i2.1057.

[5] A. Lumute Unihehu and I. Suharjo, “Klasifikasi Jenis Ikan Berbasis Jaringan Saraf Tiruan Menggunakan Algoritma Principal Component Analysis (PCA),” *Jurnal Ilmiah Ilmu Komputer*, vol. 7, no. 2, pp. 27–32, 2021, [Online]. Available: http://ejournal.fikom-unasman.ac.id

[6] R. Uthama, Yuhandri, and Billy Hendrik, “Vision Transformer untuk Identifikasi 15 Variasi Citra Ikan Koi,” *Jurnal CoSciTech (Computer Science and Information Technology)*, vol. 5, no. 1, pp. 159–168, May 2024, doi: 10.37859/coscitech.v5i1.6711.

[7] D. Jana, J. Patil, S. Herkal, S. Nagarajaiah, and L. Duenas-Osario, “CNN and Convolutional Autoencoder (CAE) based real-time sensor one fault detection, localization, and correction,” in *Elsevier*, Dec. 2021.

[8] I. Anugrah, A. Cendekia Siregar, and B. C. Octariadi, “Perbandingan Model Arsitektur CNN Dengan Metode Transfer Learning Untuk Klasifikasi Spesies Ikan Laut,” *Progresif: Jurnal Ilmiah Komputer*, vol. 20, no. 1, pp. 444–453, 2024, doi: 10.35889/progresif.v20i1.1834.

[9] S. Zhang *et al.*, “Data-driven Mineral Prospectivity Mapping by Joint Application of Unsupervised Convolutional Auto-encoder Network and Supervised Convolutional Neural Network,” *Natural Resources Research*, vol. 30, no. 2, pp. 1011–1031, Apr. 2021, doi: 10.1007/s11053-020-09789-y.

[10] E. Pintelas, I. E. Livieris, and P. E. Pintelas, “A convolutional autoencoder topology for classification in high-dimensional noisy image datasets,” *Sensors*, vol. 21, no. 22, Nov. 2021, doi: 10.3390/s21227731.

[11] G. Boukhlifa and Y. Chibani, “A Lightweight CNN Design Based on Convolutional Autoencoder for Tomato Disease Identification,” in *Proceedings - 8th IEEE International Conference on Image and Signal Processing and their Applications, ISPA 2024*, Institute of

Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/ISPA59904.2024.10536761.

[12] A. P. Putri, R. Herawati, and N. Marlina, "Transformasi Digital Pada Pelayanan Pelelangan Ikan Di Tpi Juwana Unit Ii Kabupaten Pati (Studi Penelitian pada Aplikasi Sistem Informasi Pelelangan Ikan Pati (SIPIPA))," *Journal of Politic and Government Studies*, vol. 13, no. 4, pp. 134–147, 2024.

[13] N. Rachmat, A. Mahendra, P. Studi Informatika, and F. Imu Komputer dan Rekayasa, "Klasifikasi Jenis Ikan Laut Menggunakan Metode SVM Dengan Fitur HOG Dan HSV," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 8, no. 4, pp. 2253–2247, 2021, [Online]. Available: http://jurnal.mdp.ac.id

[14] N. Abdurrahman, B. Rahmat, and A. N. Sihananto, "Perbandingan Performa Klasifikasi Citra Ikan Menggunakan Metode K-Nearest Neighbor (K-NN) Dan Convolutional Neural Network (CNN)," *Jurnal Sistem Informasi dan Informatika (JUSIFOR)*, vol. 2, no. 2, pp. 84–93, Dec. 2023, doi: 10.33379/jusifor.v2i2.3728.

[15] Michael and W. Widhiarso, "Klasifikasi Jenis Spesies Ikan Hiu Menggunakan Metode Convolutional Neural Network (CNN)," *3RD MDP STUDENT CONFERENCE (MSC) 2024*, pp. 159–166, 2023, [Online]. Available: https://www.kaggle.com/datasets/larusso94/shark-species.

[16] T.-Y. Cheng, N. Li, C. J. Conselice, A. Aragón-Salamanca, S. Dye, and R. B. Metcalf, "Identifying Strong Lenses with Unsupervised Machine Learning using Convolutional Autoencoder," *Mon Not R Astron Soc*, vol. 494, no. 3, pp. 3750–3765, 2020, doi: 10.1093/mnras/staa1015.

[17] S. Mirzaei and P. Ghasemi, "EEG motor imagery classification using dynamic connectivity patterns and convolutional autoencoder," *Biomed Signal Process Control*, vol. 68, Jan. 2021, doi: 10.1016/j.bspc.2021.102584.

[18] J. Park and J. Jeong, "Assessment of the effectiveness of a convolutional autoencoder for digital image-based automated core logging," *Geoenergy Science and Engineering*, vol. 227, 2023, doi: doi.org/10.1016/j.geoen.2023.211802.

[19] Z. Li, Z. Jiao, A. He, and N. Xu, "A denoising-classification neural network for power transformer protection," *Protection and Control of Modern Power Systems*, vol. 7, no. 1, Dec. 2022, doi: 10.1186/s41601-022-00273-8.

[20] M. Mansoor Alhammadi, "Computer-Aided Diagnosis of Acute Lymphoblastic Leukemiaby Using a Novel CAE-CNN Framework," *Iraqi Journal for Computers and Informatics*, vol. 50, no. 2, pp. 186–206, Dec. 2024, doi: 10.25195/ijci.v50i2.502.

[21] Z. Zhang, Q. Han, W. Liu, and Y. Zhao, "A lightweight network based on SCMYOLO for accurate and efficient underwater fish detection," *ICES Journal of Marine Science*, vol. 82, no. 4, Apr. 2025, doi: 10.1093/icesjms/fsaf038.

[22] D. Salie, D. L. Brown, and K. Chieza, "Deep Neural Network Compression for Lightweight and Accurate Fish Classification Deep Neural Network Compression for Lightweight and Accurate Fish Classification," in *SACAIR 2024*, Springer, 2025. doi: https://doi.org/10.1007/978-3-031-78255-8_18.

[23] H. Malik, A. Naeem, S. Hassan, F. Ali, R. A. Naqvi, and D. K. Yon, "Multi-classification deep neural networks for identification of fish species using camera captured images," *PLoS One*, vol. 18, no. 4, Apr. 2023, doi: 10.1371/journal.pone.0284992.

[24] Pedro María Latasa, "CNN Fish Classification," Kaggle. Accessed: May 20, 2025. [Online]. Available: https://www.kaggle.com/code/pedromaralatasa/cnn-fish-classification/notebook#Fish-Species-Classification-with-EfficientNetB1

[25] Hossam Fakher, "Quantum with CNN," Kaggle. Accessed: May 20, 2025. [Online]. Available: https://www.kaggle.com/code/hossamfakher/quantum-with-cnn/notebook

[26] Fardin, "Fish Classification CNN 99.1," Kaggle. Accessed: May 20, 2025. [Online]. Available: https://www.kaggle.com/code/fardinp/fish-classification-cnn-99-1/notebook

[27] A. Fikri, B. F. Hutabarat, and U. Khaira, "Komparasi Antara Metode K-Means Clustering Dan Complete Linkage Dalam Pengelompokan Penyaluran Pinjaman Oleh Financial Technology," *Jurnal Ilmiah Media Sisfo*, vol. 17, no. 2, pp. 228–239, Oct. 2023, doi: 10.33998/mediasisfo.2023.17.2.1373.

[28] F. Syuhada, A. A. Firdaus, A. T. Ni'mah, Y. Sa'adatai, and M. Tajuddin, "Convolutional Autoencoder for Reconstruction of Historical Document Images: Ancient Manuscript Babad Lombok," *Rekayasa*, vol. 17, no. 1, pp. 175–185, Jun. 2024, doi: 10.21107/rekayasa.v17i1.26101.

[29] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J Big Data*, vol. 6, no. 1, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

[30] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," Mar. 2020, [Online]. Available: http://arxiv.org/abs/2003.05991

[31] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," in *ICLR 2015*, Dec. 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[32] M. Yusuf, Khoirunnisa, D. Kurniawan, and T. Agustin, "Klasifikasi Penyakit Tanaman Jagung Dengan Kecerdasan Buatan Berbasis Cnn," In *Seminar Nasional Amikom Surakarta (SEMNASA) 2024*, Sukoharjo, Nov. 2024, pp. 355–368.

[33] C. Miller, T. Portlock, D. M. Nyaga, and J. M. O'Sullivan, "A review of model evaluation metrics for machine learning in genetics and genomics," *Frontiers in Bioinformatics*, vol. 4, Sep. 2024, doi: 10.3389/fbinf.2024.1457619.

[34] I. wayan A. A. Wiguna, R. Rudolf Huizen, and G. Angga Pradipta, "Model Deteksi Objek Menggunakan Yolov5 untuk Pengendalian Pengaturan Lalu Lintas," *Prosiding Seminar Hasil Penelitian Informatika dan Komputer 2024*, vol. 1, no. 2, pp. 840–844, 2024.

[35] R. Merdiansah, Siska, and A. Ali Ridha, "Analisis Sentimen Pengguna X Indonesia Terkait Kendaraan Listrik Menggunakan IndoBERT," *Jurnal Ilmu Komputer dan Sistem Informasi (JIKOMSI)*, vol. 7, no. 1, pp. 221–228, 2024, Accessed: May 20, 2025. [Online]. Available: https://ejournal.sisfokomtek.org/index.php/jikom

[36] Siti Shofiah, Faris Humami, M. Iman Nur Hakim, Azimatun Lissyifa, and Agus Siswono, "Pendekatan Machine Learning untuk Analisis dan Visualisasi Data Jembatan Timbang," *Journal of Student Research*, vol. 2, no. 1, pp. 381–387, Jan. 2024, doi: 10.55606/jsr.v2i1.2666.