Journal of Electronics, Electromedical Engineering, and Medical Informatics
Homepage: jeeemi.org; Vol. 8, No. 3, July 2026, pp: 1184-1203 e-ISSN: 2656-8632

RESEARCH ARTICLE OPEN ACCESS

Non-Contact Multispectral Image Binary Classification of Water and Sodium Hydroxide Solutions Using Convolutional Neural Networks

Siti Rusdiana[1], Asep Rusyana[2], Juwita[3], Mauliza Putri[4], Aufa Rafiki[4], and Souvik Das[5]

[1] Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Syiah Kuala, Banda Aceh, Aceh, Indonesia.
[2] Department of Statistics, Faculty of Mathematics and Natural Sciences, Universitas Syiah Kuala, Banda Aceh, Aceh, Indonesia.
[3] Department of Informatics, Faculty of Mathematics and Natural Sciences, Universitas Syiah Kuala, Banda Aceh, Aceh, Indonesia.
[4] Department of Electrical and Computer Engineering, Faculty of Engineering, Universitas Syiah Kuala, Banda Aceh, Aceh, Indonesia.
[5] Department of Industrial Design, National Institute of Technology Rourkela, Rourkela, India.

Corresponding author: Siti Rusdiana (e-mail: siti.rusdiana@usk.ac.id), **Author(s) Email**: Asep Rusyana (e-mail: asep.rusyana@usk.ac.id), Juwita (e-mail: juwita@usk.ac.id), Mauliza Putri (e-mail: mauliza6@mhs.usk.ac.id), Aufa Rafiki (e-mail: aufa35@mhs.usk.ac.id), Souvik Das (e-mail: dass@nitrkl.ac.in)

Abstract **Identification of visually transparent liquids remains a challenging problem in non-contact sensing because chemically different solutions can appear nearly identical under normal observation. In laboratory and industrial environments, direct-contact chemical measurements are reliable but may require sample handling, probe calibration, cleaning, and additional processing time, which can limit their use in rapid or automated monitoring systems. This study aims to develop and evaluate a non-contact image-based classification framework for distinguishing pure water (H_2O) from sodium hydroxide solution (H_2O with NaOH) using multispectral fluctuation-pattern images. The proposed approach integrates image preprocessing, K-means segmentation, and a convolutional neural network (CNN)-based classification. A balanced dataset of 1,050 multispectral images, consisting of 525 images for each class, was used in the experiment. Each image was resized, converted to grayscale, normalized, and segmented using K-means clustering to emphasize the dominant liquid-region fluctuation pattern before classification. Three CNN architectures, namely InceptionV3, VGG19, and DenseNet201, were trained and compared under identical data-splitting and evaluation conditions. The experimental results showed that VGG19 achieved the best testing performance, with an accuracy of 97.47%, precision of 95.18%, recall of 100.00%, and F1-score of 97.53%. DenseNet201 obtained 94.30% accuracy, while InceptionV3 achieved 89.24% accuracy. These results indicate that multispectral fluctuation-pattern images contain discriminative optical information that can be learned effectively by CNN models, even when the liquid samples are visually indistinguishable to the human eye. The proposed framework demonstrates the feasibility of non-contact transparent liquid identification and may support the development of automated monitoring systems for laboratory, chemical, and industrial applications where direct sample contact is undesirable or impractical.**

Keywords **Multispectral Imaging; Transparent Liquid Classification; Sodium Hydroxide Solution; Convolutional Neural Networks; Optimization.**

I. Introduction

Water is an essential solvent in many industrial, laboratory, and chemical processes [1]. In practical applications, water is often mixed with sodium hydroxide (NaOH) to form strongly alkaline aqueous solutions with distinctive ionic conductivity behavior [2]. Although pure H_2O and H_2O with NaOH exhibit different chemical characteristics, both solutions appear visually transparent under normal observation, making direct identification difficult without supporting instruments [3]. Conventional identification methods, such as pH-based acid-base measurements, provide reliable results; however, they require direct physical contact with the sample and are therefore less suitable for automated, rapid, non-contact monitoring systems [4], [5]. Although contact-based chemical sensing

Manuscript received 4 May 2026; Revised 28 June 2026; Accepted 5 July 2026; Available online 21 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1727

methods can provide reliable measurements, their practical use may require probe-sample interaction, calibration, cleaning, and repeated sample handling, which can limit their suitability for rapid and automated non-contact monitoring [4], [5]. Conventional RGB imaging offers a non-contact alternative, but it mainly captures broad visible color and shape information through three color channels. This becomes a major limitation for transparent liquids such as pure H_2O and H_2O with NaOH, as both appear visually similar under normal observation [3]. Recent evidence also indicates that hyperspectral representation can improve classification performance compared with RGB representation; for example, Leung et al. reported that replacing RGB-3DCNN with HSI-3DCNN improved water-pollution classification accuracy from 76% to 80% [6]. In contrast, multispectral fluctuation-pattern imaging provides richer optical information by representing band-dependent intensity variations and time-dependent fluctuation patterns derived from multispectral measurements. These features can reveal subtle optical differences that are not apparent in standard RGB images, while still maintaining a non-contact image-based sensing strategy [7], [8]. Therefore, multispectral fluctuation-pattern imaging is more suitable for distinguishing visually similar transparent liquid solutions and for supporting subsequent segmentation and CNN-based feature extraction.

Recent developments in image processing and artificial intelligence have opened new opportunities for analyzing subtle visual patterns that the human eye cannot easily recognize [9], [10]. In particular, multispectral and hyperspectral imaging have shown potential for capturing the optical response and chemical characteristics of materials through spectral image representations [7], [8]. This capability is especially relevant for transparent liquid discrimination because spectral and fluctuation-based representations can provide discriminative information beyond ordinary color appearance. In image-based classification frameworks, segmentation is an important step because it isolates relevant regions from the background and improves the quality of feature extraction [11], [12], [13]. Among commonly used segmentation methods, K-means clustering is widely applied due to its simplicity and effectiveness in grouping pixels based on intensity similarity [11], [12], [13]. For classification, convolutional neural networks have demonstrated strong capability in learning complex spatial representations from image data [14], [15]. Among widely used deep architectures, InceptionV3, VGG19, and DenseNet-based models have all shown strong performance in image classification applications [16], [17], [18].

Despite these advances, several limitations remain. First, many conventional liquid identification techniques are still contact-based and therefore less suitable for non-invasive applications [4], [5]. Second, although related non-contact liquid and chemical identification studies have reported promising classification performance, generally in the range of approximately 91–96%, most of them focused on transparent object detection, general liquid-content recognition, hyperspectral chemical analysis, or signal-based sensing rather than binary image-based discrimination of visually similar transparent liquid solutions [3], [5], [7], [8], [9]. Third, although CNN-based methods have been widely used in image classification, the use of segmented multispectral fluctuation-pattern images for binary classification of pure H_2O and H_2O with NaOH remains insufficiently investigated, particularly under a unified comparison of deep CNN architectures [19], [20]. Therefore, the specific research gap addressed in this study is the lack of a quantitatively evaluated, non-contact, image-based framework that combines a multispectral fluctuation-pattern representation, segmentation, and CNN classification to distinguish two visually indistinguishable transparent liquid classes.

To address these limitations, this study proposes a non-contact multispectral image classification framework that combines grayscale conversion, resizing, normalization, K-means segmentation, and convolutional neural network classification. The dataset used in this study consists of multispectral fluctuation-pattern images representing pure H_2O and H_2O with NaOH solutions. These images were prepared from previously acquired spectral measurements and further processed for classification in the present work. Three CNN architectures are employed to learn discriminative visual features from the segmented images [16], [17], [18]. The novelty of the proposed framework lies not only in applying CNNs, but also in integrating multispectral fluctuation-pattern image generation, K-means-based region isolation, and a controlled comparative evaluation of InceptionV3, VGG19, and DenseNet201 for non-contact transparent liquid discrimination. Therefore, the aim of this study is to develop and evaluate a reliable non-contact classification method for distinguishing pure H_2O and H_2O with NaOH solutions from multispectral fluctuation-pattern images. The proposed method is expected to support transparent liquid identification in situations where direct sample contact is undesirable or impractical. The main contributions of this study are as follows:

1. It proposes a non-contact classification framework for distinguishing H_2O and H_2O with NaOH solutions using multispectral fluctuation-pattern images.

Manuscript received 4 May 2026; Revised 28 June 2026; Accepted 5 July 2026; Available online 21 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1727

2. It integrates K-means segmentation with deep convolutional classification to enhance the extraction of relevant visual features from transparent liquid images [11], [12], [13].
3. It compares the performance of three CNN architectures, namely InceptionV3, VGG19, and DenseNet201, on the same multispectral dataset [16], [17], [18].
4. It provides a quantitative evaluation of the proposed framework using accuracy, precision, recall, specificity, F1-score, MCC, confidence intervals, and confusion matrix analysis to determine the most reliable CNN architecture for non-contact transparent liquid classification.

The remainder of this paper is organized as follows. Section II describes the materials and methods, including the dataset, preprocessing pipeline, segmentation stage, CNN architectures, and evaluation procedure. Section III presents the experimental results. Section IV discusses the main findings and their implications. Finally, Section V concludes the paper and outlines possible directions for future work.

II. Materials and Methods

This section describes the proposed framework, including dataset preparation, image preprocessing, K-means segmentation, data splitting and augmentation, CNN-based classification, and performance evaluation. The overall workflow is illustrated in Fig. 1, whereas Algorithm 1 provides the complete training and testing procedure, covering preprocessing, segmentation, training-only augmentation, CNN training, validation-based model selection, testing, and metric evaluation. This procedural description was added to improve the reproducibility and clarity of the proposed framework across InceptionV3, VGG19, and DenseNet201.

Algorithm 1. Complete training and testing procedure of the proposed framework.

Input: Multispectral fluctuation-pattern image dataset $D = \{(X_i, y_i)\}_{i=1}^{N}$, where X_i denotes the i-th image and $y_i \in \{\mathrm{H_2O}, \mathrm{H_2O + NaOH}\}$ denotes its class label; CNN architecture set $A = \{\mathrm{InceptionV3}, \mathrm{VGG19}, \mathrm{DenseNet201}\}$; number of clusters $K = 3$; image size 224×224 pixels; batch size $B = 32$; learning rate $\alpha = 0.0001$; maximum epochs $E = 30$.

Output: Trained CNN models, predicted class labels $\hat{y}_i$, confusion matrices, and evaluation metrics.

1. Load the prepared multispectral fluctuation-pattern image dataset D.
2. For each image $X_i \in D$, perform preprocessing and segmentation:
   2.1 Resize X_i to 224×224 pixels.
   2.2 Convert X_i from RGB to grayscale using the luminance-based grayscale formulation.
   2.3 Normalize pixel intensities to the range $[0, 1]$.
   2.4 Apply K-means segmentation with $K = 3$ to separate the solution region, container edge, and background.
   2.5 Select the dominant solution-region cluster and generate the binary mask $M(x, y)$.
   2.6 Apply morphological closing to refine the segmented solution region.
   2.7 Save the segmented image as the final CNN input representation X_i'.
3. Split the processed dataset $D' = \{(X_i', y_i)\}_{i=1}^{N}$ into training, validation, and testing subsets using a stratified split strategy.
4. Apply data augmentation only to the training subset using rotation, flipping, translation, and scaling operations.
5. For each CNN architecture $a \in A$:
   5.1 Initialize model M_a using ImageNet pretrained weights.
   5.2 Replace the final classification layer with a two-class softmax output layer.
   5.3 Train M_a using the augmented training subset, categorical cross-entropy loss, and Adam optimizer.
   5.4 Monitor validation performance at each epoch.
   5.5 Select the model checkpoint with the best validation performance.
   5.6 Evaluate the selected model using the independent testing subset.
   5.7 Generate the predicted class label using the maximum softmax probability: $\hat{y}_i = \arg\max_{c \in \{\mathrm{H_2O}, \mathrm{H_2O+NaOH}\}} p(y = c \mid X_i')$
   5.8 Compute the confusion matrix, accuracy, precision, recall, specificity, F1-score, MCC, and the 95% confidence interval for accuracy.
6. Compare the testing performance of InceptionV3, VGG19, and DenseNet201.
7. Identify the best-performing CNN architecture based on testing accuracy, F1-score, and class-wise prediction performance.

As shown in Algorithm, data augmentation was applied only to the training subset, while the validation and test subsets remained unchanged. This design was used to reduce the risk of overfitting and prevent information leakage during model evaluation, which is important when evaluating deep learning models on limited datasets [21], [22].

A. Input Multispectral Images

The input data used in this study consisted of multispectral fluctuation-pattern images representing two transparent liquid classes, namely pure H_2O and H_2O with NaOH. The dataset was obtained from

Manuscript received 4 May 2026; Revised 28 June 2026; Accepted 5 July 2026; Available online 21 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1727

Journal of Electronics, Electromedical Engineering, and Medical Informatics
Homepage: jeeemi.org; Vol. 8, No. 3, July 2026, pp: 1184-1203 **e-ISSN: 2656-8632**

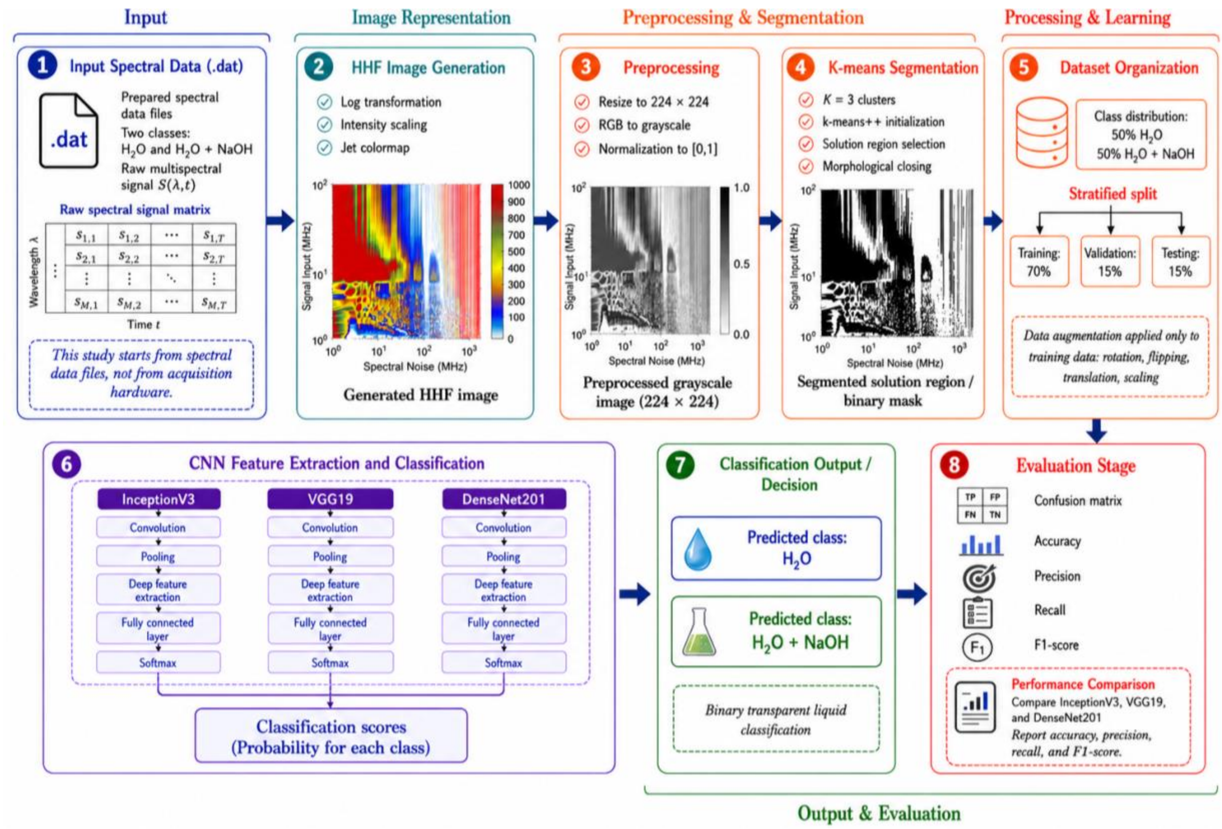


Fig. 1. Overall workflow of the proposed non-contact multispectral fluctuation-pattern image classification framework. The input images represent two transparent liquid classes, pure H_2O and H_2O with NaOH. The workflow includes image preprocessing, K-means segmentation of the liquid region, dataset splitting, CNN-based feature extraction and classification using InceptionV3, VGG19, and DenseNet201, class prediction, and performance evaluation.

previously acquired multispectral measurements of pure H_2O and H_2O with NaOH solutions using a Multi-Spectrum Chemical Sensing (MSCS) system. The original spectral data were recorded in .dat format and then converted into two-dimensional fluctuation-pattern images using logarithmic transformation, linear intensity scaling, and color mapping. This conversion was performed to enhance the visual representation of spectral intensity variations and to provide an image-compatible input for subsequent preprocessing, segmentation, and CNN-based classification.

The transformation from unprocessed spectral information to two-dimensional fluctuation-pattern images was conducted as follows. Let $S(\lambda,t)$ represent the spectral intensity captured in the wavelength band λ and time sample t. A logarithmic transformation was first applied (Eq. (1)):

$$F(\lambda,t) = \log_{10}(1 + S(\lambda,t)) \quad (1)$$

where $F(\lambda,t)$ denotes the log-transformed fluctuation value. The addition of 1 prevents undefined values when the spectral intensity is zero, while the logarithmic operation compresses the dynamic range of the original spectral response. This step is important because raw multispectral measurements may contain large intensity variations that can dominate the image representation and reduce the visibility of subtle fluctuation patterns. Therefore, the log transformation helps preserve relative fluctuation differences across wavelength bands and time samples. The resulting matrix F, consisting of M wavelength bands and N time samples, was then linearly scaled to the range [0,255]. This scaling converts the transformed spectral-fluctuation matrix into a standard image intensity range, allowing the data to be represented as a two-dimensional image. The scaled matrix was subsequently converted to an RGB image using a jet colormap, with lower and higher fluctuation values represented by different color intensities. In practical terms, this process transforms the original spectral-time measurement $S(\lambda,t)$ into a multispectral fluctuation-pattern image that can be processed using image preprocessing, K-means segmentation, and CNN-based feature extraction.

The prepared dataset contained 1,050 images in total, comprising 525 images of pure H_2O and 525 images of H_2O with NaOH. All images were prepared in a uniform format prior to preprocessing and

Manuscript received 4 May 2026; Revised 28 June 2026; Accepted 5 July 2026; Available online 21 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1727

Journal of Electronics, Electromedical Engineering, and Medical Informatics
Homepage: jeeemi.org; Vol. 8, No. 3, July 2026, pp: 1184-1203 **e-ISSN: 2656-8632**

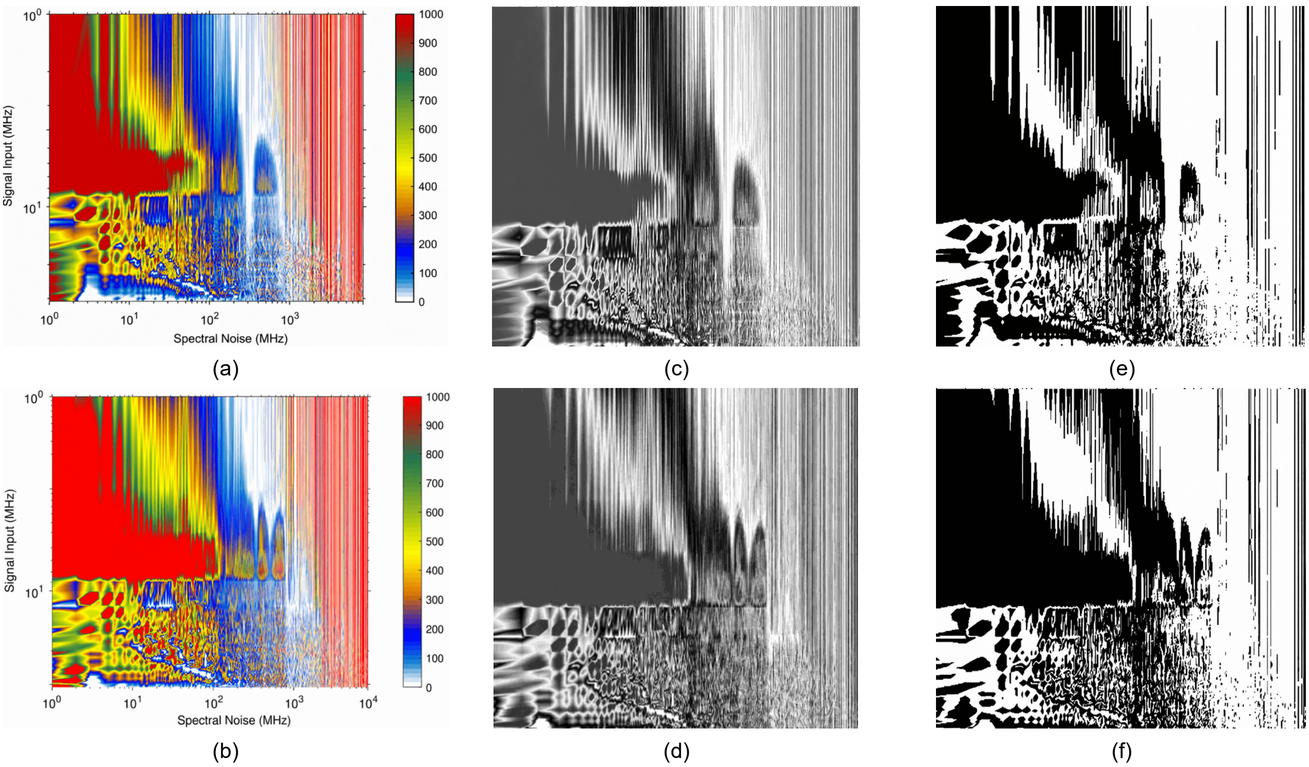


Fig. 2. Representative image preprocessing and segmentation results for the two transparent liquid classes. Panels (a), (c), and (e) show pure H_2O, while panels (b), (d), and (f) show H_2O with NaOH. Panels (a) and (b) show the original multispectral fluctuation-pattern images, panels (c) and (d) show the grayscale-converted images, and panels (e) and (f) show the K-means segmentation results used as CNN input.

classification. Representative examples of the input multispectral fluctuation-pattern images, grayscale images, and K-means segmentation results are presented in Fig. 2. Visually, the H_2O with NaOH class tends to exhibit more irregular and higher-contrast fluctuation structures than the pure H_2O class. This indicates that the addition of NaOH may influence the optical response pattern captured in the multispectral image representation, thereby providing discriminative visual information for subsequent preprocessing, segmentation, and classification stages.

B. Image Preprocessing

As shown in Fig. 1, the preprocessing stage in this study consists of resizing images to 224 × 224 pixels, converting from RGB to grayscale, and normalizing intensities to the range [0, 1]. These operations were performed to standardize the multispectral fluctuation-pattern images before segmentation and classification. In image-based classification frameworks, preprocessing is essential because it improves data consistency and facilitates subsequent feature extraction [11], [12], [13], [23], [24].

1. Image Resizing

All multispectral fluctuation-pattern images were resized to 224 × 224 pixels to provide uniform input dimensions for InceptionV3, VGG19, and DenseNet201. Standardizing image size is important for reducing computational complexity, stabilizing model training, and maintaining consistent spatial representation across samples. This step also helps preserve the main fluctuation-pattern region while preparing the images for further processing.

2. RGB-to-Grayscale Conversion

After resizing, each image was converted from RGB to grayscale to simplify the visual representation to a single intensity channel. This transformation emphasizes luminance-based fluctuation structures while reducing channel redundancy. The grayscale conversion was performed using Eq. (2):

$$I = 0.299R + 0.587G + 0.114B \quad (2)$$

where R, G, and B denote the red, green, and blue intensity components, respectively. The converted grayscale images were then used as inputs for the subsequent segmentation stage. Representative grayscale outputs are presented in Fig. 2.

3. Intensity Normalization

Following grayscale conversion, pixel intensities were normalized to the range [0,1] to improve numerical stability and reduce scale differences among samples. This normalization step helps prevent large intensity variations from dominating the learning process and

Manuscript received 4 May 2026; Revised 28 June 2026; Accepted 5 July 2026; Available online 21 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1727

provides a more consistent input distribution for segmentation and CNN-based classification. Recent reviews on image harmonization and data preparation also emphasize that normalization and related preprocessing strategies are important for improving the consistency, comparability, and downstream reliability of AI-based image analysis [23], [24]. The normalized images were subsequently forwarded to the K-means segmentation stage. Let $I(x, y)$ denote the grayscale intensity value at pixel location (x, y). The normalized pixel value $I_n(x, y)$ was computed as Eq. (3).

$$I_n(x, y) = \frac{I(x, y) - I_{\min}}{I_{\max} - I_{\min}} \tag{3}$$

where $I_{\min}$ and $I_{\max}$ represent the minimum and maximum intensity values in the image, respectively. This transformation scales the pixel intensity values into the range $[0, 1]$, thereby improving numerical stability before segmentation and CNN-based classification. The grayscale formulation in Eq. (2) was used to summarize RGB intensity information into a single luminance-based channel, thereby reducing redundant color-channel information while preserving the main fluctuation structures. The normalization in Eq. (3) was applied to standardize the pixel intensities to [0,1], improving numerical stability and ensuring that the K-means segmentation and CNN models operate on comparable input scales. The parameters $I_{\min}$ and $I_{\max}$ are obtained directly from each image, so the normalization is adaptive to the intensity range of individual samples. In practice, this step reduces scale-related bias across images and improves the consistency of the input representation before segmentation and classification.

C. K-Means Segmentation

K-means segmentation was applied to separate the main solution region from the background by grouping image pixels according to intensity similarity. In image analysis, K-means remains a practical, widely used clustering-based segmentation method due to its simplicity, computational efficiency, and ability to partition pixels into meaningful regions prior to feature extraction [11], [12], [13], [25], [26]. In this study, the number of clusters was chosen as $K = 3$, representing three key intensity regions in the multispectral fluctuation-pattern images: the liquid solution region, the container edge, and the background. Cluster centroids were initialized using the k-means++ strategy to improve convergence stability and reduce sensitivity to poor initialization. The algorithm was iterated until convergence, and the cluster corresponding to the dominant solution region was selected based on its mean intensity. A binary mask was then generated to isolate the solution region from the background. Morphological closing was subsequently applied to fill small holes and smooth region boundaries before the segmented image was forwarded to the CNN-based classification stage. Let x_j denote the intensity value of the j-th pixel and let μ_i denote the centroid of the cluster i. The K-means algorithm aims to minimize the within-cluster sum of squares, which can be expressed as Eq. (4).

$$J = \sum_{i=1}^{K} \sum_{x_j \in C_i} \| x_j - \mu_i \|^2 \tag{4}$$

where J is the clustering objective function, K is the number of clusters, and C_i is the set of pixels assigned to cluster i. In addition to minimizing the within-cluster sum of squares, the K-means optimization procedure consists of two main iterative steps: pixel assignment and centroid update. For each pixel feature x_j, the cluster label is assigned based on the nearest centroid as Eq. (5).

$$c_j^{(t)} = \arg \min_{i \in \{1,\dots,K\}} \| x_j - \mu_i^{(t)} \|^2 \tag{5}$$

where $c_j^{(t)}$ denotes the cluster index assigned to the j-th pixel at iteration t, and $\mu_i^{(t)}$ represents the centroid of the cluster i at iteration t. After the assignment step, each centroid is updated by computing the mean of all pixels assigned to the corresponding cluster (Eq. (6)):

$$\mu_i^{(t+1)} = \frac{1}{|C_i^{(t)}|} \sum_{x_j \in C_i^{(t)}} x_j \tag{6}$$

where $|C_i^{(t)}|$ represents the number of pixels assigned to cluster C_i at iteration t. The iteration is continued until the centroid displacement between two consecutive iterations becomes smaller than a predefined tolerance (Eq. (7)):

$$\max_i \| \mu_i^{(t+1)} - \mu_i^{(t)} \| < \epsilon \tag{7}$$

where $\epsilon = 1 \times 10^{-4}$ was used as the convergence threshold. In this study, the maximum number of iterations was set to 300 to prevent excessive computation when convergence was slow. After convergence, the selected solution-region cluster was converted into a binary segmentation mask (Eq. (8)):

$$M(x, y) = \begin{cases} 1, & \text{if } c(x, y) = c_s \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where $M(x, y)$ is the binary mask at pixel location (x, y), $c(x, y)$ is the cluster label of the pixel, and c_s denotes the cluster corresponding to the dominant solution region. This formulation clarifies how the K-means segmentation stage isolates the liquid-region pattern before CNN-based feature extraction. Clustering-based segmentation is commonly used to partition image pixels into meaningful regions and improve feature extraction in computer vision tasks [11], [12], [13], [26]. The parameter $K = 3$ was selected based on

Manuscript received 4 May 2026; Revised 28 June 2026; Accepted 5 July 2026; Available online 21 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1727

the expected visual structure of the multispectral fluctuation-pattern images, namely the liquid solution region, the container edge, and the background. This choice provides sufficient separation of the region of interest without introducing excessive fragmentation. The k-means++ initialization was used to improve centroid initialization and reduce sensitivity to poor starting points, while the convergence threshold $\epsilon = 1 \times 10^{-4}$ and a maximum iteration limit of 300 was selected to balance segmentation stability and computational efficiency. The practical significance of Eqs. (4)–(8) is that they convert the preprocessed image into a liquid-region mask, allowing the CNN models to focus on the dominant solution area rather than background or container-related artifacts. In the multispectral fluctuation-pattern images used in this study, differences in the intensity distribution between pure H_2O and H_2O with NaOH enable isolation of the dominant solution region from the background. The segmented output provides a clearer representation of the fluctuation-pattern structure, thereby facilitating feature extraction and improving the quality of the subsequent classification stage [11], [12], [13], [26]. Representative segmentation results are shown in Fig. 2.

D. Dataset Split and Data Augmentation

After preprocessing and segmentation, the dataset was divided into training, validation, and testing subsets using a stratified split strategy to preserve class balance across the two liquid categories. The training set was used for model learning, the validation set for monitoring model performance and selecting the best configuration during training, and the testing set for final evaluation on unseen data. A proper separation of training and testing data is important because different sampling configurations may affect performance estimates and model selection, particularly when the dataset size is limited [21]. The sample distribution is summarized in Table 1.

Table 1. Stratified image-level dataset split for the two transparent liquid classes using an approximately 70%:15%:15% training–validation–testing ratio.

Class	Training	Validation	Testing	Total
H_2O	367 (70%)	79 (15%)	79 (15%)	525 (100%)
H_2O with NaOH	367 (70%)	79 (15%)	79 (15%)	525 (100%)
Total	734 (70%)	158 (15%)	158 (15%)	1050 (100%)

The stratified split was applied at the image level using randomized assignment. As all images were derived from a single acquisition campaign using a fixed MSCS setup, acquisition-level separation between subsets was not enforced. This represents a limitation of the current dataset, and future work should consider session-level stratification to further reduce potential information leakage across subsets. To improve model generalization and reduce overfitting, data augmentation was applied only to the training subset. The augmentation process included rotation, flipping, translation, and scaling, which introduced controlled variations in object orientation and positioning without changing the semantic class of the image. In image-classification tasks, augmentation is widely used to expand the effective feature space and improve model robustness when the available dataset is limited [22]. The augmentation scheme used in this study is illustrated in Fig. 3. The geometric augmentation process can be generally represented using an affine transformation as Eq. (9).

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{9}$$

where (x, y) and (x', y') denote the original and transformed pixel coordinates, respectively, $a_{11}, a_{12}, a_{21}, a_{22}$ represent rotation, scaling, or flipping parameters, and t_x and t_y denote translation shifts. This formulation represents the controlled geometric transformations applied to the training images.

The affine transformation in Eq. (9) provides a unified mathematical representation of the geometric augmentation operations used in this study, including rotation, flipping, translation, and scaling. These transformations were applied only to the training subset to increase visual variability without altering the semantic class of the liquid image. The parameters a_{11}, a_{12}, a_{21}, and a_{22} represent geometric changes such as rotation, scaling, or flipping, while t_x and t_y represent translation shifts. In practice, this augmentation strategy helps reduce overfitting and enables CNN models to learn more robust fluctuation-pattern features from a limited dataset.

By applying augmentation only to the training data, the validation and testing subsets remained unchanged and could therefore provide a more reliable estimate of model generalization. This strategy helps the model learn more invariant visual features from multispectral fluctuation-pattern images while preventing information leakage into the evaluation subsets [21], [22].

E. CNN-Based Classification

After preprocessing, segmentation, and training data augmentation, the resulting images were used as inputs for CNN-based classification. The objective of this stage was to distinguish two classes, namely pure H_2O and H_2O with NaOH, from their multispectral fluctuation-pattern images. Convolutional neural networks are widely used in image classification

Manuscript received 4 May 2026; Revised 28 June 2026; Accepted 5 July 2026; Available online 21 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1727

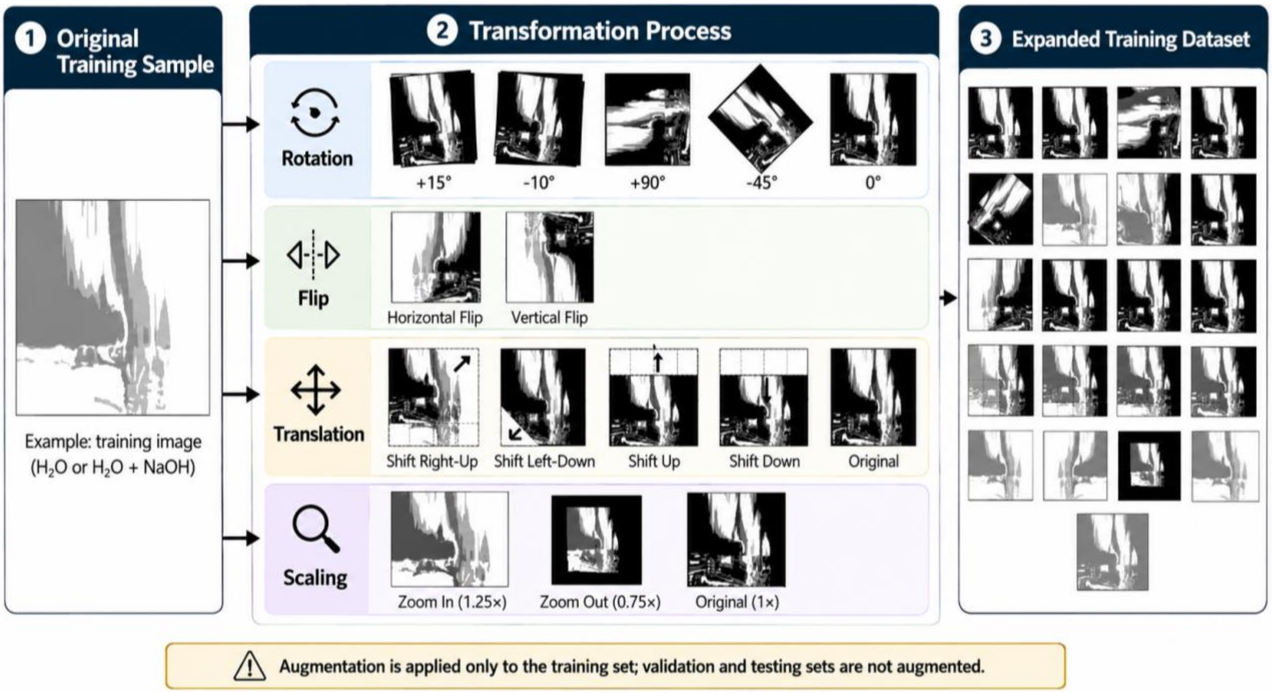


Fig. 3. Data augmentation scheme applied only to the training subset. The augmentation process includes rotation, horizontal or vertical flipping, translation, and scaling to increase visual variability while preserving the original class labels. The validation and testing subsets were not augmented.

because they can automatically learn hierarchical spatial features from visual input data [14], [15]. In this study, three architectures were selected for comparative evaluation, namely InceptionV3, VGG19, and DenseNet201, because these models have shown strong performance across image-classification tasks [16], [17], [18], [19], [20].

Each architecture was trained independently using the same training, validation, and testing subsets to ensure a fair comparison. InceptionV3 was included for its effectiveness in extracting multi-scale visual features, VGG19 was used for its stable deep convolutional structure, and DenseNet201 was selected for its dense inter-layer connectivity, which supports efficient feature propagation [16], [17], [18]. The three CNN architectures are shown in Fig. 4, Fig. 5, and Fig. 6, respectively. All three CNN architectures (InceptionV3, VGG19, and DenseNet201) were initialized with ImageNet pretrained weights and fine-tuned end-to-end on the multispectral fluctuation-pattern dataset, with no layers frozen during training. This transfer learning strategy provides a stronger initialization than random weight assignment and is commonly adopted to improve convergence and performance on datasets of limited size. To provide a clearer mathematical representation of the evaluated architectures, the main feature extraction mechanisms of InceptionV3, VGG19, and DenseNet201 are formulated based on their architectural characteristics. InceptionV3 performs multi-scale feature extraction through parallel convolutional branches with different receptive fields. The output of an Inception module can be expressed as Eq. (10).

$$F_{\text{Inc}}(x) = \text{Concat}\big(f_{1\times1}(x), f_{3\times3}(x), f_{5\times5}(x), f_{\text{pool}}(x)\big) \quad (10)$$

where x denotes the input feature map, $f_{1\times1}(x)$, $f_{3\times3}(x)$, and $f_{5\times5}(x)$ represent convolutional operations with different kernel sizes, $f_{\text{pool}}(x)$ denotes the pooling branch, and $\text{Concat}(\cdot)$ represents feature-map concatenation. This structure enables InceptionV3 to capture visual features at multiple spatial scales [16]. For VGG19, feature extraction is performed through a sequence of stacked convolutional layers with small kernels. The transformation at the l-th convolutional layer can be written as Eq. (11).

$$F_{\text{VGG}}^{(l)} = \sigma\left(W^{(l)} * F_{\text{VGG}}^{(l-1)} + b^{(l)}\right) \quad (11)$$

where $F_{\text{VGG}}^{(l-1)}$ and $F_{\text{VGG}}^{(l)}$ denote the input and output feature maps of the layer l, $W^{(l)}$ is the convolution kernel, $b^{(l)}$ is the bias term, $*$ represents the convolution operation, and $\sigma(\cdot)$ denotes the nonlinear activation function. This stacked convolutional structure enables hierarchical feature extraction from low-level intensity patterns to deeper spatial representations [17]. For DenseNet201, each layer receives feature maps from all preceding layers through dense connectivity. This mechanism can be formulated as Eq. (12).

$$F_{\text{Dense}}^{(l)} = H_l\left(\left[F^{(0)}, F^{(1)}, \dots, F^{(l-1)}\right]\right) \quad (12)$$

Manuscript received 4 May 2026; Revised 28 June 2026; Accepted 5 July 2026; Available online 21 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1727

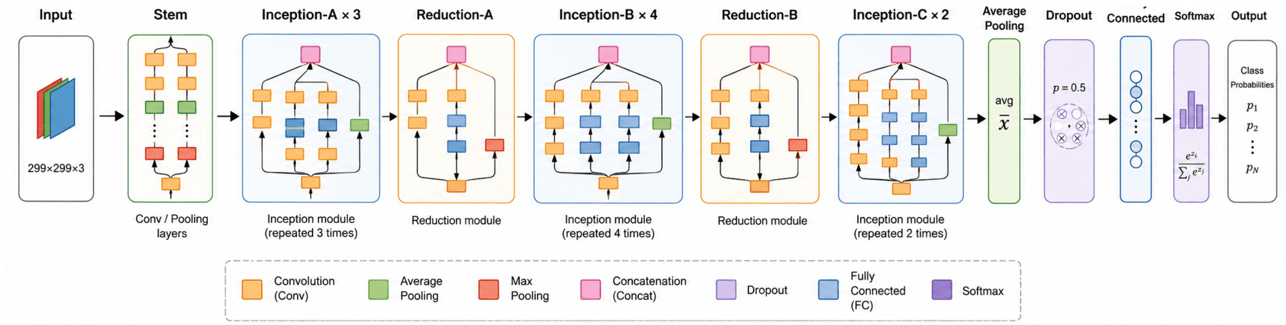


Fig. 4. InceptionV3-based transfer-learning architecture used for binary classification of multispectral fluctuation-pattern images. The model receives 224 × 224 × 3 input images, uses an ImageNet-pretrained InceptionV3 feature extractor, and replaces the original classification layer with a two-class softmax output layer for H_2O and H_2O with NaOH classification.

where $F^{(l)}_{\text{Dense}}$ is the output of the l-th layer, $H_l(\cdot)$ represents a composite transformation consisting of convolution, activation, and normalization operations, and $\left[F^{(0)}, F^{(1)}, \dots, F^{(l-1)}\right]$ denotes the concatenation of feature maps from all previous layers. This dense connection strategy improves feature reuse and supports information propagation across deeper layers [18].

For a more complete mathematical representation of the CNN learning process, the general convolutional operation at the layer l can be expressed as Eq. (13).

$$H^{(l)}_{i,j,k} = \sigma\left(\sum_{m=1}^{M} \sum_{u=1}^{r} \sum_{v=1}^{r} W^{(l)}_{u,v,m,k}\, H^{(l-1)}_{i+u,j+v,m} + b^{(l)}_{k}\right) \quad (13)$$

where $H^{(l)}_{i,j,k}$ is the output feature value at the spatial location (i, j) and channel k, $H^{(l-1)}$ is the input feature map from the previous layer, $W^{(l)}$ is the convolution kernel, $b^{(l)}_k$ is the bias term, r is the kernel size, M is the number of input channels, and $\sigma(\cdot)$ denotes the activation function. This operation allows the CNN to learn local intensity and texture patterns from segmented multispectral fluctuation images [10], [14], [15]. The max-pooling operation used to reduce spatial dimensions can be written as Eq. (14).

$$P^{\max}_{i,j,k} = \max_{(u,v)\in\Omega} H^{(l)}_{i+u,j+v,k} \quad (14)$$

where Ω denotes the local pooling window. In some CNN blocks, average pooling can also be represented as Eq. (15).

$$P^{\text{avg}}_{i,j,k} = \frac{1}{|\,\Omega\,|} \sum_{(u,v)\in\Omega} H^{(l)}_{i+u,j+v,k} \quad (15)$$

where $|\,\Omega\,|$ is the number of elements in the pooling region. Pooling reduces feature-map resolution while preserving dominant discriminative responses, thereby lowering computational cost and improving spatial robustness. The nonlinear activation function used in convolutional feature extraction can be represented by the rectified linear unit (Eq. (16)):

$$\sigma(z) = \max(0, z) \quad (16)$$

where z is the pre-activation value. The activation function introduces nonlinearity into the CNN model, enabling it to learn complex visual differences between pure H_2O and H_2O with NaOH fluctuation patterns. Eqs (10)–(16) describe the main feature-extraction mechanisms used by the evaluated CNN architectures. InceptionV3 uses parallel convolutional branches to capture multi-scale visual responses, VGG19 applies stacked small-kernel convolutions to learn hierarchical spatial patterns, and DenseNet201 reuses features through dense inter-layer connectivity. The general convolution operation in Eq. (13) learns local intensity and texture patterns from segmented fluctuation images, while pooling in Eqs. (14) and (15) reduce spatial resolution and help preserve dominant discriminative responses. The ReLU activation in Eq. (16) introduces nonlinearity, enabling CNN models to

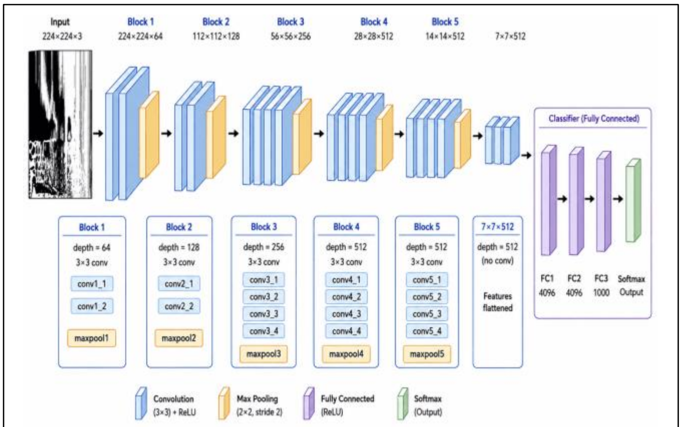


Fig. 5. VGG19-based transfer-learning architecture used in this study. The 224 × 224 segmented multispectral fluctuation-pattern images are passed through an ImageNet-pretrained VGG19 feature extractor, followed by a modified classification head with a two-class softmax output for distinguishing H_2O from H_2O with NaOH.

Manuscript received 4 May 2026; Revised 28 June 2026; Accepted 5 July 2026; Available online 21 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1727

learn complex differences between pure H_2O and H_2O with NaOH images. Practically, these operations transform the segmented liquid-region image into increasingly abstract feature representations for classification. To ensure consistency across models, all CNN architectures were trained using the same basic hyperparameter configuration. The images were resized to 224 × 224 pixels before being used as model inputs, and each model was trained using a batch size of 32, a learning rate of 0.0001, and a maximum of 30 epochs, as summarized in Table 2. Validation performance was monitored at each epoch to identify the most suitable model checkpoint and to reduce the risk of overfitting [19], [20].

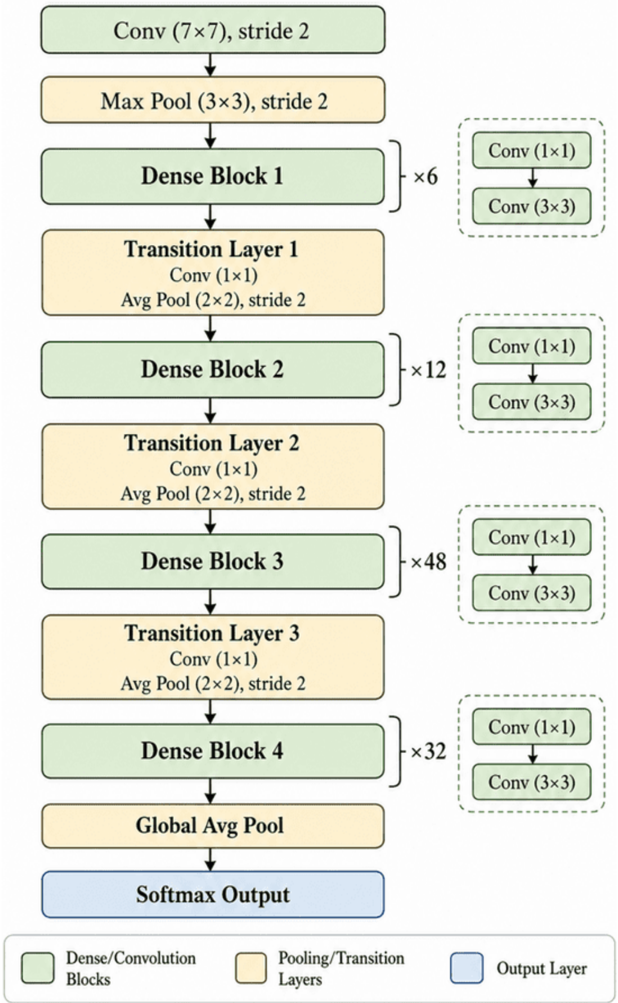


Fig. 6. DenseNet201-based transfer-learning architecture used for binary transparent liquid classification. The model uses 224 × 224 segmented multispectral fluctuation-pattern images as input, an ImageNet-pretrained DenseNet201 feature extractor, and a modified two-class softmax classifier to predict H_2O or H_2O with NaOH.

For an input image x, the output of the final classification layer was represented as a class probability using the softmax function (Eq. (17))

$$p(y = c \mid x) = \frac{e^{z_c}}{\sum_{k=1}^{K} e^{z_k}} \tag{17}$$

where z_c denotes the logit for class c, and K is the number of classes. In this study, $K = 2$, corresponding to the H_2O and H_2O with NaOH classes. The model parameters were optimized using categorical cross-entropy loss (Eq. (18)).

$$L = -\sum_{c=1}^{K} y_c \log p(y = c \mid x) \tag{18}$$

where y_c is the ground-truth class label. These probability outputs were used to determine the final predicted class for each CNN model.

Table 2. CNN training hyperparameter configuration.

Parameter	Setting
CNN architectures	InceptionV3, VGG19, DenseNet201
Input image size	224 × 224 pixels
Number of classes	2
Pretrained weights	ImageNet
Fine-tuning strategy	End-to-end fine-tuning
Frozen layers	None; all layers were trainable
Batch size	32
Initial learning rate	0.0001
Optimizer	Adam
Adam parameters	$\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$
Maximum epochs	30
Output activation	Softmax
Loss function	Categorical cross-entropy
Data augmentation	Rotation, horizontal/vertical flipping, translation, and scaling
Augmented subsets	Training subset only
Validation criterion	Best validation performance, monitored at each epoch
Model checkpoint	Best-validation checkpoint
Early stopping	Enabled with a patience of 5 epochs
Random seed	42 for randomized dataset splitting
Number of training runs	1
Software environment	Python 3.12.12, TensorFlow 2.19.0, and Keras 3.10.0
Hardware environment	Kaggle Notebook with an Intel Xeon CPU @ 2.00 GHz, NVIDIA Tesla P100-PCIE-16GB GPU, and 31.35 GB system RAM
Hyperparameter consistency	Identical settings were applied to all three CNN models

During model training, the network parameters θ were optimized by minimizing the categorical cross-

Manuscript received 4 May 2026; Revised 28 June 2026; Accepted 5 July 2026; Available online 21 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1727

entropy loss. The basic gradient-based update direction is obtained through backpropagation as Eq. (19).

$$g_t = \nabla_\theta L_t(\theta_t) \tag{19}$$

where g_t denotes the gradient of the loss function L_t with respect to the model parameters θ_t at iteration t. The Adam optimizer was used because it adaptively estimates first- and second-order moments of the gradient, which supports stable convergence during CNN training. The first-moment estimate is given by Eq. (20)

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)g_t \tag{20}$$

and the second-moment estimate is computed as Eq. (21)

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)g_t^2 \tag{21}$$

where m_t and v_t are the first- and second-moment estimates, respectively, while β_1 and β_2 are exponential decay rates. To correct initialization bias, the moment estimates are adjusted as Eq. (22) and Eq. (23).

$$\widehat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{22}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{23}$$

The CNN parameters are then updated using Eq. (24).

$$\theta_{t+1} = \theta_t - \alpha\frac{\widehat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon} \tag{24}$$

where α is the learning rate and ε is a small constant used to avoid division by zero. In this study, $\alpha = 0.0001$ was used for all CNN architectures to ensure a consistent training configuration. The learning rate was set to 0.0001 for all CNN architectures to maintain a stable and comparable fine-tuning process across InceptionV3, VGG19, and DenseNet201. Adam was selected because it adaptively estimates the first- and second-order moments of the gradient, allowing more stable parameter updates during deep CNN training. The default Adam parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-7}$ were used to improve optimization stability and reproducibility. The practical significance of Eqs. (19)–(24) defines the gradient computation and adaptive parameter updates used to minimize the classification loss during CNN training. Finally, the predicted class was determined using the maximum posterior probability from the softmax output (Eq. (25)):

$$\hat{y} = \arg\max_{c\in\{1,\dots,K\}} p(y = c \mid x) \tag{25}$$

where $\hat{y}$ is the predicted liquid class, $K = 2$ represents the two classes, and $p(y = c \mid x)$ is the predicted probability for class c. This decision rule assigns each multispectral fluctuation-pattern image to either the H_2O or H_2O with NaOH class based on the highest softmax probability.

F. Performance Evaluation

The performance of the individual CNN models was evaluated using the testing subset [27]. The evaluation was conducted to quantify the ability of each model to distinguish pure H_2O and H_2O with NaOH from multispectral fluctuation-pattern images. The same evaluation procedure was applied to InceptionV3, VGG19, and DenseNet201 to ensure a fair comparison across all classification approaches. The evaluation metrics used in this study included accuracy, precision, recall, specificity, F1-score, Matthews correlation coefficient (MCC), 95% confidence intervals for accuracy, and confusion matrix analysis. These metrics were selected because they provide complementary information about classification performance. Accuracy measures the overall proportion of correctly classified samples, precision measures the reliability of positive predictions, recall measures the ability of the model to identify positive samples, and F1-score provides a balanced measure between precision and recall. The confusion matrix was used to analyze class-wise correct and incorrect predictions, including true positives, true negatives, false positives, and false negatives [28].

Let TP, TN, FP, and FNdenote true positive, true negative, false positive, and false negative predictions, respectively. In this study, the classification performance was calculated using the following Eq. (26) to Eq. (29):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{26}$$

$$Precision = \frac{TP}{TP + FP} \tag{27}$$

$$Recall = \frac{TP}{TP + FN} \tag{28}$$

$$F1\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{29}$$

For binary classification, one class can be treated as the positive class when computing TP, FP, and FN, while the other class serves as the negative class. In the final reporting, the confusion matrix was used to verify the number of correct and incorrect predictions for both H_2O and H_2O with NaOH classes. This evaluation scheme allowed the performance of the three CNN models to be compared under the same testing conditions [28].

The metrics in Eqs. (26)–(29) were used to provide complementary interpretations of binary classification performance. Accuracy measures the overall proportion of correct predictions, while precision indicates the reliability of positive-class predictions. Recall measures the ability of the model to identify positive samples, and the F1-score provides a

Manuscript received 4 May 2026; Revised 28 June 2026; Accepted 5 July 2026; Available online 21 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1727

balanced measure between precision and recall. These metrics are practically important because a model may achieve high accuracy while still producing class-specific errors. Therefore, the combination of accuracy, precision, recall, F1-score, specificity, MCC, and confusion matrix analysis provides a more complete evaluation of the CNN models under the same testing condition. The final performance results were then summarized in the Results section using a performance comparison table and confusion matrix visualizations. This separation ensures that the present section describes only the evaluation procedure, while the numerical outcomes are reported and interpreted in the Results and Discussion sections. In addition to accuracy, precision, recall, and F1-score, specificity and Matthews Correlation Coefficient (MCC) were also calculated as complementary metrics for interpreting binary classification performance. Specificity, defined as $TN\,/\,(TN\,+\,FP)$, measures the model's ability to correctly identify negative-class samples, whereas MCC provides a single balanced measure that accounts for true positives, true negatives, false positives, and false negatives. These metrics are useful because high accuracy alone may not fully represent class-wise prediction behavior. Since the present study evaluates model performance using confusion matrix-based metrics at a single operating point, ROC-AUC analysis, formal statistical comparisons between models, and repeated experiments with different random seeds are recommended for future work.

III. Results

This section presents the experimental results obtained from the individual CNN models, including training and validation performance, testing performance, confusion matrix analysis, and best-performing model selection.

A. Training and Validation Performance

The training and validation performance of InceptionV3, VGG19, and DenseNet201 was monitored across epochs to evaluate learning stability, convergence behavior, and potential overfitting. The training histories of the three CNN architectures are presented in Fig. 7, including the accuracy and loss curves for each model. Overall, all three CNN architectures showed increasing training and validation accuracy, accompanied by decreasing training and validation loss. This pattern indicates that the models were able to learn discriminative fluctuation-pattern features from the multispectral images. The final

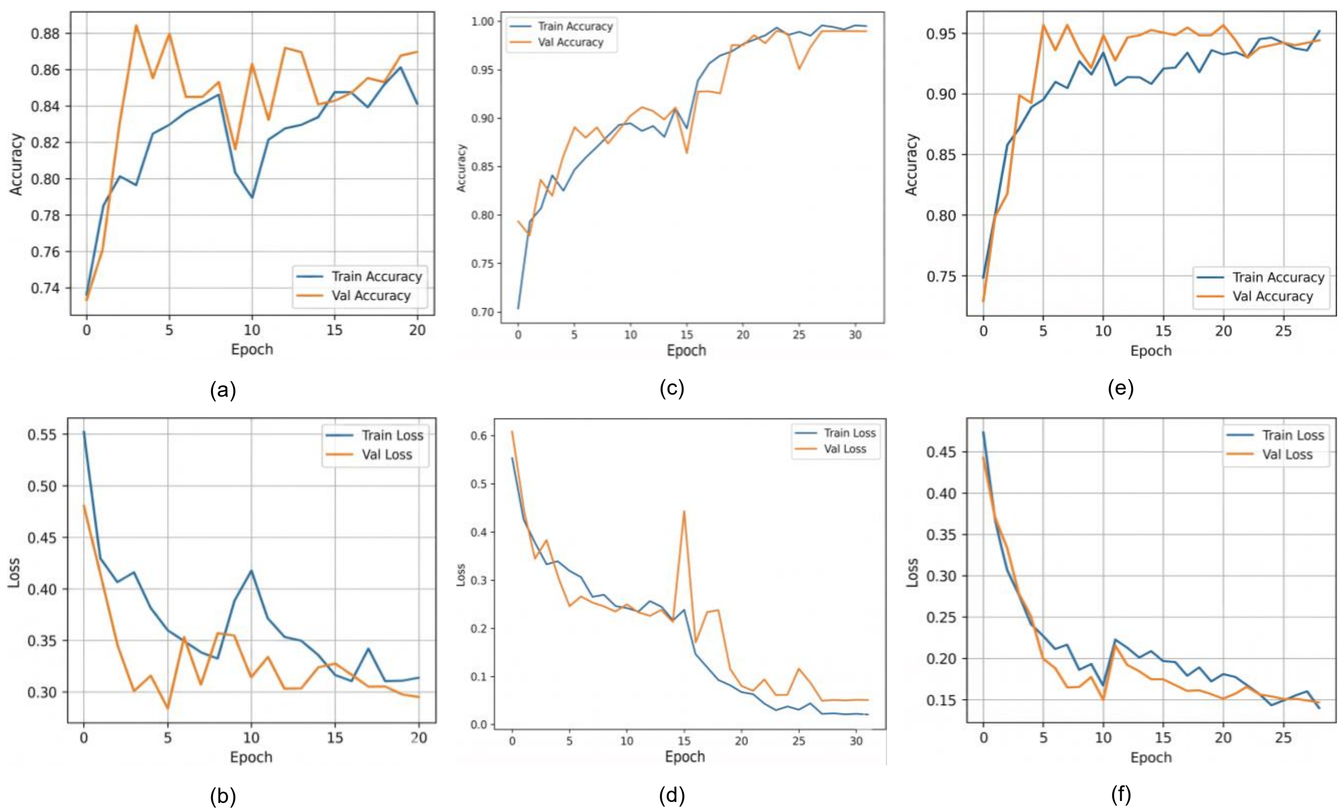


Fig. 7. Training and validation performance curves of the evaluated CNN architectures across epochs. Panels (a) and (b) show InceptionV3 accuracy and loss, panels (c) and (d) show VGG19 accuracy and loss, and panels (e) and (f) show DenseNet201 accuracy and loss. Each plot compares training and validation curves to assess convergence behavior and possible overfitting.

Manuscript received 4 May 2026; Revised 28 June 2026; Accepted 5 July 2026; Available online 21 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1727

training and validation performance estimated from these curves is summarized in Table 3. Together, the training curves and summarized metrics indicate that VGG19 showed the most favorable convergence behavior, as reflected by its high training and validation accuracy and low final loss values. DenseNet201 also demonstrated stable learning, with training and validation curves remaining relatively close throughout training. In contrast, InceptionV3 showed greater fluctuation in validation accuracy, suggesting less stable learning behavior compared with VGG19 and DenseNet201. The final testing performance of each model is discussed in the next subsection.

Table 3. Training and validation performance of the CNN models at the final recorded epoch.

Metric	Inception V3	VGG19	DenseNet 201
Training accuracy	0.84	0.99	0.95
Validation accuracy	0.87	0.98	0.94
Training loss	0.31	0.03	0.15
Validation loss	0.30	0.05	0.14

B. Testing Performance of Individual CNN Models

The testing performance of the three CNN architectures was evaluated using the testing subset to assess their ability to classify unseen multispectral fluctuation-pattern images. The confusion matrices of InceptionV3, VGG19, and DenseNet201 are presented in Fig. 8, providing class-wise prediction outcomes for the H_2O and H_2O with NaOH classes. InceptionV3 correctly classified 74 H_2O samples and 67 H_2O with NaOH samples, with 5 and 12 misclassifications, respectively. VGG19 achieved the best performance with 79 correct H_2O samples and 75 correct H_2O with NaOH samples, resulting in only 4 total errors. DenseNet201 correctly classified 76 H_2O samples and 73 H_2O with NaOH samples, with 3 and 6 misclassifications, respectively. The derived evaluation metrics are summarized in Table 4. Precision, recall, and F1-score were computed by treating the H_2O class as the positive class. Among the evaluated models, VGG19 achieved the highest accuracy of 97.47%, followed by DenseNet201 with 94.30% and InceptionV3 with 89.24%. VGG19 also obtained the highest precision of 95.18%, recall of 100.00%, and F1-score of 97.53%. In terms of comparative performance, VGG19 improved accuracy by 8.23 percentage points over InceptionV3 and by 3.17 percentage points over DenseNet201, corresponding to relative improvements of 9.22% and 3.36%, respectively. DenseNet201 also outperformed InceptionV3 by 5.06 percentage points, equivalent to a relative improvement of 5.67%. From a class-wise perspective, VGG19 eliminated all misclassifications in the H_2O class and reduced errors in the H_2O with NaOH class from 12 in InceptionV3 and 6 in DenseNet201 to 4, corresponding to reductions of 66.67% and 33.33%, respectively. InceptionV3 exhibited the highest error rate, particularly for the H_2O with NaOH class, while DenseNet201 showed moderate but asymmetric misclassification behavior.

Overall, VGG19 provided the most stable discrimination performance for multispectral fluctuation-pattern images, particularly in reducing false predictions for chemically modified transparent solutions. It should be noted that the testing results in Table 4 were obtained using a fixed train-validation-test split and one training run for each CNN architecture. Therefore, the reported metrics represent model performance under the controlled experimental setting used in this study. Repeated experiments using different random seeds and formal statistical significance testing were not conducted in the current work. The 95% confidence intervals reported in Table 4 quantify uncertainty in test-set accuracy and do not represent performance variability across repeated training runs.

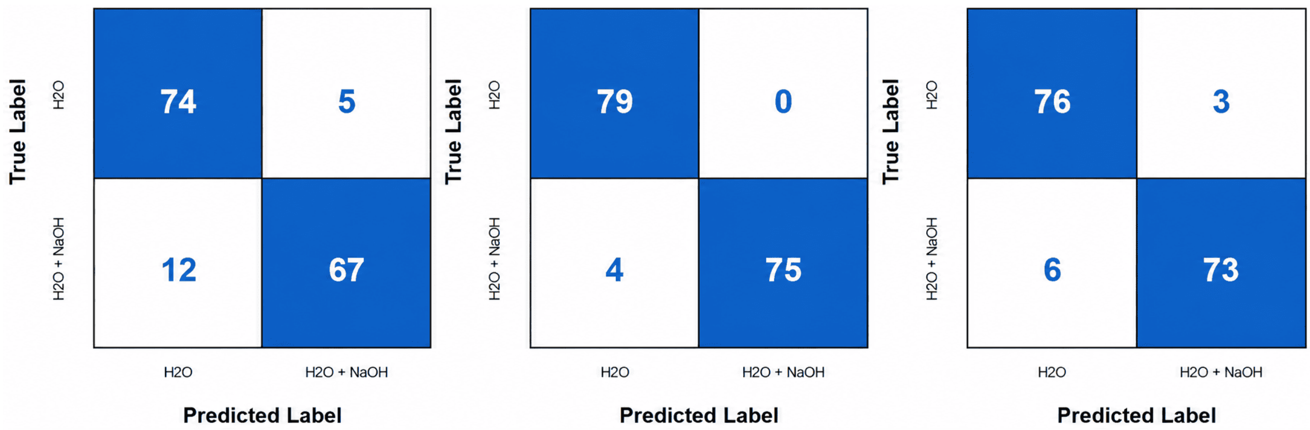
C. Best-Performing Model Selection

Based on the testing performance summarized in Table 4 and the confusion matrix analysis in Fig. 8, VGG19 was selected as the best-performing CNN architecture in this study. VGG19 achieved the highest testing accuracy of 97.47%, with a precision of 95.18%, a recall of 100.00%, and an F1-score of 97.53%. These results indicate that VGG19 provided the most balanced classification performance among the three evaluated models. Compared with InceptionV3 and

Table 4. Testing performance of the CNN models on the independent testing subset of 158 images, with H_2O treated as the positive class.

Model	Test samples	Accuracy, % (95% CI)	Precision (%)	Recall (%)	Specificity (%)	F1-score (%)	MCC
InceptionV3	158	89.24 (83.45–93.17)	86.05	93.67	84.81	89.70	0.7879
VGG19	158	97.47 (93.67–99.01)	95.18	100.00	94.94	97.53	0.9506
DenseNet201	158	94.30 (89.53–96.97)	92.68	96.20	92.41	94.41	0.8867

Manuscript received 4 May 2026; Revised 28 June 2026; Accepted 5 July 2026; Available online 21 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1727



(a) (b) (c)

Fig. 8. **Confusion matrices of the CNN models on the independent testing subset consisting of 158 images, with 79 H_2O images and 79 H_2O with NaOH images. Panels (a), (b), and (c) show the classification results of InceptionV3, VGG19, and DenseNet201, respectively. The matrices summarize correct and incorrect predictions for the two transparent liquid classes. Rows represent the true class labels, whereas columns represent the predicted class labels.**

DenseNet201, VGG19 produced fewer misclassifications and showed stronger class-wise prediction consistency. InceptionV3 achieved lower testing performance, particularly in terms of precision and F1-score, indicating less balanced prediction behavior. DenseNet201 performed better than InceptionV3 and showed good recall, but its overall accuracy and F1-score remained lower than those of VGG19.

The superior performance of VGG19 may be related to its deep and sequential convolutional structure, which is suitable for extracting localized intensity and texture features from segmented fluctuation-pattern images. However, this interpretation remains limited without direct feature attribution analysis. Future work should incorporate feature visualization methods such as Grad-CAM, saliency maps, activation maps, or feature importance analysis to provide direct evidence of which spatial regions and fluctuation-pattern features drive classification in each CNN architecture.

IV. Discussion

The experimental results demonstrate that multispectral fluctuation-pattern images contain discriminative visual information for distinguishing pure H_2O and H_2O with NaOH solutions. Although both solutions appear visually transparent under normal observation, the multispectral image representations reveal differences in intensity distribution and fluctuation patterns that can be learned by deep convolutional models. This finding supports the use of non-contact image-based analysis as an alternative approach for identifying transparent liquid samples without direct physical contact. To strengthen the comparative analysis with previous studies, Table 5 presents a structured comparison between the proposed method and recent related studies in transparent liquid-content detection, transparent chemical vessel detection, hyperspectral chemical analysis, water-pollution classification, and liquid-related computer vision recognition. The comparison includes dataset size, sensing modality, classification or analysis method, reported accuracy or main performance, advantages, and limitations. This structured comparison is intended to clarify the novelty and practical relevance of the proposed framework relative to existing non-contact sensing and image-based liquid analysis approaches.

The comparison shows that previous studies have contributed to several related areas, including transparent container detection, liquid-content estimation, hyperspectral chemical analysis, water-pollution classification, and liquid-related visual recognition. Wu et al. [3] and Tiong et al. [9] mainly focused on detecting transparent containers or chemical vessels, whereas Zhu et al. [8] and Leung et al. [6] demonstrated the usefulness of hyperspectral or spectral image representations for chemical or environmental analysis. Chen et al. [29] further showed that computer-vision-based deep learning can achieve high accuracy in liquid-related industrial recognition tasks. However, these studies did not specifically address binary classification of visually indistinguishable transparent chemical solutions such as pure H_2O and H_2O with NaOH. In contrast, the proposed method directly combines multispectral fluctuation-pattern images, K-means segmentation, and CNN-based classification for non-contact transparent liquid discrimination, achieving 97.47% accuracy using VGG19 on a balanced dataset of 1,050 images. This comparison indicates that the proposed

Manuscript received 4 May 2026; Revised 28 June 2026; Accepted 5 July 2026; Available online 21 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1727

Journal of Electronics, Electromedical Engineering, and Medical Informatics
Homepage: jeeemi.org; Vol. 8, No. 3, July 2026, pp: 1184-1203 e-ISSN: 2656-8632

Table 5. Qualitative comparison of the proposed framework with recent related studies using different sensing modalities and prediction tasks.

Study	Dataset size and sensing modality	Method and task	Main reported performance	Main advantage	Main limitation
Wu et al. [3]	5,916 RGB images of transparent containers	LCD-YOLOF/YOLOX for container and liquid-content detection	Composite mAP = 0.548	Benchmark for transparent-container analysis	No chemical-composition discrimination
Tiong et al. [9]	1,148 source vial images; 17,174 training and 2,282 testing cases using RGB machine vision	DenseSSD for transparent-vial detection	mAP = 95.2%	Robust transparent-vial detection	No liquid-composition classification
Zhu et al. [8]	187 batches; hyperspectral imaging at 389–1020 nm	SE-ReHIA for chemical-constituent regression	R_p^2 = 0.930–0.975	Non-destructive spectral-spatial analysis	Solid samples and regression task
Leung et al. [6]	2,545 RGB and hyperspectral water-pollution images	3D-CNN for water-pollution classification	HSI accuracy = 80%; RGB accuracy = 76%	Demonstrates the benefit of hyperspectral representation	Different classification task and target classes
Chen et al. [29]	1,920 computer-vision samples	DBN-AGS-FLSS for liquid-surface pointer recognition	Accuracy = 99.37%; F1-score = 99.16%	High reported recognition performance	No chemical-composition classification
Proposed method	1,050 multispectral fluctuation-pattern images of H_2O and H_2O with NaOH	K-means segmentation with InceptionV3, VGG19, and DenseNet201 for binary classification	Best accuracy = 97.47%; F1-score = 97.53% using VGG19	Direct non-contact discrimination of visually similar transparent liquids	Two classes, one NaOH concentration, and controlled conditions

Note: The reported performance values are not directly comparable because the studies differ in dataset size and composition, sensing modality, prediction task, class definition, evaluation protocol, and performance metric. The comparison is intended to provide qualitative methodological context rather than a direct performance ranking.

framework provides a targeted solution for transparent liquid classification, although broader validation across different concentrations, liquid types, and acquisition conditions is still required.

Among the three CNN architectures evaluated in this study, VGG19 achieved the best overall testing performance, with an accuracy of 97.47%, precision of 95.18%, recall of 100.00%, and F1-score of 97.53%. This result indicates that VGG19 provided the most reliable class discrimination between pure H_2O and H_2O with NaOH under the current testing conditions. The stronger performance of VGG19 may be explained by its regular stacked convolutional structure, which is well-suited for learning localized intensity and texture variations from segmented multispectral fluctuation-pattern images. Because the visual differences between pure H_2O and H_2O with NaOH are subtle and appear mainly as local fluctuation-pattern differences, a sequential convolutional hierarchy may provide more stable feature abstraction than more complex multi-branch or densely connected feature propagation mechanisms. InceptionV3 is designed to extract multi-scale features through parallel convolutional branches. However, the discriminative patterns in the present dataset may be more localized than strongly multi-scale. This may explain why InceptionV3 showed lower testing performance and greater validation fluctuation than VGG19. DenseNet201 supports feature reuse through dense inter-layer connectivity, which is useful for information propagation in deep networks. Nevertheless, for this binary transparent liquid classification task, reusing many feature maps may not necessarily improve discrimination if the most relevant class differences are concentrated in localized intensity and texture variations. Therefore, the superior performance of VGG19 can be interpreted as the result of a better match between its sequential convolutional feature extraction and the local fluctuation-pattern characteristics of the segmented multispectral images.

Although VGG19 achieved the best classification performance, practical deployment also requires consideration of computational complexity. VGG19 has

Manuscript received 4 May 2026; Revised 28 June 2026; Accepted 5 July 2026; Available online 21 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1727

a relatively simple and sequential convolutional structure that is effective for learning localized texture and intensity patterns; however, it is generally associated with a large parameter burden compared with more parameter-efficient architectures. This may increase memory consumption and training cost, particularly when the model is fine-tuned end-to-end. DenseNet201 uses dense inter-layer connectivity to support feature reuse and information propagation, but its deep structure may still increase computational burden during training and inference. InceptionV3 employs multi-branch and factorized convolutional operations that are generally designed to improve computational efficiency; however, in the present dataset, its classification performance remained lower than those of VGG19 and DenseNet201.

Therefore, selecting the most suitable CNN architecture should consider both predictive performance and deployment constraints. VGG19 may be preferred when classification accuracy is the primary objective and sufficient GPU resources are available. DenseNet201 and InceptionV3 may be considered when computational efficiency, feature reuse, or lower deployment cost becomes more important. In this study, training time, inference time, parameter count, GPU requirements, GPU memory usage, and peak memory consumption were not directly profiled; therefore, the computational comparison is limited to architectural and deployment-related considerations. Future work should include systematic profiling of training time, inference latency per image, parameter count, GPU requirements, and memory consumption to evaluate the feasibility of deploying the proposed framework in real-time or resource-constrained monitoring systems.

The preprocessing and K-means segmentation stages also played an important role in improving the quality of the input representation. Image resizing, grayscale conversion, and normalization helped standardize the input data before classification, while K-means segmentation separated the relevant solution region from the background. By emphasizing the main fluctuation-pattern area, segmentation reduced background influence and helped the CNN models focus on discriminative visual features. This supports the importance of combining conventional image processing with deep learning for transparent liquid classification tasks. The results also show that direct visual transparency does not necessarily imply the absence of discriminative image features. The addition of NaOH can affect the optical response of the solution, which is reflected in the multispectral fluctuation-pattern images. CNN models, particularly VGG19, were able to learn these subtle differences and convert them into reliable class predictions. This finding is relevant for non-contact monitoring systems, where conventional contact-based measurements, such as pH meters or chemical indicators, may be less suitable for rapid or automated applications.

Despite these promising results, several limitations should be considered. First, the dataset may contain acquisition-related bias because all images were obtained from a single MSCS acquisition campaign under a fixed experimental configuration. Although the dataset was balanced between the H_2O and H_2O with NaOH classes, the use of uniform container geometry, fixed imaging distance, controlled background, and consistent illumination may cause the CNN models to learn acquisition-specific patterns rather than fully general liquid-specific optical signatures. In addition, although data augmentation, validation monitoring, and a separate testing subset were used, potential overfitting cannot be completely excluded because all samples were derived from the same controlled acquisition setting, and no independent external validation dataset was available. Therefore, the high testing accuracy should be interpreted as performance under controlled experimental conditions, not yet as evidence of general performance across all practical environments [30], [31], [32].

Second, this study did not conduct repeated training experiments with different random seeds. Although the fixed train-validation-test split allows direct comparison among InceptionV3, VGG19, and DenseNet201 under identical conditions, it does not quantify the variability caused by random initialization, batch ordering, or stochastic optimization. Therefore, the reported results reflect a single experimental realization rather than a statistical distribution of model performance. Future studies should perform multiple runs using different random seeds and report mean performance, standard deviation, confidence intervals across repeated runs, and statistical significance tests to provide stronger validation of model comparisons [30], [31], [32].

Third, feature visualization methods were not implemented in the current study. Although the performance differences among VGG19, DenseNet201, and InceptionV3 were interpreted based on architectural characteristics, training behavior, confusion matrices, and testing metrics, this interpretation has not yet been supported by visual explainability analysis. Future studies should apply feature visualization methods such as Grad-CAM, Grad-CAM++, saliency maps, activation maps, or feature importance analysis to examine whether each CNN model focuses on the segmented liquid-region fluctuation patterns or on less relevant background and container-related regions [33], [34], [35], [36], [37]. Such analysis would provide stronger visual evidence for explaining why VGG19 outperforms DenseNet201 and InceptionV3 in transparent liquid classification.

Manuscript received 4 May 2026; Revised 28 June 2026; Accepted 5 July 2026; Available online 21 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1727

Fourth, the transferability of the proposed framework to real industrial environments remains untested. In practical monitoring systems, transparent liquid images may be affected by variations in sensor-to-sample distance, vibration, container material, sample volume, surface reflection, bubbles, impurities, temperature, background conditions, and sensor calibration. Illumination sensitivity is also important because multispectral fluctuation-pattern images are generated from intensity variations; changes in illumination intensity, spectral distribution of the light source, or ambient light interference may shift the image representation and affect CNN predictions. Therefore, future studies should include external validation under different acquisition sessions, devices, lighting conditions, container types, and environmental settings to evaluate the robustness and industrial applicability of the proposed framework [38].

Fifth, the present study only evaluated one NaOH concentration in the H_2O with NaOH class. As a result, the model has not yet been tested on concentration-dependent spectral variations or borderline samples with low NaOH concentration. Different NaOH concentrations may produce gradual changes in optical response and fluctuation-pattern intensity, which may affect the classification boundary between pure H_2O and H_2O with NaOH. Future research should expand the dataset by including multiple NaOH concentrations, additional transparent chemical solutions, and independently acquired samples to determine whether the proposed framework can distinguish not only binary liquid classes but also concentration-dependent chemical variations [6], [39].

Future work should therefore focus on expanding the dataset to include different NaOH concentrations, additional transparent chemical solutions, and independently acquired samples from different acquisition sessions. Further studies should also evaluate robustness under varied illumination conditions, container materials, sample volumes, sensor-to-sample distances, and real industrial environments. In addition, repeated experiments with multiple random seeds, systematic computational profiling of training time, inference latency, parameter count, GPU requirements, and memory consumption, as well as explainability analysis using Grad-CAM, Grad-CAM++, saliency maps, activation maps, or feature importance methods, should be explored to improve reliability, interpretability, and practical applicability.

Overall, the results confirm that CNN-based classification using multispectral fluctuation-pattern images is a feasible approach for non-contact identification of transparent liquid samples. Among the evaluated architectures, VGG19 provided the most reliable classification performance and was identified as the best-performing model for distinguishing pure H_2O and H_2O with NaOH solutions in this study.

V. Conclusion

This study aimed to develop and evaluate a non-contact multispectral image classification framework for distinguishing pure H_2O and H_2O with NaOH solutions, which are visually indistinguishable under normal observation. The proposed framework combines image preprocessing, K-means segmentation, data augmentation, and CNN-based classification using InceptionV3, VGG19, and DenseNet201. The experimental results showed that VGG19 achieved the best performance, with an accuracy of 97.47%, precision of 95.18%, recall of 100.00%, and F1-score of 97.53%. VGG19 also outperformed DenseNet201 and InceptionV3 by 3.17 and 8.23 percentage points in accuracy, respectively, and produced only 4 misclassified samples out of 158 testing images.

These findings indicate that multispectral fluctuation-pattern images combined with K-means-based region isolation and CNN feature extraction, can capture subtle optical differences between pure H_2O and H_2O with NaOH. The 97.47% accuracy is practically significant because it demonstrates the feasibility of rapid, non-contact, and image-based transparent liquid identification without direct sample contact, which may support automated monitoring in laboratory and industrial environments. Future research should expand the dataset to include multiple NaOH concentrations, additional transparent chemical solutions, and independently acquired samples. Further validation under varied illumination conditions, container materials, sample volumes, sensor-to-sample distances, and real industrial environments is also needed, along with repeated experiments, model explainability analysis, and lightweight or ensemble-based deployment strategies

Acknowledgment

The authors would like to acknowledge Universitas Syiah Kuala for providing institutional research support.

Funding

This study received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data Availability

The dataset used and analyzed in this study was derived from previously acquired multispectral measurements of pure H_2O and H_2O with NaOH solutions. The data are available from the corresponding author upon reasonable request.

Author Contribution

Manuscript received 4 May 2026; Revised 28 June 2026; Accepted 5 July 2026; Available online 21 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1727

Siti Rusdiana contributed to the conceptualization, supervision, and overall direction of the study. Asep Rusyana contributed to methodological validation and data interpretation. Juwita contributed to technical review, methodological input, and manuscript improvement. Mauliza Putri contributed to data preparation, experimental implementation, and manuscript drafting. Aufa Rafiki contributed to the implementation of the CNN model, performance evaluation, visualization, and manuscript revision. Souvik Das contributed to research supervision, technical validation, and critical review of the manuscript. All authors reviewed and approved the final version of the manuscript and agreed to be responsible for the integrity and accuracy of the work.

Declarations

Ethical Approval

This study did not involve human participants, human biological samples, or animal subjects. Therefore, ethical approval was not required.

Consent for Publication Participants.

Not applicable.

Competing Interests

The authors declare no competing interests.

References

[1] L. Lajoie, A. S. Fabiano-Tixier, and F. Chemat, “Water as Green Solvent: Methods of Solubilisation and Extraction of Natural Products—Past, Present and Future Solutions,” Pharmaceuticals, vol. 15, no. 12, Art. no. 1507, 2022, doi: 10.3390/ph15121507.

[2] C. D. Rodríguez-Fernández, L. M. Varela, C. Schröder, and E. L. Lago, “Charge delocalization and hyperpolarizability in ionic liquids,” J. Mol. Liq., vol. 349, Art. no. 118153, 2022, doi: 10.1016/j.molliq.2021.118153.

[3] Y. Wu, H. Ye, Y. Yang, Z. Wang, and S. Li, “Liquid Content Detection in Transparent Containers: A Benchmark,” Sensors, vol. 23, no. 15, Art. no. 6656, 2023, doi: 10.3390/s23156656.

[4] I. Shaw and P. Magee, “Acid–base quantification: A review of developing technology,” BJA Educ., vol. 22, no. 11, pp. 440–447, 2022, doi: 10.1016/j.bjae.2022.07.006.

[5] W. Zhang et al., “Non-contact measurement method of liquid composition using microwave radar cross-section,” Sci. Rep., vol. 14, no. 1, Art. no. 29744, 2024, doi: 10.1038/s41598-024-81043-4.

[6] J.-H. Leung et al., “Water pollution classification and detection by hyperspectral imaging,” Opt. Express, vol. 32, no. 14, pp. 23956–23965, 2024, doi: 10.1364/OE.522932.

[7] S. Kendler, Z. Mano, R. Aharoni, R. Raich, and B. Fishbain, “Hyperspectral imaging for chemicals identification: A human-inspired machine learning approach,” Sci. Rep., vol. 12, no. 1, Art. no. 17580, 2022, doi: 10.1038/s41598-022-22468-7.

[8] J. Zhu, J. Bao, and Y. Tao, “A Nondestructive Methodology for Determining Chemical Composition of Salvia miltiorrhiza via Hyperspectral Imaging Analysis and Squeeze-and-Excitation Residual Networks,” Sensors, vol. 23, no. 23, Art. no. 9345, 2023, doi: 10.3390/s23239345.

[9] L. C. O. Tiong et al., “Machine vision-based detections of transparent chemical vessels toward the safe automation of material synthesis,” npj Comput. Mater., vol. 10, no. 1, Art. no. 42, 2024, doi: 10.1038/s41524-024-01216-7.

[10] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, “A review of convolutional neural networks in computer vision,” Artif. Intell. Rev., vol. 57, no. 4, Art. no. 99, 2024, doi: 10.1007/s10462-024-10721-6.

[11] Z. Khan and J. Yang, “Nonparametric K-means clustering-based adaptive unsupervised colour image segmentation,” Pattern Anal. Appl., vol. 27, no. 1, Art. no. 17, 2024, doi: 10.1007/s10044-024-01228-5.

[12] M. Sabha and M. Saffarini, “Selecting optimal k for K-means in image segmentation using GLCM,” Multimed. Tools Appl., vol. 83, no. 18, pp. 55587–55603, 2024, doi: 10.1007/s11042-023-17615-9.

[13] H. Mittal, A. C. Pandey, M. Saraswat, S. Kumar, R. Pal, and G. Modwel, “A comprehensive survey of image segmentation: Clustering methods, performance parameters, and benchmark datasets,” Multimed. Tools Appl., vol. 81, no. 24, pp. 35001–35026, 2022, doi: 10.1007/s11042-021-10594-9.

[14] M. Krichen, “Convolutional Neural Networks: A Survey,” Computers, vol. 12, no. 8, Art. no. 151, 2023, doi: 10.3390/computers12080151.

[15] M. M. Taye, “Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions,” Computation, vol. 11, no. 3, Art. no. 52, 2023, doi: 10.3390/computation11030052.

[16] J. Ran, G. Dong, F. Yi, L. Li, and Y. Wu, “Automatic Measurement of Comprehensive Skin Types Based on Image Processing and Deep Learning,”

Manuscript received 4 May 2026; Revised 28 June 2026; Accepted 5 July 2026; Available online 21 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1727

Electronics, vol. 14, no. 1, Art. no. 49, 2025, doi: 10.3390/electronics14010049.
[17] I. A. Kandhro et al., "Performance evaluation of E-VGG19 model: Enhancing real-time skin cancer detection and classification," Heliyon, vol. 10, no. 10, Art. no. e31488, 2024, doi: 10.1016/j.heliyon.2024.e31488.
[18] M. Bundea and G. M. Danciu, "Pneumonia Image Classification Using DenseNet Architecture," Information, vol. 15, no. 10, Art. no. 611, 2024, doi: 10.3390/info15100611.
[19] B. Dey, J. Ferdous, R. Ahmed, and J. Hossain, "Assessing deep convolutional neural network models and their comparative performance for automated medicinal plant identification from leaf images," Heliyon, vol. 10, no. 1, Art. no. e23655, 2024, doi: 10.1016/j.heliyon.2023.e23655.
[20] H. Zhou, X. Wang, K. Xia, Y. Ma, and G. Yuan, "Transfer Learning-Based Hyperspectral Image Classification Using Residual Dense Connection Networks," Sensors, vol. 24, no. 9, Art. no. 2664, 2024, doi: 10.3390/s24092664.
[21] R. Hou, J. Y. Lo, J. R. Marks, E. S. Hwang, and L. J. Grimm, "Classification performance bias between training and test sets in a limited mammography dataset," PLOS ONE, vol. 19, no. 2, Art. no. e0282402, 2024, doi: 10.1371/journal.pone.0282402.
[22] Z. Yang, R. O. Sinnott, J. Bailey, and Q. Ke, "A survey of automated data augmentation algorithms for deep learning-based image classification tasks," Knowl. Inf. Syst., vol. 65, no. 7, pp. 2805–2861, 2023, doi: 10.1007/s10115-023-01853-2.
[23] S. Seoni et al., "All you need is data preparation: A systematic review of image harmonization techniques in multi-center/device studies for medical support systems," Comput. Methods Programs Biomed., vol. 250, Art. no. 108200, 2024, doi: 10.1016/j.cmpb.2024.108200.
[24] F. Hu et al., "Image harmonization: A review of statistical and deep learning methods for removing batch effects and evaluation metrics for effective harmonization," NeuroImage, vol. 274, Art. no. 120125, 2023, doi: 10.1016/j.neuroimage.2023.120125.
[25] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, "K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data," Inf. Sci., vol. 622, pp. 178–210, 2023, doi: 10.1016/j.ins.2022.11.139.
[26] X. Chai, Z. Wu, W. Li, H. Fan, X. Sun, and J. Xu, "Image Segmentation Based on the Optimized K-Means Algorithm with the Improved Hybrid Grey Wolf Optimization: Application in Ore Particle Size Detection," Sensors, vol. 25, no. 9, Art. no. 2785, 2025, doi: 10.3390/s25092785.
[27] H. T. Lee, H. R. Cheon, S. H. Lee, M. Shim, and H. J. Hwang, "Risk of data leakage in estimating the diagnostic performance of a deep-learning-based computer-aided system for psychiatric disorders," Sci. Rep., vol. 13, no. 1, Art. no. 16633, 2023, doi: 10.1038/s41598-023-43542-8.
[28] O. Rainio, J. Teuho, and R. Klén, "Evaluation metrics and statistical tests for machine learning," Sci. Rep., vol. 14, no. 1, Art. no. 6086, 2024, doi: 10.1038/s41598-024-56706-x.
[29] L. Chen et al., "An Adaptive Parameter Optimization Deep Learning Model for Energetic Liquid Vision Recognition Based on Feedback Mechanism," Sensors, vol. 24, no. 20, Art. no. 6733, 2024, doi: 10.3390/s24206733.
[30] M. A. Lones, "Avoiding common machine learning pitfalls," Patterns, vol. 5, no. 10, Art. no. 101046, 2024, doi: 10.1016/j.patter.2024.101046.
[31] H. Feng, Y. Wang, Z. Li, N. Zhang, Y. Zhang, and Y. Gao, "Information Leakage in Deep Learning-Based Hyperspectral Image Classification: A Survey," Remote Sens., vol. 15, no. 15, Art. no. 3793, 2023, doi: 10.3390/rs15153793.
[32] G. Gallitto et al., "External validation of machine learning models—registered models and adaptive sample splitting," GigaScience, vol. 14, Art. no. giaf036, 2025, doi: 10.1093/gigascience/giaf036.
[33] S. S. Band et al., "Application of explainable artificial intelligence in medical health: A systematic review of interpretability methods," Inform. Med. Unlocked, vol. 40, Art. no. 101286, 2023, doi: 10.1016/j.imu.2023.101286.
[34] W. Chmiel, J. Kwiecień, and K. Motyka, "Saliency Map and Deep Learning in Binary Classification of Brain Tumours," Sensors, vol. 23, no. 9, Art. no. 4543, 2023, doi: 10.3390/s23094543.
[35] J. Sigut, F. Fumero, J. Estévez, S. Alayón, and T. Díaz-Alemán, "In-Depth Evaluation of Saliency Maps for Interpreting Convolutional Neural Network Decisions in the Diagnosis of Glaucoma Based on Fundus Imaging," Sensors, vol. 24, no. 1, Art. no. 239, 2024, doi: 10.3390/s24010239.
[36] Y. Gao, J. Liu, W. Li, M. Hou, Y. Li, and H. Zhao, "Augmented Grad-CAM++: Super-Resolution Saliency Maps for Visual Interpretation of Deep Neural Network," Electronics, vol. 12, no. 23, Art.

Manuscript received 4 May 2026; Revised 28 June 2026; Accepted 5 July 2026; Available online 21 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1727

no. 4846, 2023, doi: 10.3390/electronics12234846.

[37] K. Venkatesh, S. Mutasa, F. Moore, J. Sulam, and P. H. Yi, “Gradient-Based Saliency Maps Are Not Trustworthy Visual Explanations of Automated AI Musculoskeletal Diagnoses,” Journal of Imaging Informatics in Medicine, vol. 37, no. 5, pp. 2490–2499, 2024, doi: 10.1007/s10278-024-01136-4.

[38] A. Baumann et al., “Neural illumination calibration for surgical workflow-optimized spectral imaging,” International Journal of Computer Assisted Radiology and Surgery, vol. 21, no. 4, pp. 665–675, 2026, doi: 10.1007/s11548-025-03525-8.

[39] S. Nie et al., “Hyperspectral imaging combined with deep learning models for the prediction of geographical origin and fungal contamination in millet,” Front. Sustain. Food Syst., vol. 8, Art. no. 1454020, 2024, doi: 10.3389/fsufs.2024.1454020.

Author Biography

Siti Rusdiana was born in Banda Aceh, Indonesia, on 10 September 1963. She received the B.S. degree in mathematics science from the Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia, in 1988, the M.Eng. degree in mathematics science from Osaka University, Osaka, Japan, in 1998, and the Ph.D. degree in applied mathematics from Universitas Sumatera Utara, Medan, Indonesia, in 2013. She was the Chairperson of the Department of Mathematics, Universitas Syiah Kuala, Banda Aceh, Indonesia (1998 – 2004). From 2013 to 2020, she was the Head of the Laboratory for Dynamic and Optimization Applications in the Department of Mathematics at Universitas Syiah Kuala, where she has been an Associate Professor with the Department of Mathematics since 2014. She is also a member of IEEE. Her research interests in applied mathematics include optimization, operations research, and data science. She can be contacted at email: siti.rusdiana@usk.ac.id.

Asep Rusyana graduated from the Department of Statistics, IPB University, in 2000 with a Bachelor of Science. Then he continued to the same department and graduated in 2005 with a Master of Science. Then the author continued studying at the Doctoral Program in Study Program of Statistics and Data Science from 2019 until 2024. He became a lecturer at the Department of Mathematics, Universitas Syiah Kuala (USK), Banda Aceh, from 2006 to 2012. He has been a lecturer at the Department of Statistics, USK, since 2013. During his time as a USK lecturer, the author serve as head of the computational statistics laboratory, chair of the Mathematics Department from January 2012 to December 2015, and staff at the USK quality assurance agency between 2008 and 2018. The author has been a member of IEEE since January 2023. While teaching at USK, the author teaches several courses including machine learning, computational statistics, algorithm and programming, experimental design, matrix algebra, multivariate analysis, introductory statistics, operations research, time series analysis, and calculus. More than 50 publications in applied statistics and data sciences have been published in journals or at conferences at national and international levels. The author's areas of research interest are data science and applied statistics.

Juwita is a senior lecturer in the Department of Informatics at Universitas Syiah Kuala, Indonesia. She earned her Ph.D. in Computer Science and Software Engineering from the University of Western Australia in 2025. Her recent research has primarily focused on medical image segmentation. In addition, her research interests encompass broader multidisciplinary domains, particularly in the development of information systems aimed at enhancing human quality of life, including applications in education and healthcare. juwita@usk.ac.id.

Mauliza Putri was born on December 27, 2002, in Pulau Tiga. She is a student majoring in Electrical and Computer Engineering at Syiah Kuala University. Her undergraduate studies focused on multimedia technology, and her research explored the Comparative Performance Analysis of H_2O and H_2O–NaOH Mixtures Based on K-Means Segmentation, CNNs, and Decision Fusion Classification. She actively participates in class and continues to develop her knowledge in her field. Enrolled in the class of 2022, she is committed to expanding her expertise and gaining practical experience. Her academic journey reflects her dedication to both theoretical and applied learning, preparing her to contribute to technological advancement in her field. She can be reached at mauliza6@mhs.usk.ac.id.

Aufa Rafiki was born on April 20, 2003, in Banda Aceh, Indonesia. He received his Bachelor’s degree in Computer Engineering from Universitas Syiah Kuala in 2025 and graduated as the best graduate of the Faculty of Engineering.

Manuscript received 4 May 2026; Revised 28 June 2026; Accepted 5 July 2026; Available online 21 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1727

He also completed a fast-track Master's degree in Electrical Engineering with a biomedical engineering concentration at Universitas Syiah Kuala. His research interests include EEG signal processing, artificial intelligence, deep learning, biomedical signal processing, and early detection of autism spectrum disorder using medical technology. He has experience in collaborative research, scientific publication, AI-based system development, and intellectual property outputs. He has also served as a research assistant and laboratory assistant. He can be contacted at aufa35@mhs.usk.ac.id.

Souvik Das completed his B.Tech. in Electrical Engineering from RCC Institute of Information Technology (RCCIIT), West Bengal, India, in 2014. He then obtained his M.Tech. in Industrial Engineering and Management from the Indian Institute of Technology (IIT) Kharagpur in 2017. He earned his Ph.D. in Industrial and Systems Engineering from IIT Kharagpur in 2023, where his research focused on data-driven modeling of cognitive workload using eye-tracking metrics. During his doctoral studies, he worked as a Principal Research Scientist at the Centre of Excellence in Safety Engineering and Analytics (CoE-SEA), IIT Kharagpur. He later served as a Postdoctoral Research Fellow (Visiting Scientist) at Purdue University, USA, working on AI-based analysis of accident and incident narratives for safety analytics. Since August 2024, he has been serving as an Assistant Professor in the Department of Industrial Design at the National Institute of Technology (NIT) Rourkela, India. His research interests include human factors, safety analytics, eye-tracking technology, virtual reality, and artificial intelligence for safety and design applications. He can be contacted at dass@nitrkl.ac.in.

Manuscript received 4 May 2026; Revised 28 June 2026; Accepted 5 July 2026; Available online 21 July 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i3.1727