Jurnal Mandiri IT
Vol. 14 No. 1, July (2025), pp. 157-168

ISSN 2301-8984 (Print), 2809-1884 (Online)
Published by Institute of Computer Science

Attention-based convolutional neural networks for interpretable classification of maritime equipment

Luky Fabrianto[1], Tiwuk Wahyuli Prihandayani[2], Rasenda[3], Novianti Madhona Faizah[4]
[1]Digital Business, Nusa Mandiri University, Jakarta, Indonesia
[2] Information Systems Department, Mercu Buana University, Jakarata, Indonesia
[3]Information Systems Department, Pembangunan Nasional "Veteran" University, Jakarta, Indonesia
[4]Computer Science Department, Tama Jagakarsa University, Jakarta, Indonesia

ARTICLE INFO

Article history:

Received Jul 05, 2025
Revised Jul 16, 2025
Accepted Jul 23, 2025

Keywords:

Attention Mechanism;
CNN;
Explainability;
Grad-CAM;
Ship Components.

ABSTRACT

This study introduces a Convolutional Neural Network with an Attention Mechanism (CNN+AM), utilizing the Squeeze-and-Excitation (SE) block, to classify critical ship components: generators, engines, and oil-water separators (OWS). The SE block enhances the model's ability to focus on discriminative features, thereby improving classification performance. To overcome the limitation of the original dataset, which contained only 199 images, extensive data augmentation techniques were applied, expanding the dataset to 2,648 images. The augmented dataset was divided into training (70%), validation (15%), and testing (15%) sets to ensure reliable evaluation. Experimental results show that the CNN-AM achieved an accuracy of 72.39%, surpassing the baseline CNN model with 68.16%. These findings confirm that the attention mechanism significantly improves generalization and the ability to differentiate visually similar classes. Furthermore, the integration of interpretability tools, such as Gradient-weighted Class Activation Mapping (Grad-CAM), provides visual explanations of model predictions, increasing trust and reliability for safety-critical maritime applications. The proposed approach demonstrates strong potential for real-time ship component monitoring, offering meaningful contributions to predictive maintenance and operational safety within the maritime industry.

Corresponding Author:

Novianti Madhona Faizah,
Computer Science Department,
Tama Jagakarsa University,
Jl. TB Simatupang No.152, Kec. Jagakarsa, Kota Jakarta Selatan, DKI Jakarta 12530, Indonesia
E-mail: novianti@jagakarsa.ac.id

1. INTRODUCTION

In the maritime industry, the accurate classification of machinery such as generators, engines, and oil-water separators (OWS) is crucial for efficient maintenance, operational decision-making, and ensuring safety compliance(Sardar, 2024)(Lee et al., 2023). Traditional methods of identifying and classifying these components often rely on manual inspection, which is time-consuming, prone to human error, and heavily dependent on the availability of expert personnel. The advancement of deep learning techniques, particularly automated image classification, offers a promising solution to streamline this process and enhance its reliability(Beyer et al., 2022)(Sharma & Kumar, 2024).

However, applying deep learning to maritime machinery classification presents several challenges. A primary obstacle is the limited availability of labeled datasets. This scarcity of data can hinder the training of robust and generalizable models. Furthermore, classifying maritime machinery

often requires discerning subtle visual differences between similar components—a task known as fine-grained image classification(Mahadevkar et al., 2022). These components may share overlapping visual features, making accurate differentiation difficult. However, there is still a lack of automated, explainable, and robust classification models specifically designed for maritime machinery images, which often leads to inefficient maintenance and operational delays. This research aims to address this gap by proposing an improved deep learning approach

This study addresses these challenges by exploring and comparing the performance of machine learning architectures for maritime machinery image classification: CNN and CNN+Attention Mechanism (CNN+AM)(Mohiuddin et al., 2023). The CNN model serves as a baseline, utilizing traditional convolutional feature extraction. To further enhance performance and mitigate the limitations of a small dataset, we employ data augmentation strategies to artificially expand the training data, increasing the model's robustness and applicability (da Costa et al., 2020). Finally, we incorporate attention mechanisms into the baseline CNN (CNN+AM) to enhance interpretability by highlighting the image regions that contribute most significantly to the classification decision. The primary objective of this study is to evaluate whether the integration of attention mechanisms and data augmentation can improve classification performance and explainability compared to a baseline CNN model.

The novelty of this study lies in combining a CNN model with a Squeeze-and-Excitation (SE) attention mechanism for fine-grained classification of ship components, a domain that has received little attention in prior research. Moreover, the use of Grad-CAM for visual interpretability provides practical insights that enhance trust in AI-based maritime applications. Deep learning has transformed image classification in many fields, including industrial and maritime applications(Theodoropoulos et al., 2021). CNN have been widely adopted because they can learn hierarchical features from images (Sarvamangala et al., 2065).

Interpretability in deep learning has garnered increasing attention, with techniques such as Gradient-weighted Class Activation Mapping (Grad-CAM) providing visual explanations for model predictions(Morbidelli et al., 2020). Demonstrated the utility of Grad-CAM in highlighting critical regions that influence decision-making in CNNs, fostering trust and reliability in AI systems(Selvaraju et al., 2016). Grad-CAM technique investigates how a prediction is formed, focusing on the outputs of the last convolutional layer. Each prediction involves a weighted aggregation of the feature maps to highlight the key regions in the original image that truly drove the model's output (Moujahid et al., 2022).

In the maritime domain, studies on machinery classification remain limited. Prior works have focused on fault detection in ship engines (Wang et al., 2023) and predictive maintenance using sensor data (Shang et al., 2022). Ships require automated spare-part management to operate safely (Lee et al., 2023). The remainder of this paper is organized as follows: Section 2 reviews related works and the theoretical foundation of CNN and attention mechanisms. Section 3 describes the dataset, data augmentation strategies, and the proposed CNN+AM model. Section 4 presents the experimental results and analysis. Finally, Section 5 concludes the study and outlines future research directions.

2. RESEARCH METHOD

The dataset used in this research was created from original images of key maritime machinery components, encompassing three classes: generators, engines, and oil-water separators (OWS). The original dataset consisted of 80 images of generators, 85 images of engines, and 34 images of OWS. All images were resized into 224x224 pixels. To address the limitations posed by this relatively small original dataset and to enhance the generalization capabilities of the trained models, a comprehensive data augmentation strategy was implemented(Xu et al., 2023). This strategy included a range of geometric and photometric transformations. Input space data augmentation techniques refer to methods that involve directly altering the input image (or its components) to introduce variability, thereby enhancing the model's ability to generalize(Mumuni & Mumuni, 2022). The geometric transformations applied were random rotations (between 0° and 360°), horizontal and vertical flips, random scaling (within ±10%), random translations (within ±10% of image dimensions), and random shearing (within ±5 degrees). The photometric transformations consisted of random adjustments to brightness, contrast, and saturation (each within ±20%), as well as the addition of Gaussian noise. These augmentations were applied multiple times to each original image, resulting in a significantly expanded dataset. The final augmented dataset consisted of 2648 images, which

were then split into training, validation, and testing sets with 70%, 15%, and 15% split. Table 1 summarizes the dataset composition before and after augmentation.

Tabel I. Dataset Compostition

Class	Original	Augmented	Train (70%)	Val (15%)	Test (15%)
Generators	80	924	647	139	138
Engines	85	966	676	145	145
OWS	34	758	531	114	113

The three classes in the dataset exhibit considerable visual similarity, particularly between engines and generators. This close resemblance posed a significant challenge for several established methods; this research proposes a novel CNN model enhanced with an attention mechanism to address this fine-grained classification problem. This attention mechanism is designed to enable model to focus on the most discriminative features at images, which enhances its capacity to differentiate between visually similar maritime machinery classes, the baseline CNN architecture consists of three convolutional layers with 32, 64, and 128 filters, each using a kernel size of 3×3 and ReLU activation. Max-pooling layers with a 2×2 window follow each convolutional block to reduce spatial dimensions. The flattened output is connected to two fully connected layers with 128 and 64 neurons, respectively, followed by a dropout layer (rate 0.5) to prevent overfitting. The final output layer uses a Softmax activation function with three neurons corresponding to the classes (generators, engines, OWS). Figure 1 depicts randomly selected images from the dataset, representing generator, engine, and oil water separator (OWS) and Tabel 2 describes information and implementation detail for classification of marine equipment, the models were trained for 50 epochs with a batch size of 32. The Adam optimizer was used with an initial learning rate of 0.001, and the categorical cross-entropy loss function was applied to handle multi-class classification. Early stopping with a patience of 5 epochs was employed to avoid overfitting.

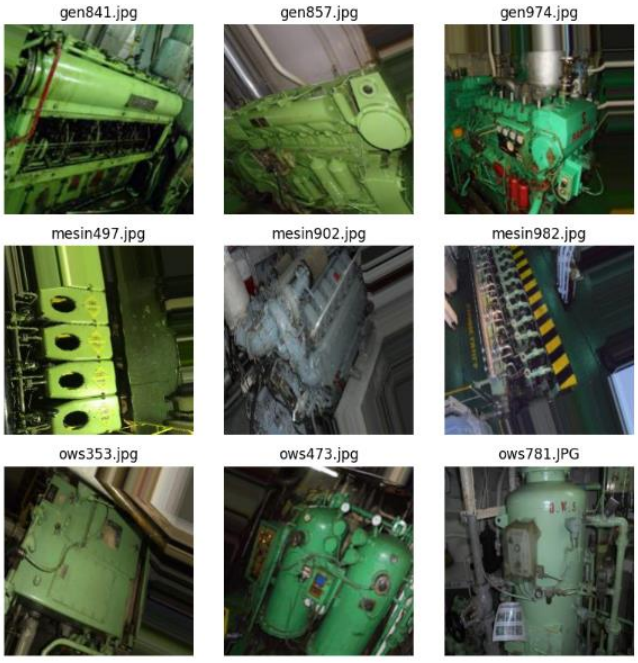

Figure 1 Example of randomly selected images from the dataset, representing generator, engine, and oil water separator (OWS)

Tabel 2. Information and implementation detail

Content	Description
Method Use	CNN + Attention Mechanism
Volume of Dataset	2684
Implementation	Python

Image Resolution in pixels	224 x 224

2.1 CNN

CNNs are form of neural network primarily used for image data processing and classification. CNN excels at tasks like image classification & object detection becuase CNN ability to automatically learn spatial hierarchies of features within images

Convolution Layer

Convolution layer uses kernels to extract features from input images.

$$O(i,j) = \sum_{m=1}^{M}\sum_{n=1}^{N} I(i+m, j+n).K(m,n) + b \quad (1)$$

Notation used in this process: *I(i,j)* represents the input pixel value at coordinates *(i,j)*. *K(m,n)* represents the kernel value at *(m,n)* position. *b* denotes bias term, which is added to the result of the convolution. *O(i,j)* represents the output after the convolution operation (feature map).

Strides and padding can affect the output size.

In image processing, particularly within convolutional operations, the output size of a feature map is affected by several factors, including the input image size (I_{size}), the kernel size (K_{size}), padding (P), and stride (S). The relationship between these factors determines the spatial dimensions of the resulting output(Z. Zhang & Peng, n.d.). Specifically, the stride dictates how many pixels the kernel shifts with each step, both horizontally and vertically. A larger stride leads to a smaller output size because the kernel covers less of the input image. Padding, on the other hand, adds extra pixels around the border of the input image. This can be crucial for controlling the output size.

$$O_{size} = \frac{(I_{size} - K_{size} + 2P)}{S} + 1 \quad (2)$$

Activation Function

Typically, activation functions such as ReLU(Hayou et al., n.d.)

$$(f(x) = \max(o, x)) \quad (3)$$

are used to add non-linearity.

Pooling Layer

Pooling layers in CNNs are responsible for reducing the spatial size of feature maps (Akhtar & Ragavendran, 2020). Two main types exist: Max Pooling takes the maximum value from each small patch, retaining prominent features and being robust to minor input variations(Gholamalinezhad & Khosravi, n.d.)(Zafar et al., 2022). Average Pooling averages each patch, providing smoother down sampling and capturing more general features. Both reduce parameters and computation, speeding up training and expanding the receptive field. Max Pooling is generally preferred for its often-superior performance.

$$O_{pool} = max/avg(I_{patch}) \quad (4)$$

Fully Connected Layer

In this layer, the data is flattened and continued with dot product operations to produce the final output(Kossaifi et al., 2020).

$$y = W.x + b \quad (5)$$

In a fully connected layer of a neural network, the output (y) is computed through a linear transformation of the input (x) using a weight matrix (W) and a bias vector (b)

Loss Function

For classification tasks, a commonly used loss function is categorical cross-entropy(Damrich & Hamprecht, n.d.)(Li et al., n.d.):

$$Loss = -\sum_{i-1}^{C} y_i . \log(\hat{y}_i) \quad (6)$$

Where C is number of classes, y_i is actual label value (0 or 1), and $\hat{y}_i$ is model prediction probability

Backpropagation and Optimization

The parameters (kernel/weights and bias) are optimized using algorithms such as Gradient Descent:

$$\theta = \theta - \eta \cdot \nabla_\theta \mathcal{L} \quad (7)$$

In machine learning, we adjust model parameters (θ) to minimize a loss function ($\mathcal{L}$). Gradient descent is a common method where we calculate the gradient of the loss ($\nabla_\theta \mathcal{L}$), which points towar`ds increasing loss. We move in the opposite direction to decrease loss. The learning rate (η) controls the step size of this movement.

2.2 The Squeeze-And-Excitation Attention Mechanism

The basic formulas of a CNN remain unchanged even with the addition of an attention mechanism. The attention mechanism enhances the network by assigning weights to features produced by CNN layers, helping the model concentrate on the most important features for classification tasks (Y. Zhang et al., n.d.)(M. Zhang et al., n.d.). Below is an explanation of how the attention mechanism, specifically the Squeeze-and-Excitation (SE) block, it is integrated into a CNN.

Feature Extraction with CNN

Initially, the image input is processed through standard CNN layers such as convolution, activation (e.g., ReLU), and pooling (e.g., Max or Average Pooling) to extract feature maps. The fundamental convolution, activation, and pooling equations are the same as in a standard CNN.

Attention Mechanism: SE Block

1. Squeeze Operation

The spatial dimensions of the feature map are reduced to channel descriptors via global average pooling, calculated as:

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X(i,j,c) \quad (8)$$

Here, z_c is the global context for channel c, H and W are the height and width of the feature map, and $X(i,j,c)$ represents the feature map value at position (i,j) in channel c. The result is a vector $z \in \mathbb{R}^C$ summarizing the global information of each channel.

2. Excitation Operation

The squeezed descriptor z is passed through two fully connected (FC) layers with ReLU and sigmoid activations:

$$s = \sigma\,(W_2 \delta(W_1 z)) \quad (9)$$

Here, $W_1 \in \mathbb{R}^{\frac{C}{r} \times C}$ and $W_2 \in \mathbb{R}^{C \times \frac{C}{r}}$ are the weight matrices for the FC layers, δ represents the ReLU activation, σ is the sigmoid function, and r is a reduction ratio (e.g., r=16). This process learns attention weights $s \in \mathbb{R}^C$, indicating the importance of each channel.

3. Scale Operation

The original feature map X is scaled using the learned attention weights s via element-wise multiplication:

$$X' = X \cdot s \quad (10)$$

Here, each channel in the feature map is multiplied by its corresponding weight, emphasizing channels with higher relevance.

4. Subsequent Layers

The scaled feature map X' is then processed through additional CNN layers, including further convolution, pooling, flattening, and fully connected layers for classification. The equations for these operations remain the same as in a standard CNN.

2.3 Evaluation

The confusion matrix offers a more detailed analysis by organizing predictions into a tabular format(Krstinić et al., 2020)(Markoulidakis YannisMarkoulidakis & Kopsiaftis, 2021). For multi-class classification, the confusion matrix extends to a K x K matrix, where K is the number of classes, with diagonal elements representing correct predictions for each class. It contains four key metrics: *TP, TN, FP*, and *FN.* Accuracy tells us the percentage of correct predictions a model makes from all predictions, calculated as the sum of True Positives (*TP*) and True Negatives (*TN*) divided by the total number of predictions, including False Positives (*FP*) and False Negatives (*FN*). It delivers a simple measure of overall correctness but may not fully reflect performance on imbalanced datasets.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (11)$$

$$\textbf{Precision}\ (Positive\ Predictive\ Value) = \frac{TP}{TP+FP} \quad (12)$$

$$\textbf{\textit{Recall}}\ (Sensitivity\ or\ True\ Positive\ Rate) = \frac{TP}{TP+FN} \quad (13)$$

$$F1 - Score = 2.\frac{Precision.Recall}{Precision+Recall} \quad (14)$$

2.4 Grad-Cam

To understand what parts within image a CNN finds most important, we use a technique called Gradient-weighted Class Activation Mapping (Grad-CAM) to make a specific prediction. It works by calculating the gradients of the class score with respect to the activations of the final convolutional layer(Morbidelli et al., 2020)(Polytechnic et al., n.d.). These gradients highlight the image regions most critical to the model's decision (Chen et al., n.d.)(Chakraborty et al., 2022). By visualizing these gradients as a heatmap overlaid on the original image, we can see which regions the CNN is focusing on to make its prediction. This provides insight into how the model makes its decisions and gain insights into its strengths and weaknesses. All experiments were conducted on a laptop equipped with an Intel Core i5-10300H CPU, 16GB RAM, and an NVIDIA GeForce MX250 GPU (2GB). The models were implemented using Python 3.10, TensorFlow 2.13, and Keras, with supporting libraries such as NumPy, OpenCV, and Matplotlib.

3. RESULTS AND DISCUSSIONS

This study evaluated the performance of a baseline Convolutional Neural Network (CNN) and an enhanced CNN with Squeeze-and-Excitation attention blocks (CNN+AM) for classifying three categories of ship machinery: generators, engines, and oil-water separators (OWS). The dataset, consisting of 199 original images, was expanded to 2,648 images using geometric and photometric data augmentation to address overfitting and improve generalization. Both models were developed in Python with TensorFlow and trained using a fixed split of 70% training, 15% validation, and 15% testing sets. As this research focused on accuracy and confusion matrix metrics, cross-validation was not employed. The baseline CNN achieved an accuracy of 68%, while the CNN+AM model achieved 72%, indicating that the addition of attention mechanisms enhances the ability to extract discriminative features in fine-grained classification tasks. The confusion matrix revealed that most misclassifications occurred between generators and engines, which share similar shapes, metallic textures, and component layouts. Variations in lighting, image angles, and background clutter also contributed to these errors, highlighting the inherent challenge of distinguishing between visually similar machinery components.

To the best of our knowledge, no previous studies have applied image-based classification to this private maritime machinery dataset, making direct quantitative comparisons with other research unfeasible. However, our findings are consistent with results from similar industrial and mechanical image classification domains, where attention-enhanced models have shown superior performance in distinguishing visually similar classes. The novelty of this research lies in its application of CNN+AM to a dataset that has not been explored before, demonstrating its potential for reliable, automated classification in maritime settings. Although cross-validation and statistical significance tests such as t-tests or ANOVA were not applied in this study, the consistent performance improvements observed in the confusion matrix support the robustness of the CNN+AM approach. Future work will focus on expanding the dataset, performing statistical validation, and incorporating interpretability methods like Grad-CAM to provide deeper insights into model decision-making and to further reduce classification errors.

3.1 CNN Implementation

The layer expects images of size 224x224 pixels with 3 color channels (RGB). It uses 32 filters (Channels), kernel size 3x3 grid, to scan the image and extract features. After each scan, it applies a ReLU activation function.

$$O(i,j,k) = \sum_{m=1}^{3}\sum_{n=1}^{3}\sum_{c=1}^{3} I(i+m, j+n, c).K(m,n,c,k) + b_k \quad (15)$$

O(i,j,k) represents the output value at a specific location *(i,j)* in the output feature map for channel *k*. *I(i,j,c)* is the input pixel value at location *(i,j)* for input channel *c*. *K(m,n,c,k)* denotes the kernel (or filter) values; it's a 3x3x3 tensor connecting input channel *c* to output channel *k*. Finally, b_k is a bias term added to each output channel *k*.

The output feature map is passed through the ReLU activation function: $(f(x) = \max(o, x))$

It shrinks the size of the data by looking at 2x2 blocks of values and keeping only the largest value in each block. Because it moves in steps of 2 (stride of 2), the output is half the size of the input in both height and width.

$$O_{pool}(i,j,k) = \max_{p,q \in \text{patch}}[O(i+p, j+q, k)] \quad (16)$$

The formulas remain the same as the first convolutional layer, but the number of filters increases to 64 and 128, respectively.

Layer 2: Conv2D(64, (3, 3), activation='relu')

Layer 3: Conv2D(128, (3, 3), activation='relu')

The spatial dimensions are flattened into a 1D vector. For example, if the output from the last convolution is (28 x 28 x128), flattening converts it into: *flattened output* $\in \mathbb{R}^{28.28.128}$

Flattened input vector of size 100352 (resulting from flattening a previous layer's output, perhaps of shape 28x28x128). Each of the 100352 inputs is connected to each of the 128 neurons.

$$y_j = \sum_{i=1}^{N} x_i . W_{ij} + b_j \quad (17)$$

These notations describe how a fully connected layer (or Dense layer) computes its output. y_j represents the output value of neuron *j*. x_i is the input value from neuron *i*. W_{ij} is the weight connecting input neuron *i* to output neuron *j*. b_j is the bias term for output neuron *j*. The output is passed through the ReLU activation function: $(f(x) = \max(0, x))$. Dropout randomly disables 50% of neurons to prevent overfitting.

The final layer of a classification neural network is the output layer. A Dense (fully connected) layer where the number of output neurons is equal to the number of classes the network is trying to predict *C*. SoftMax activation function converts the raw outputs of the neurons into a probability distribution over the classes, where each output is a value between 0 and 1.

$$\hat{y}_i = \frac{\exp(z_i)}{\sum_{j=1}^{C} \exp(z_j)} \quad (18)$$

Where, z_i is output of the last layer for class *i* and $\hat{y}_i$ is Predicted probability for class *i*.

Categorical cross entropy is a loss function commonly used in multi-class classification problems where the labels are one-hot encoded (meaning each sample's label is represented as a vector where all elements are 0 except for a 1 at the index corresponding to the correct class).

$$\mathcal{L} = -\sum_{i-1}^{C} y_i . \log(\hat{y}_i) \quad (19)$$

Here, y_i is True label for class *i* (1 for correct class, 0 otherwise) and $\hat{y}_i$ is Predicted probability for class *i*.

Optimization, Adam optimizer updates the weights using:

$$\theta = \theta - \eta . \frac{m_t}{\sqrt{v_t} + \epsilon} \quad (20)$$

in the context of training neural networks, θ represents the model's parameters (weights and biases) that are adjusted to minimize the loss. η is the learning rate, controls the size of these adjustments at each step. To improve the optimization process, techniques like momentum m_t and RMSProp v_t are often used. A constant ϵ is added to avoid division by zero o during calculations.

3.2 Attention Block SE (Squeeze-and-Excitation)

First, Global Average Pooling (Squeeze)

$$GAP(k) = \frac{1}{H \text{ x } W} \sum_{i=1}^{H} \sum_{j=1}^{W} O^{(2)}(i,j,k) \quad (21)$$

Here output dimension is 64.

Second, Fully Connected Layer 1 (Reduction),

$$SE_{reduced}(k) = ReLU\ (W_{reduced} . GAP(k) + b_{reduced}) \quad (22)$$

$W_{reduced} \in \mathbb{R}^{64x4}$ (64 channels reduced by ratio of 16) and output dimension is 4.

Third, Fully Connected Layer 2 (Expansion)

$$SE_{expanded}(k) = Sigmoid\left(W_{expanded}.SE_{reduced}(k) + b_{reduced}\right) \quad (23)$$

$W_{expanded} \in \mathbb{R}^{4x64}$ output dimension is 64.

Fourth, Channel-wise Multiplication (Excitation)

$$O_{SE}^{(2)}(i,j,k) = O^{(2)}(i,j,k).SE_{expanded}(k) \quad (24)$$

Final output dimensions: (112x112x64)

3.3 Model with Attention Block (CNN+AM)

The network begins with an input tensor X of shape 224x224x3, representing an RGB image. This input is then processed using multiple convolutional and pooling layers. The first convolutional block (Conv Block 1) applies 32 filters of size 3x3 with ReLU activation and 'same' padding, followed by max pooling with a 2x2 kernel, resulting in an output of 112x112x32. The second convolutional block (Conv Block 2) uses 64 filters with the same configuration, producing an output of 112x112x64. An attention mechanism (Squeeze-and-Excitation block) is then applied, consisting of global average pooling (squeezing to 64), a dense layer reducing dimensionality to 4, another dense layer expanding back to 64, and finally, channel-wise multiplication (excitation), maintaining the 112x112x64 output shape. Another max pooling layer reduces the dimensions to 56x56x64. A third convolutional block (Conv Block 3), similar to the previous ones but with 128 filters, is followed by another Squeeze-and-Excitation block operating on 128 channels, resulting in a 28x28x128 output. The tensor is then flattened to a vector of size 28 * 28 * 128. This flattened vector is fed into a dense layer with 128 neurons and ReLU activation, producing a 128-dimensional output. A dropout layer with a 50% dropout rate is applied. Finally, the output layer consists of a dense layer with num_classes neurons and SoftMax activation, producing the final classification output of size num_classes. Figure 2 and 3 are diagram of CNN and CNN with attention mechanism architecture.

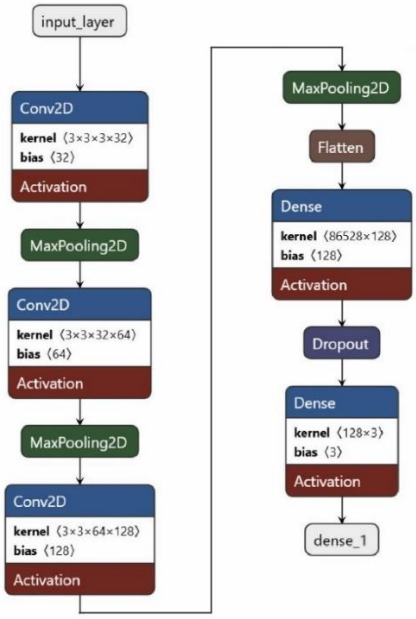

Figure 2 Baseline CNN architecture

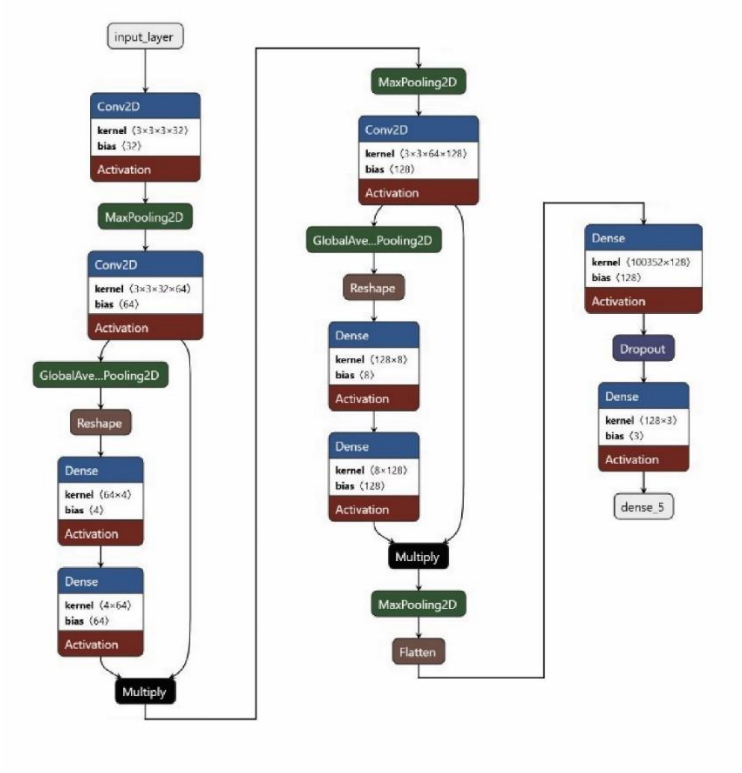

Figure 3 CNN with attention mechanism architecture

3.4 Experimental Result

The metrics used are precision, recall, f1-score, and accuracy, confusion metrics (CF). The results show that the CNN+Attention mechanism model outperforms the standard CNN across all evaluated metrics as shown at Tabel 3. Compare with Tabel 4. CNN Baseline.

Tabel 3. CNN + Attention mechanism Confusion metric

	Generator	Engine	OWS
Generator	98	28	12
Engine	35	108	6
OWS	25	4	85

Tabel 4. Baseline CNN Confusion metric

	Generator	Engine	OWS
Generator	84	36	16
Engine	33	108	8
OWS	24	8	82

This demonstrates that incorporating the attention mechanism leads to a noticeable improvement in the model's classification accuracy and balance precision and recall. The additional data provided alongside the main comparison table seems to detail the count of correct and incorrect classifications per class (Generator, Engine, OWS) by each model. In these details also, CNN+AM consistently has better classification counts compared to CNN. Tabel 5. is result recapitulation.

Tabel 5. Recapitulation result

	precision	recall	f1-score	accuracy
CNN	68.36%	68.16%	68.22%	68.16%
CNN+AM	73.24%	72.39%	72.64%	72.39%

The Grad-CAM results show heatmaps overlayed on the original images to highlight the region's most critical in the model's predictions. Using the "multiply_3" layer, the Grad-CAM effectively focuses on critical features in each image as shown at figure 4. The overlay was created with a blending method (cv2.addWeighted) combining the original images (weighted at 0.8) and the Grad-CAM heatmaps (also weighted at 0.8), providing clear visualization of the model's areas of

attention. Figure 4 is an interpretation of the Grad-CAM results which shows areas of concern in classification These visualizations validate that the model identifies relevant details specific to the input data.

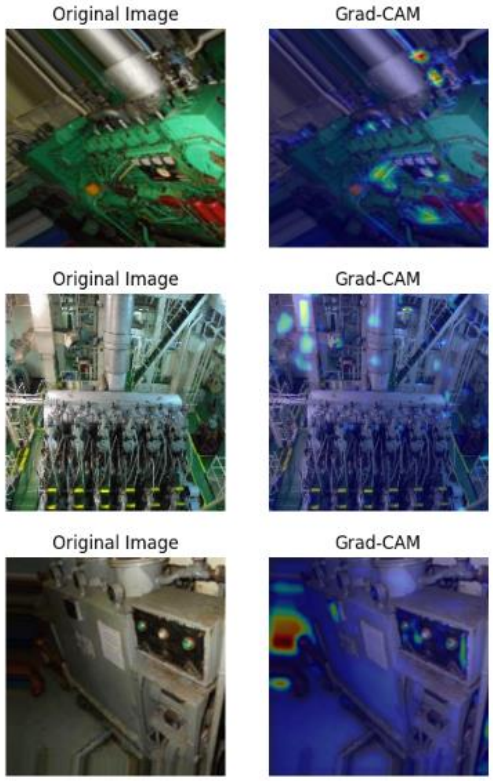

Figure 4 Grad-Cam result

4. CONCLUSION

This study demonstrates the effectiveness of integrating the SE block into a CNN for classifying ship components—Generators, Engines, and Oil-Water Separators—achieving a 4% accuracy improvement (68% to 72%) over the baseline CNN. Misclassifications, especially between generators and engines, arise from their high visual similarity, indicating the need for larger datasets, advanced attention mechanisms, and fine-grained feature extraction in future work. Grad-CAM visualizations confirmed that CNN-AM focuses on distinctive structural features, enhancing interpretability and reliability. Although direct comparisons with prior studies are limited due to the novelty of this private dataset, the findings align with trends in industrial image classification where attention-based models outperform conventional CNNs. Practically, this model can support automated maritime inspections, with potential improvements through real-time optimization techniques such as pruning, quantization, and enhanced explainability for safety-critical applications.

REFERENCES

Akhtar, N., & Ragavendran, U. (2020). Interpretation of intelligence in CNN-pooling processes: a methodological survey. *Neural Computing and Applications*, *32*(3), 879–898. https://doi.org/10.1007/s00521-019-04296-5

Beyer, L., Zhai, X., Royer, A., Markeeva, L., Anil, R., & Kolesnikov, A. (2022). Knowledge distillation: A good teacher is patient and consistent. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2022-June*, 10915–10924. https://doi.org/10.1109/CVPR52688.2022.01065

Chakraborty, T., Trehan, U., Mallat, K., & Dugelay, J. L. (2022). Generalizing Adversarial Explanations with Grad-CAM. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, *2022-June*, 186–192. https://doi.org/10.1109/CVPRW56347.2022.00031

Chen, L., Chen, J., Hajimirsadeghi, H., Mori, G., & Ai, B. (n.d.). *Adapting Grad-CAM for Embedding Networks*.

da Costa, A. Z., Figueroa, H. E. H., & Fracarolli, J. A. (2020). Computer vision based detection of external defects on tomatoes using deep learning. *Biosystems Engineering*, *190*, 131–144.

https://doi.org/10.1016/j.biosystemseng.2019.12.003
Damrich, S., & Hamprecht, F. A. (n.d.). *On UMAP's True Loss Function*.
Gholamalinezhad, H., & Khosravi, H. (n.d.). *Pooling Methods in Deep Neural Networks, a Review*.
Hayou, S., Doucet, A., & Rousseau, J. (n.d.). *On the Impact of the Activation Function on Deep Neural Networks Training*.
Kossaifi, J., Kolbeinsson, A., Khanna, A., Furlanello, T., & Anandkumar, A. (2020). Tensor Regression Networks. *Journal of Machine Learning Research*, *21*, 1–21. http://jmlr.org/papers/v21/18-503.html.
Krstinić, D., Braović, M., Šerić, L., & Božić-Štulić, D. (2020). *MULTI-LABEL CLASSIFIER PERFORMANCE EVALUATION WITH CONFUSION MATRIX*. 1–14. https://doi.org/10.5121/csit.2020.100801
Lee, C. M., Jang, H. J., & Jung, B. G. (2023). Development of an Automated Spare-Part Management Device for Ship Controlled by Raspberry-Pi Microcomputer Based on Image-Progressing & Transfer-Learning. *Journal of Marine Science and Engineering*, *11*(5). https://doi.org/10.3390/jmse11051015
Li, Z., Ji, J., Ge, Y., & Zhang, Y. (n.d.). *AutoLossGen: Automatic Loss Function Generation for Recommender Systems*. 12. https://doi.org/10.1145/3477495.3531941
Mahadevkar, S. V., Khemani, B., Patil, S., Kotecha, K., Vora, D. R., Abraham, A., & Gabralla, L. A. (2022). A Review on Machine Learning Styles in Computer Vision - Techniques and Future Directions. *IEEE Access*, *10*(September), 107293–107329. https://doi.org/10.1109/ACCESS.2022.3209825
Markoulidakis YannisMarkoulidakis, I., & Kopsiaftis, G. (2021). *Multi-Class Confusion Matrix Reduction method and its application on Net Promoter Score classification problem*. https://doi.org/10.1145/3453892.3461323
Mohiuddin, K., Welke, P., Alam, M. A., Martin, M., Alam, M. M., Lehmann, J., & Vahdati, S. (2023). Retention Is All You Need. *International Conference on Information and Knowledge Management, Proceedings*, *Nips*, 4752–4758. https://doi.org/10.1145/3583780.3615497
Morbidelli, P., Carrera, D., Rossi, B., Fragneto, P., & Boracchi, G. (2020). Augmented Grad-CAM: Heat-Maps Super Resolution Through Augmentation. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, *2020-May*, 4067–4071. https://doi.org/10.1109/ICASSP40776.2020.9054416
Moujahid, H., Cherradi, B., Al-Sarem, M., Bahatti, L., Eljialy, A. B. A. M. Y., Alsaeedi, A., & Saeed, F. (2022). Combining cnn and grad-cam for covid-19 disease prediction and visual explanation. *Intelligent Automation and Soft Computing*, *32*(2), 723–745. https://doi.org/10.32604/iasc.2022.022179
Mumuni, A., & Mumuni, F. (2022). Data augmentation: A comprehensive survey of modern approaches. *Array*, *16*(November), 100258. https://doi.org/10.1016/j.array.2022.100258
Polytechnic, N. A., Uni-, K., Engineering, M., & Technological, N. (n.d.). *Version of Record: https://www.sciencedirect.com/science/article/pii/S0010482522003420*. 1–46.
Sardar, A. (2024). *Improving safety and efficiency in the maritime industry: a multi-disciplinary approach*. https://doi.org/10.25959/26011102.V1
Sarvamangala, D. R., Raghavendra, ·, & Kulkarni, V. (2065). Convolutional neural networks in medical image understanding: a survey. *Evolutionary Intelligence*, *15*, 1–22. https://doi.org/10.1007/s12065-020-00540-3
Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2016). Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization*, *17*, 331–336. http://arxiv.org/abs/1610.02391
Shang, D., Zhang, J., Zhou, K., Wang, T., & Qi, J. (2022). Research on the Application of Visual Recognition in the Engine Room of Intelligent Ships. *Sensors*, *22*(19). https://doi.org/10.3390/s22197261
Sharma, H., & Kumar, H. (2024). A computer vision-based system for real-time component identification from waste printed circuit boards. *Journal of Environmental Management*, *351*(December 2023), 119779. https://doi.org/10.1016/j.jenvman.2023.119779
Theodoropoulos, P., Spandonidis, C. C., Giannopoulos, F., & Fassois, S. (2021). A deep learning-based fault detection model for optimization of shipping operations and enhancement of maritime safety. *Sensors*, *21*(16). https://doi.org/10.3390/s21165658
Wang, Y., Zhang, J., Zhu, J., Ge, Y., & Zhai, G. (2023). Research on the Visual Perception of Ship Engine Rooms Based on Deep Learning. *Journal of Marine Science and Engineering*, *11*(7). https://doi.org/10.3390/jmse11071450
Xu, M., Yoon, S., Fuentes, A., & Park, D. S. (2023). A Comprehensive Survey of Image Augmentation Techniques for Deep Learning. *Pattern Recognition*, *137*, 109347. https://doi.org/10.1016/j.patcog.2023.109347
Zafar, A., Aamir, M., Mohd Nawi, N., Arshad, A., Riaz, S., Alruban, A., Dutta, A. K., & Almotairi, S. (2022). A Comparison of Pooling Methods for Convolutional Neural Networks. *Applied Sciences 2022, Vol. 12, Page 8643*, *12*(17), 8643. https://doi.org/10.3390/APP12178643
Zhang, M., Gao, H., Liao, X., Ning, B., Gu, H., & Yu, B. (n.d.). *Problem Solving Protocol DBGRU-SE: predicting drug-drug interactions based on double BiGRU and squeeze-and-excitation attention mechanism*. https://doi.org/10.1093/bib/bbad184

Zhang, Y., Li, K., Li, K., & Fu, Y. (n.d.). *MR Image Super-Resolution with Squeeze and Excitation Reasoning Attention Network*.

Zhang, Z., & Peng, H. (n.d.). *Deeper and Wider Siamese Networks for Real-Time Visual Tracking*. Retrieved January 23, 2025, from https://github.com/