Journal of Soft Computing and Computer Applications

Manuscript 1031

TinyVGG-Based Real-Time Degradation Classification for Adverse Driving Scenes Using a Newly Collected Iraqi Driving Dataset

Yousif N. Abbas

Matheel E. Abdulmunim

Nada H. Ali

Ismail A. Mageed

ORIGINAL STUDY

TinyVGG-Based Real-Time Degradation Classification for Adverse Driving Scenes Using a Newly Collected Iraqi Driving Dataset

Yousif N. Abbas [a], **Matheel E. Abdulmunim** [b], **Nada H. Ali** [b], **Ismail A. Mageed** [c,*]

[a] University of Technology-Iraq, College of Computer Science, Al-Sina'a St., Al-Wehda District, 10066 Baghdad, Iraq
[b] University of Technology-Iraq, College of Computer Science, Department of Multimedia and Digital Media, Al-Sina'a St., Al-Wehda District, 10066 Baghdad, Iraq
[c] University of Derby-United Kingdom, Institute of Education and Skills, Kedleston Road, Derby, DE22 1GB, United Kingdom.

ABSTRACT

Environmental conditions such as low-light at night, fog scattering, glare artifacts, rain streaks, and rain smear distortions are significant issues of camera-based perception in Autonomous Vehicles (AVs). These degradations alter the statistics of the scene, mask structure, introduce non-uniform noise, and adversely affect downstream vision processes, including detection and tracking. To overcome this shortcoming, this paper presents a lightweight TinyVGG-based degradation classification system that runs in real time. The network extracts discriminative spatial features with hierarchical convolutional encoding and projects them to a lower-dimensional semantic representation with fully connected layers and a multi-class predictor based on SoftMax. In addition, a confidence estimation mechanism is incorporated to assess the reliability of each prediction, providing an additional level of robustness and interpretability, particularly in safety-critical driving scenarios. A custom real-world driving dataset was built and annotated into five degradation categories. The evaluation shows high performance, with an overall accuracy of 98.17%, a macro F1-score of 98.20%, and an ROC-AUC of 0.9978, along with low computational overhead and real-time inference. These results support the usefulness and feasibility of the proposed framework in embedded automotive systems with strict latency and resource limitations.

Keywords: Degradation classification, TinyVGG architecture, Adverse weather conditions, Real-time inference, Lightweight convolutional neural networks

Received 6 March 2026; revised 5 June 2026; accepted 5 June 2026.
Available online 16 June 2026

* Corresponding author.
E-mail addresses: cs.24.27@grad.uotechnology.edu.iq (Y. N. Abbas), matheel.e.abdulmunim@uotechnology.edu.iq (M. E. Abdulmunim), nada.h.ali@uotechnology.edu.iq (N. H. Ali), 100800315@unimail.derby.ac.uk (I. A. Mageed).

https://doi.org/10.70403/3008-1084.1031

1. Introduction

Camera-based perception systems are crucial for the interpretational performance of Autonomous Vehicles (AVs), allowing them to perform tasks such as object detection, lane recognition, scene comprehension, and tracking. Thereby, reliable visual awareness is very crucial for safe and reliable autonomous driving in diverse scenarios like the actual real world. Deep learning, on the other hand, improves the visual perception systems' efficiency, but the perceptual accuracy of these systems is also related to the quality and clarity of the visual data obtained [1]. In real-world driving scenes, camera images have been affected by many adverse visual conditions (AVCs) including low illumination at night, fog scattering, glare artifacts, rain streaks, and rain smear effects [2, 3]. Such unpleasant situations significantly modify the scene view, hide structural information, and intensify multi-faceted noise, which can be detrimental to downstream perception tasks, including object detection, lane recognition, and tracking. Although a lot of work has concentrated on improving degradation before detection, relatively little attention has been paid to the explicit identification of the nature of environmental degradation. However, degradation classification is a critical pre-processing step for adaptive perception pipelines and the different distortions have different spatial and statistical characteristics, therefore degrading correctly. This helps to spot such patterns used in intelligent decision-making at the same time as increasing adaptability of autonomous systems working in different weather and lighting conditions [4].

Current studies provide some knowledge on adverse visibility in driving environments. First, most approaches treat degradation categories separately, such as rain, fog, and low light. While practical driving scenarios often have multiple visually overlapping degradation factors. This specific treatment restricts the generalization capacity of the available models to various real-life situations. Secondly, most current methods focus on image restoration or enhancement performance without specifying the main type of degradation before downstream processing. However, different degradation categories may require distinct enhancement or perception approaches; therefore, awareness of degradation is a crucial step for adaptive driving systems. Third, few studies focus on the reliability of prediction and recognizing degradation type, where current models are typically used to predict classes but do not estimate confidence/uncertainty of the prediction. Unreliable predictions in ambiguous scenes may adversely affect decision-making in safety-critical scenarios, such as those in intelligent transportation systems. Lastly, realistic driving datasets that include a variety of adverse visual conditions are still limited. Current datasets may not be sufficiently diverse in terms of environments or scene complexity, potentially compromising the robustness and generalization of models in real-world scenarios. These restrictions emphasize the importance of lightweight and reliable degradation recognition structures that can be deployed in real-world driving conditions, with degradation identification and prediction confidence information.

The main goal of this paper is to develop a lightweight framework for classifying degradation in camera-based driving images. The proposed work is intended to serve as a first degradation-awareness module for intelligent camera-based driving systems. The detection of the dominant degradation type can be beneficial for subsequent processing, such as adaptive image enhancement, degradation-aware perception, and automotive vision applications, where different processing strategies can be selected based on the detected environmental condition. This paper also examines the importance of prediction confidence score in the task of Adverse-Scene Recognition (ASR). Confidence score estimation provides a predicted degradation category and indicates the model's confidence in its output. The predicted degradation label and confidence score could be helpful for

downstream decisions, especially in environments that are unclear or highly degraded. The other goal is to present a real-world driving dataset with various complex environmental conditions. The dataset was manually labeled with categorical degradation labels and supplemented with "enhanced" reference images, which were kept for future studies using adaptive enhancement methods.

This paper makes the following main contributions:

1. We created and labeled a custom real-world driving dataset to help fill the gap in datasets that cover a wide range of challenging visual conditions. Our dataset features five types of visual degradation: rain streaks, rain smear, fog, night, and nighttime glare.
2. A lightweight real-time degradation classification framework is proposed to be ready for deployment in camera-based driving systems and operating under adverse environmental conditions.
3. The used Convolutional Neural Network (CNN) architecture is developed to efficiently learn discriminative spatial features for multi-class degradation classification while maintaining low computational complexity.
4. A confidence-aware prediction strategy was included to estimate prediction reliability and support robust decision-making in visually ambiguous driving scenarios.

This paper is organized as follows. Section 2 reviews related work on CNN-based classification techniques. Section 3 explains the proposed confidence-aware TinyVGG-based framework for degradation classification. Section 4 covers dataset preparation, experimental evaluation, and quantitative results. Section 5 presents the conclusions and suggests directions for future research.

2. Related Works

Recent works have also examined lightweight convolutional neural networks. Cordts et al. describe the Cityscapes dataset (2016), a large-scale benchmark to examine the semantics of urban scenes. It comprises high-resolution images of driving scenes from several European cities, which have fine-grained pixel-level semantic annotations. Urban clear-weather environments are covered by this set of images. It gives detailed object labeling for vehicles, pedestrians, roads, and traffic infrastructure. Thanks to its quality annotations and standardized procedures for assessment [5], it was first developed for such studies.

Sakaridis et al. (2021) proposed the Adverse Conditions Dataset with Correspondences (ACDC) to resolve semantic scene understanding in adverse weather. The data consists of real-life driving scenes from conditions of various weather, i.e., fog, rain, snow, and night, accompanied by pixel-based semantic annotations of high quality [6].

Fei et al. in 2024 introduced the Lightweight-Visual Geometry Group (VGG), an efficient deep learning-based framework capable of dealing with hyperspectral images. The proposed model is also composed of feature dimension-reduction and nonlinear feature-enhancement techniques in order to enhance the classification performance with a small computational cost. The architecture is faster and simpler than deep CNNs that are typically used for representing features and is faster for inference, because it simplifies the typical VGG architecture, while also improving feature representation. They demonstrated that lightweight VGG-based architectures are effective, albeit with limited resources, in classification tasks [7] while addressing highly specific hyperspectral images rather than driving scenes here.

Saini et al. developed an efficient TinyVGG-based model for classification of sugarcane leaf diseases for the purpose of precision agriculture study. They proved that a VGG-like convolutional architecture can be successfully implemented for image classification tasks with a lightweight structure for realization. They conducted the detection of patterns in the images of sugarcane leaves for disease recognition and then showed that TinyVGG is an effective tool for extracting discriminative visual features from the images in the field of the dataset. While their results were performed to classify agricultural diseases more than identify the degradation recognition of driving scenes, that is a favourable development for the potential applicability of TinyVGG-based architectures in classification of visual images with low computational needs [8].

López et al. introduced WeldVGG, a deep learning model inspired by VGG, for weld defect classification in X-ray images. The model was developed for automated visual inspection of radiographic weld images, providing defect classification across different categories and introducing visual interpretability. They found that VGG-based architectures still perform well for image classification problems specialized to a particular application area when tailored to that domain. Their work focuses on the classification of weld defects in industrial images, not on image degradation in driving situations, but the results are relevant to the use of compact VGG-inspired models for learning discriminative visual patterns in domain-specific images [9].

Imani and Cerra suggested a triple graph convolutional network for feature fusion and classification of hyperspectral images. The technique they employ is a graph-based method that enhances classification accuracy by introducing relationships between hyperspectral image features. While the study is centered on hyperspectral remote sensing classification rather than adverse driving-scene degradation, it demonstrates the current practice of creating lightweight or domain-specific deep learning models by leveraging specialized tasks and features [10].

In 2025, Zhang introduced a lightweight CNN framework for efficient image classification, particularly for edge computing environments. We build on knowledge distillation and structured pruning methods to simplify the complexity of this type of machine learning model while preserving classification performance. The solution can provide better computational efficiency with only a slight performance loss by transferring knowledge from a larger teacher network to a smaller student model and discarding unnecessary parameters. While the framework focuses on general image classification, it also reflects recent development of efficient CNN architectures for deploying in real-time on low powered hardware [11].

A general overview of the classifications already in use can be found in Table 1, which features distinctions in task domain, environmental degradation awareness, and the presence of confidence estimation techniques.

3. The Proposed Framework

The proposed framework is realistic for real-time degradation detection, even though it suffers from computational constraints common in industry embedded systems like automobiles. For the low number of parameters and inference speed, lightweight network architecture named “TinyVGG” is applied instead of the backbone network commonly employed in large-scale networks. The input data for this system is RGB frames from road-based testing scenarios. All the input images are resized to a fixed size (256 $\times$ 256 pixels) to ensure dimensional uniformity among the samples. Normality of the images is done considering channel-wise mean and standard deviation values of (0.5, 0.5, 0.5) to

Table 1. General comparison of existing classification approaches.

Ref.	Task Focus	Degradation Awareness	Confidence Estimation	Real-Time Suitability	Notes
[5]	Urban scene understanding	Limited	N/A	N/A	Clear-weather urban driving benchmark
[6]	Adverse-condition scene understanding	Yes	N/A	N/A	Dataset for adverse driving scenes
[7]	Hyperspectral classification	Limited	N/A	Moderate	Domain-specific
[8]	Leaf disease classification	Limited	N/A	High	TinyVGG-based model for precision agriculture
[9]	WeldVGG defect classification	N/A	N/A	Moderate	VGG-inspired model with visual interpretability
[10]	Hyperspectral classification	Limited	N/A	Moderate	Graph-based feature fusion and classification
[11]	Lightweight classification	Limited	N/A	High	Efficiency-oriented
Proposed Method	Degradation classification	Yes	Yes	Moderate	Limited to predefined classes; does not handle mixed degradations explicitly

stabilize training and fast convergence to the optimum. The resulting normalized image can be transmitted to the TinyVGG feature-extraction network after preprocessing. The backbone is composed of consecutive 3 × 3 convolution layers coupled with ReLU activations, and subsequent max-pooling. The layers are trained on the network to derive better and better spatial features depending on environmental degradation factors such as rain streaks, fog effects, low-light, and night-time glare. We propose the dual-branch algorithm using confidence scores extending from the original TinyVGG single-output framework. The extracted feature representation is shared in two parallel output branches. In the first branch, the SoftMax classification head predicts the dominant degradation class for 5 degradation cases: rain smear, rain, fog, night, and nighttime glare. Branch 2: The second branch is based on a Sigmoid-based confidence estimation head that generates a scalar confidence score for augmenting the network's class prediction. This extra sub-branch enables the model to classify degradation types and assess prediction reliability.

The importance of confidence estimation is further highlighted in safety-critical driving scenarios, such as when adverse visual conditions occur simultaneously or when visual features are ambiguous. Confidence in predictions can be explicitly captured in the model, thereby facilitating the identification of cases of uncertainty and providing the basis for adaptive downstream (predictive) mechanisms such as conservative decision policies, alternative processing paths, or fallback perception mechanisms. As a result, the proposed architecture extends the concept of conventional degradation classification to confidence-aware recognition for use in real driving situations.

The general architecture of the proposed TinyVGG-based degradation classification framework is provided in Fig. 1.

3.1. Problem Formulation

Let $x \in \mathbb{R}^{H\times W\times 3}$ denote an input RGB frame captured under adverse environmental conditions. The objective of the proposed model is to learn a mapping function.

$$f_\theta : x \to p \tag{1}$$

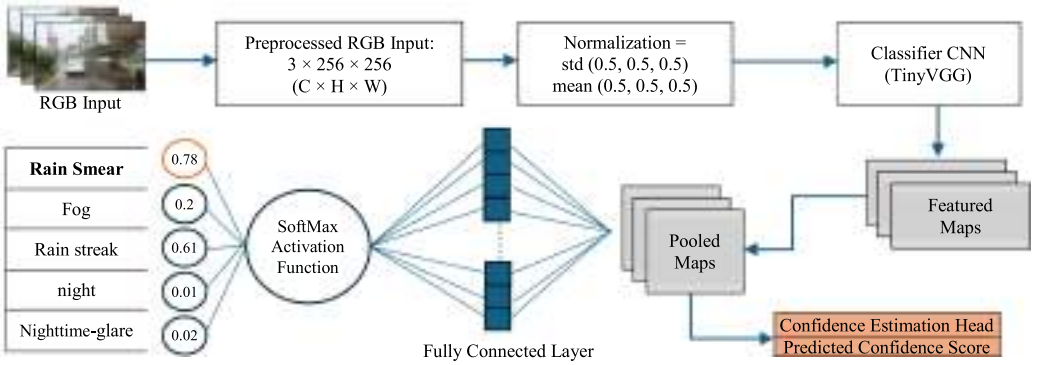

Fig. 1. Overview of the proposed confidence-aware TinyVGG-based degradation classification framework.

where $P \in \mathbb{R}^C$ represents the probability distribution over $C = 5$ predefined degradation classes, and θdenotes the trainable parameters of the TinyVGG network.

The classification decision is obtained using the SoftMax function [12]:

$$P\left(y = i|x\right) = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \tag{2}$$

where z_i represents the logit corresponding to class i, and y is the binary confidence target of the confidence branch. The value of y is assigned to be 1 if the label predicted is the same as the ground truth label, and 0 if it is not.

The network is trained by minimizing the categorical cross-entropy loss [13]:

$$\mathcal{L}_{\mathcal{CE}} = -\sum_{i=1}^{C} y_i \log\left(P\left(y = i\right)\right) \tag{3}$$

where y_i is the ground-truth label in one-hot encoding format.

The predicted confidence score c is computed as follows [14]:

$$c = \sigma\left(W_{\text{conf}}\ f\left(x\right)\right) \tag{4}$$

where c$\in$ [0, 1], $\sigma(\cdot)$ represents the Sigmoid activation function used to scale the confidence score between 0 and 1, W_{conf} are the trainable weights parameters that are learned in the confidence estimation branch, and *f(x)* is the extracted feature vector from the input image x that is extracted by the TinyVGG backbone.

The loss of confidence associated with the confidence branch $\mathcal{L}_{\text{cn}}$ favors the generation of higher confidence in case of a right prediction and lower confidence in case of a wrong prediction by the confidence branch, and is computed using the following formula [15]:

$$\mathcal{L}_{\text{cn}} = -\sum\left[y \log c + \left(1 - y\right) \log\left(1 - c\right)\right] \tag{5}$$

3.2. TinyVGG-Based Feature Extraction Backbone

The network consists of three successive convolutional blocks. Every block has two 2D convolutional layers with a 3 × 3 kernel, a stride of 1, and 1x padding to maintain the spatial resolution within the block. The convolutional layers are preceded by one ReLU

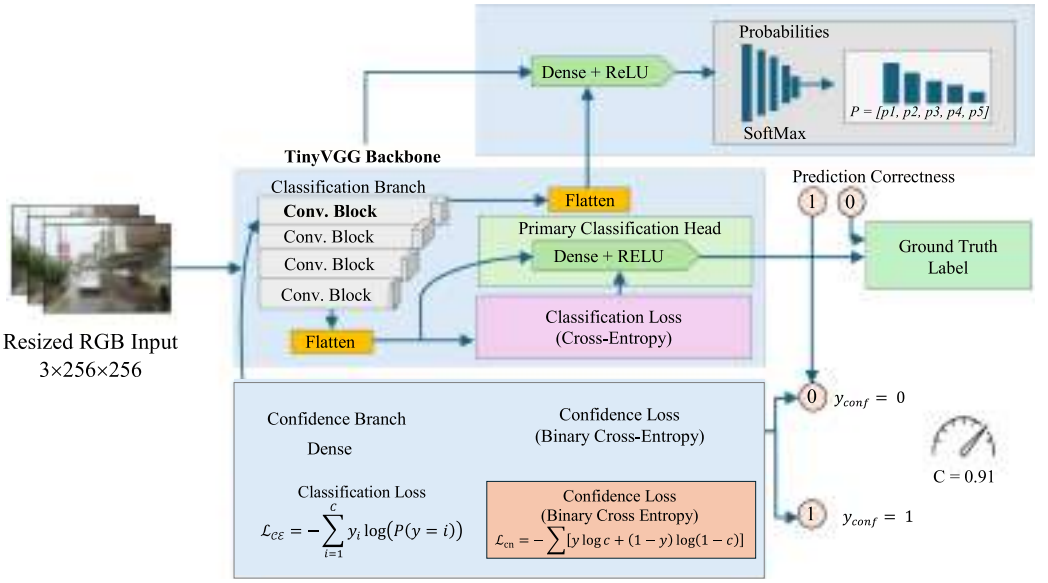

Fig. 2. Confidence-aware TinyVGG architecture with dual-branch classification and confidence estimation.

activation that provides non-linearity and better feature discrimination. Each block is followed by a max-pooling layer, which successively reduces the spatial dimensions and enables progressively more hierarchical feature abstraction. In this systematic architecture, the network learns the degradation-specific spatial connections, e.g., texture distortions, light change, blur patterns, and noise distributions. Their orderly decrease in spatial resolution enables the model to encode local and mid-level contextual characteristics of degradation without incurring unnecessary parameter increases. From the convolutional feature extraction stage, the feature maps are flattened and input to a small fully connected classification head. This layer is dense and has 64 neurons, which is to be supplemented with ReLU activation and dropout in order to alleviate overfitting and ensure better generalization. The final layer employs SoftMax activation to obtain normalized probability scores for five predefined degradation categories. Although the above architecture does effectively classify degradation, it does not quantitatively measure the validity of predictions. Hence, a new, confidence-aware extension is proposed to overcome this limitation.

3.3. Confidence-Aware Dual-Branch Architecture

The proposed framework is intended to improve the prediction accuracy and explainability in severe driving scenarios by employing a lightweight, confidence-aware dual-branch design inspired by the original TinyVGG architecture [16]. The learned feature representation f(x) is then flattened and forwarded to two parallel branches: a classification branch and a confidence-estimation branch. Fig. 2 displays the overall architecture of the proposed architecture.

The classification branch uses a SoftMax activation function to produce probability distributions over the specified degradation classes (rain smear, rain, fog, night, and night glare). The confidence estimation branch also predicts a scalar confidence score $c \in [0,1]$ using a Sigmoid activation function defined by Eq. (4).

The confidence branch is trained to predict the probability that the predicted class matches the ground-truth class label. A binary cross-entropy loss is used for optimizing

Table 2. Architectural configuration of the proposed TinyVGG-based classification model with confidence estimation.

Block	Layers Description	Output Size	Parameters
Block 1	$2 \times$ Conv(3×3, 64) + MaxPool	$128 \times 128 \times 64$	38,720
Block 2	$2 \times$ Conv(3×3, 64) + MaxPool	$64 \times 64 \times 64$	73,856
Block 3	$2 \times$ Conv(3×3, 64) + MaxPool	$32 \times 32 \times 64$	73,856
FC Head	Flatten $\rightarrow$ FC(64) $\rightarrow$ Dropout $\rightarrow$ FC(5)	5	4,194,693
Conf Head	Flatten $\rightarrow$ FC(32) $\rightarrow$ Sigmoid	1	2,081

Table 3. Computational resources summary of the proposed TinyVGG based classification model.

Metric	Value
Total Parameters	4.38 M
Trainable Parameters	4.38 M
Estimated Model Size	16.71 MB
Total Memory Footprint	206.96 MB

confidence estimation. In this architectural extension, the network can collaborate on degradation classification and reliability estimation of predictions.

Confidence estimation is especially significant in adverse driving conditions, where degradation patterns overlap or share similar visual features. For instance, categories of degradation, such as rain streaks and rain smear, can share similar visual features, leading to confusion during classification. In inference, confidence scores can be used to mark the passages they are unsure about by setting a threshold τ. Prediction with confidence intervals below τ can either prompt further processing or a different decision-making approach.

Table 2 provides an overview of the proposed TinyVGG-based architecture, including details of the convolutional blocks, fully connected layers, and the distribution of trainable parameters.

Table 3 shows the overall computational resources of the proposed classification model based on TinyVGG. Total number of parameters in the model: 4.38M, all of which are trainable. Estimated Model Size: 16.71 MB; Memory Footprint: 206.96 MB. The results show that the proposed architecture is lightweight and can be used for real-time degradation classification.

The dual-branch architecture converts TinyVGG from a traditional single-output classifier into a confidence-aware degradation recognition system that can facilitate more confident decision-making for real-world driving tasks.

3.4. Training Strategy and Loss Function

The Adam optimizer [17] was used to optimize the TinyVGG classifier due to its adaptive learning rate mechanism, which facilitates stable, efficient convergence during training. The initial learning rate was fixed at $1e^{-3}$, while the remaining optimizer parameters were maintained at their default settings.

For degradation classification, a categorical cross-entropy loss was used to optimize the primary classification branch.

To ensure clarity and reproducibility, the training workflow is outlined in Algorithm 1.

Fig. 3 presents the training and validation loss and accuracy curves across 100 epochs. In the early epochs, the loss drops quickly and the classification accuracy steadily improves,

ALGORITHM 1: Training Procedure of the TinyVGG Classifier with Confidence Estimation

Input: Training dataset D, batch size B, learning rate η, number of epochs E
Output: Trained model parameters θ, predicted class $\hat{Y}$, confidence score c

Initialize model parameters θ
for epoch = 1 to E do
  Shuffle training dataset D
  Split D into mini-batches of size B
  for each batch (X, Y) do
    Forward pass: $(\hat{Y}, c) = Model(X; \theta)$
    Compute classification loss: $L\_cls = CrossEntropy(\hat{Y}, Y)$
    Compute confidence loss: $\mathcal{L}_{cn} = -\sum[y \log c + (1-y)\log(1-c)]$
    Compute total loss: $L = L\_cls + L\_conf$
    Backward pass: compute gradients $\nabla\theta\, L$
    Update parameters: $\theta = \theta - \eta * \nabla\theta\, L$
  end for
end for
Return trained model θ

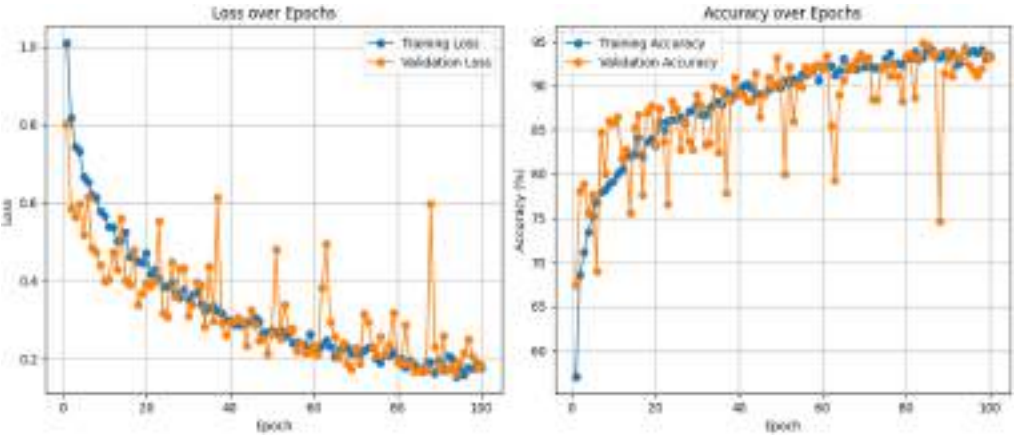

Fig. 3. Loss and accuracy curves of the TinyVGG classifier during training and validation.

which shows that the model is learning useful features. The small difference between training and validation results suggests the model generalizes well and there is little sign of overfitting.

3.5. Computational Efficiency Analysis

The TinyVGG architecture aims to balance strong feature representation with lower computational cost. Rather than using deep backbone networks like ResNet [18] or DenseNet [19], which require more resources, this framework uses lightweight convolutional operations and gradually reduces spatial dimensions. The computational complexity of the convolutional layer can be expressed as:

$$\mathcal{O}\left(H \times W \times K^2 \times C_{\text{in}} \times C_{\text{out}}\right) \tag{6}$$

Fig. 4. The AS03 camera utilized in dataset collection.

where H and W denote the height and width of the feature map, respectively, K is the convolution kernel size, and C_{in} and C_{out} represent the input and output channel dimensions.

For the proposed framework, the input image is represented as a tensor of size $3 \times 256 \times 256$, where the first dimension represents the RGB channels. The convolutional blocks use 3×3 kernels and 64 output channels, so $C_{out} = 64$. Max-pooling operations are applied after feature extraction stages, continuously minimizing spatial dimensions from $256 \times 256 \rightarrow 128 \times 128 \rightarrow 64 \times 64 \rightarrow 32 \times 32$.

4. Experimental Results and Discussion

This section explains the experimental procedure for collecting, preparing, and evaluating the dataset. The real-world driving dataset was preprocessed for cleaning, labeling, splitting, rebalancing, and augmentation before being assessed by the models.

4.1. Dataset Description and Preparation

This paper uses a real-world dataset. The dataset was recorded across different city settings to ensure natural variation in lighting, weather, and traffic. The autonomous vision system was simulated by installing a small HD camera (1080p) on the vehicle's roof to continuously record street-level frames. First, the raw videos were split into image frames using the FFMPEG toolkit, ensuring that the flow of time across all weather conditions was represented in the various frames. Frames were extracted and organized into folders to ensure that the images remain in a common format, such as PNG, for further processing and training. This setup allowed the dataset to include a variety of challenging conditions, such as rain, fog, night lighting, and glare. Fig. 4 shows the camera and where it was placed on the vehicle during data collection.

4.2. Dataset Cleaning

To protect the integrity of the dataset, duplicate and non-duplicate frames were automatically detected and removed. After this cleaning process, the remaining images were classified into five major weather degradation classes: rain, rain_obstructed, fog, night, and nightglare. The final dataset, containing over 20,000 images with various types of naturally occurring weather degradation, was obtained after filtering and refinements. Table 4 presents a summary of the sample per the five weather-degradation categories contained in the dataset. All categories include real-world frames captured under normal environmental conditions.

The datasets are classified into five major degradation types: night, nightglare, rain, rain_smear, and fog, as shown in Fig. 5.

Table 4. Dataset composition across weather conditions.

Weather Condition	Number of Samples
Rain Streaks	4,634
Nighttime Glare	867
Night	4,921
Fog	500
Rain Smear	11,521

Fig. 5. The samples of labeled datasets in various categories of weather degradations.

4.3. Dataset Rebalancing

Real-life data collection may lead to an unequal distribution of classes, especially when capturing the variety of weather conditions. To solve this problem, this paper will adopt a class rebalancing approach that ensures equal and effective training across all weather categories and prevents biased training. The rebalancing process is done in three steps. First, the initial number of samples labeled for each weather condition is calculated from the structured data directories. Second, each class has a specific number of samples, set to ensure balanced representation. Third, underrepresented classes are calculated to have the

Fig. 6. Represent input images with their degradation labels. The lower images show visually enhanced references for illustration only.

Table 5. Dataset split across subsets.

Subset	Percentage	Number of samples	Purpose
Training	70%	~15,710	Model learning and optimization
Validation	20%	~4,489	Hyperparameter tuning, early stopping
Testing	10%	~2,244	Final evaluation and generalization testing

necessary number of additional samples. To reduce class imbalance, a minimum of 4000 samples per class was set. This target value of 4000 samples was chosen to balance class imbalance and avoid overfitting minority classes through oversampling.

4.4. Dataset Annotation and Reference Images

The dataset used in this paper was structured by predefined degradation categories. The images were manually categorized into five adverse visual conditions corresponding to the dominant environmental factor in the image. To ensure consistency in the dataset, the annotation process was conducted by visual inspection of degradation features, including reductions in illumination, fog density, visibility of rain streaks, smear artifacts, and nighttime glare. This paper does not use paired clean-degraded image supervision, and the enhanced visual images were not used in this classification task. Instead, these enhanced reference images are provided only as illustrative examples in Fig. 6 and are reserved for possible future work on adaptive image enhancement.

4.5. Dataset Splitting (Train/Validation/Test)

To provide an expected assessment of the classification and improvement models, the essential step to avoid overfitting was to divide the data into three subsets: training, validation, and testing. A 70% training, 20% validation, and 10% test split was adopted, providing enough data to optimize the model and keeping the independent test data to assess its performance without bias. Weight learning was performed on the training subset, hyperparameter tuning and early stopping for overfitting on the validation subset, and final model evaluation on the test subset. Table 5 displays the sample distribution of the subsets.

Fig. 7. Examples of augmented samples generated using different augmentation probabilities.

4.6. Data Augmentation Strategy

As stated in [20], the augmentation process needs to be customized by adaptively applying the computed class weights in order to overcome the dataset imbalance condition across all weather conditions. The pipeline to augment data includes lightweight, realistic transformations that protect scene semantics while improving diversity. Derived from the RandAugment framework [21] and TrivialAugmentWide [22], we employed two magnitude bins to introduce controlled variability. Additionally, Gaussian blur is added randomly using a kernel size of image.png and a sigma range of image.png to model motion blur and subtle defocus effects. Color jittering is also randomly applied with a probability of 0.3. These augmentation operations are intended to compensate for missing samples by simulating diverse real-world driving scenarios such as camera shake, lens glare, and very lightened scenes caused by strong light reflections. Fig. 7 illustrates representative examples of augmented samples.

To address class imbalance, an oversampling-based augmentation strategy is used. Let origN be the initial number of samples per class, and targetN be the target number. The additional number of samples that are necessary to balance is determined as follows:

$$\text{sample}_{class_{size}} = 4000$$

$$\text{extraN} = \text{targetN} - \text{OrigiN} \tag{7}$$

To measure the amount of augmentation needed to achieve a certain level of augmentation against each class, a class weight is calculated as:

$$\text{Weight} = \frac{extraN \times 100}{targetN} \tag{8}$$

The weight is used to indicate the proportion of synthetic augmentation required to balance samples across each category. More weighty classes need extra augmentation, whereas well-represented classes need a small amount or no augmentation. Using this strategy, learning can be done in a balanced manner without discarding useful real-world samples.

4.7. Experimental Setup and Implementation Details

The suggested TinyVGG-based degradation classification model was developed with the PyTorch deep learning framework. All the experiments were done on a system with an NVIDIA RTX 5060. The model was trained using the Adam optimizer with a fixed learning rate of 1e-3, a batch size of 8, and 100 epochs. The input images were scaled to 256 × 256 and normalized, and then fed into the network.

Table 6. Overall classification performance.

Metric	Value
Accuracy	98.17%
Precision (Macro)	98.19%
Recall (Macro)	98.25%
F1-Score (Macro)	98.20%
ROC-AUC (OVR, Macro)	0.9978
Avg. Confidence	0.948

Table 7. Per-class performance metrics.

Class	Precision	Recall	F1-Score	Support	Avg. Confidence
Rain Streaks	0.96	0.98	0.97	280	0.93
Night Glare	0.99	1.00	0.99	273	0.97
Night	0.99	1.00	0.99	285	0.96
Fog	0.99	1.00	0.99	278	0.97
Rain Smear	0.98	0.93	0.96	280	0.91

4.8. Quantitative Results

The TinyVGG-based degradation classifier achieves good performance in multi-class classification on the custom test dataset. The overall classification accuracy is 98.17%, indicating that it is possible to discriminate among various types of environmental degradation. At the macro level, the model has a precision of 98.19, a recall of 98.25, and an F1-score of 98.20, indicating fairness across all classes with no strong bias. Moreover, the proposed confidence estimation mechanism provides an additional reliability measure for the predicted classes. An overall average confidence score of approximately 0.948 is achieved, with higher values observed for correctly classified samples (up to 0.97), while relatively lower confidence scores (around 0.91–0.93) are recorded in more challenging cases such as rain smear and rain streak confusion. This shows that the confidence head is useful for capturing prediction uncertainty and improving the interpretability of model outputs. Further, the macro-ROC AUC of 0.9978 indicates a high level of separability between degradation classes, as shown in Table 6. The night, night glare, and fog classes are performing almost perfectly, with the night, night glare, and fog classes all having the same recall of 1.00, as indicated in Table 6. The rain streaks class has an F1-score of 0.97, and the rain smear class, which is a more complex degradation, has a slightly lower F1-score of 0.96 due to slight confusion with other rain-related classes.

Table 7 shows that rain-related degradations (rain smear and rain streaks) have lower confidence scores, indicating greater uncertainty between these two similar classes. Conversely, more robust confidence values are obtained for structurally distinct classes such as night and fog, which are more clearly separable.

Fig. 8 introduces the confusion matrix of the proposed model. The presence of strong diagonal patterns indicates that most samples are classified correctly. The principal misclassifications involve those related to rain, primarily between rain smear and rain streak, which are visually similar. On the other hand, there is insignificant confusion between structurally different classes, such as night and fog. In general, the results show that the proposed method can accurately extract discriminative features for different types of degradation with high classification consistency.

The proposed classifier has also undergone another experimental evaluation in its robustness from several experiments. The three independent runs used three random seeds

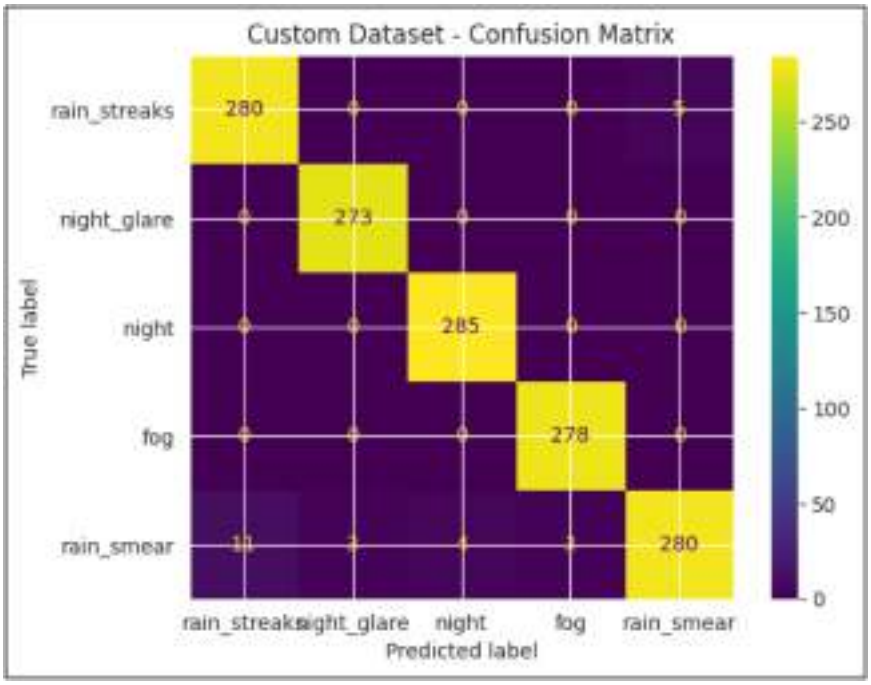

Fig. 8. Confusion matrix of the proposed TinyVGG classifier on the test dataset.

Table 8. Repeated-trial evaluation of the proposed classifier.

Metric	Run 1 (seed 42)	Run 2 (seed 64)	Run 3 (seed 195)	Mean ± Std
Accuracy (%)	96.06	94.59	93.81	94.82 ± 1.14
Precision (%)	96.07	94.64	93.94	94.88 ± 1.08
Recall (%)	95.95	94.59	93.75	94.76 ± 1.11
F1-Score (%)	95.93	94.39	93.55	94.62 ± 1.21
ROC-AUC	0.9911	0.9899	0.9888	0.9900 ± 0.0012
Avg. Confidence	0.9986	0.9981	0.9682	0.9883 ± 0.0174

(42, 64, and 195), respectively. All experiments used identical architecture, preprocessing steps, optimizer, learning rate, batch size, and 100 epochs of training. Mean and standard deviation of the main evaluation metrics are shown in Table 8.

The results are consistent across the three independent tests, with an average accuracy of 94.82% and a standard deviation of 1.14%. The high ROC-AUC value indicates that the classifier separates classes well for different random splits and augmentation patterns.

4.9. Ablation Study

To gain more insight into the contribution of the main entities to the proposed model, an ablation study was performed on data augmentation, dataset rebalancing, and confidence estimation. These factors were selected because they affect the handling of class imbalance, generalization effectiveness, and prediction reliability directly. The model was initialized without data augmentation and all the other settings were preserved. The results showed a distinct decrease in classification performance, particularly for minority classes, and less stable confidence estimates, indicative of less generalization capability. Second, the model

Table 9. Ablation study on augmentation, rebalancing, and confidence estimation.

Configuration	Accuracy (%)	F1-Score (%)	Confidence Reliability
Without Augmentation	95.8	96.1	Moderate
Without Rebalancing	94.9	95.3	Low
Without Confidence Head	97.6	97.8	Not Available
Full Model	98.17	98.20	High

Table 10. The quantitative comparison with existing methods.

Ref.	Architecture	Dataset	Dataset Type	Accuracy (%)	F1-Score (%)	Precision (%)
[7]	Lightweight-VGG	Hyperspectral dataset	Domain-specific classification	98.53–99.76	94.1	93.8
[11]	Lightweight CNN	General image dataset	Efficient classification	92.0	N/A	93.5
The proposed TinyVGG	CNN-VGG	Custom Dataset	Near real-time	98.17	98.20	98.19%

Performance values originate from different datasets and experimental protocols and should be interpreted only as contextual references.

was trained without the rebalancing strategy. This showed a bias in favor of the majority classes and lower accuracy for underrepresented classes as seen in this example, resulting in poor performance. It also produced lower confidence score values, especially for minority classes. Thirdly, the model was trained without the confidence estimation mechanism being included. While the classification performance was similar, the model had difficulty in assessing prediction reliability and as such, not applicable to augmentation, rebalancing, and confidence estimation. It worked best with greater class-wise consistency, consistent training behavior, and more plausible confidence predictions. These results show that all components add to the strength and stability of the proposed classifier, see Table 9.

4.10. Quantitative Comparison with Existing Methods

In line with this, we further demonstrated the efficiency of the proposed TinyVGG-based degradation classification model by comparing it with representative degradation methods. Given that the chosen literature was evaluated with various datasets and experimental designs, direct quantitative comparison may not be statistically valid. That is, the comparison is designed to give contextual insights into prior paradigms, rather than establish absolute superiority. Table 10 summarizes the characteristics of different representative methods, such as the kind of architecture, data used, and performance measures, from the original publications.

4.11. Computational Efficiency and Real-Time Capability Analysis

Autonomous driving requires perception systems to operate in real time while delivering low latency. Hence, the performance of the newly proposed TinyVGG classifier is judged in accordance with the network architecture, number of parameters, Floating Point Operations Per Second (FLOPs), Frames Per Second (FPS), latency, and input resolution (see Table 11). The comparison with MobileNetV2 [23], EfficientNet-B0 [24], and ShuffleNetV2 [25] is conducted on this architecture and computation, since these models have been tested on different datasets, tasks, hardware platforms, and hardware-configured implementation options. The TinyVGG model proposes an input resolution of 256 × 256 and parameters of 4.38M. It gains an inference rate of 88–93 FPS on an NVIDIA RTX 3060

Table 11. Efficiency comparison including FLOPs, FPS, and latency.

Model and Ref.	Parameters	FPS (GPU)	FLOPs (G)	Latency (ms)	Input Size
MobileNetV2 [23]	3.4M	~85–95 FPS	~0.3–0.6	~10–12	224 × 224
EfficientNet-B0 [24]	5.3M	~55–65 FPS	~0.39	~15–18	224 × 224
ShuffleNetV2 [25]	2.3M	~130–150 FPS	~0.15–0.30	~6–8	224 × 224
The proposed TinyVGG	4.38M	~88–93 FPS	~0.8–1.2	~10.7–11.3	256 × 256

Note that the FLOPs, FPS, and latency values of the baseline models are not only based on their original publications, but on generally accepted benchmarks in the literature.

GPU, corresponding to 10.7–11.3 ms per frame. We confirm that this proposed architecture could be a viable method for the classification of adverse visual conditions, weighing model complexity and inferential time. Thus, the comparison in Table 11 should be taken as a computational reference rather than a performance benchmark.

5. Conclusions

This paper introduces a lightweight TinyVGG-based model for classifying near-real-time degradation in unfavorable driving conditions. The suggested architecture focuses on effective detection of environmental distortions, including rain streaks, rain smear, fog, night, and night glare, using a compact convolutional architecture. The model incorporates stacked 3 × 3 convolutional layers and a simplified fully connected head, enabling strong discrimination and computational efficiency, making it ready for use in future embedded automotive applications. Experiments on the custom data showed high classification accuracy of 98.17%, with equal macro-level precision, recall, and F1 Scores. The confusion matrix analysis showed strong inter-class separability, with slight confusion mainly in visually similar categories, namely rain. Besides, the new framework is efficient in terms of inference speed and low computational cost, thanks to its lightweight structure, making it viable for real-world implementation. Overall, the suggested solution is an effective trade-off between accuracy and efficiency, providing a deployable solution to the problem of degradation-aware perception for autonomous driving systems.

Future research will extend the proposed framework to address mixed and overlapping AV conditions, in which multiple types of degradation occur in the same driving image. This can be done by switching from single-label recognition to multi-label degradation recognition. Furthermore, additional real-world samples will be collected under various traffic, weather, and illumination conditions to improve the classifier's robustness and generalization. The prepared enhanced reference images can also be used in future studies to develop adaptive image enhancement techniques based on the predicted degradation class and confidence score. Additionally, the model will be tested in embedded automotive hardware to further explore real-time deployment in a practical camera-based driving system.

Acknowledgment

The authors extend their heartfelt gratitude to the College of Computer Science at the University of Technology – Iraq, for providing academic support for this study. Their unwavering guidance and encouragement have been instrumental in the successful completion of this study.

Data availability

The dataset was collected and organized by the authors as part of this study and is available at: https://baghdad-ads.yousif.app/.

Authors contributions

Yousif N. Abbas: Conceptualized the study idea, collected and prepared the dataset, implemented the proposed TinyVGG-based classification framework, conducted the experiments, analyzed the results, and drafted the initial manuscript. Matheel E. Abdulmunim: Contributed to the methodological design, reviewed the experimental analysis, and provided critical revisions to improve the scientific quality of the paper. Nada H. Ali: Reviewed the dataset preparation and evaluation procedure, and provided technical feedback on the paper structure and content. Ismail A. Mageed reviewed the manuscript, helped interpret the results, and gave academic feedback to make the study clearer and better presented. All authors reviewed and approved the final version of the paper.

Conflict of interest

The authors declare no conflicts of interest concerning the publication of this paper.

Ethical and privacy considerations

The dataset was collected from public driving environments and used only for study purposes. No personally identifying information was intentionally collected, and visible sensitive information, such as faces or vehicle plates, was excluded or anonymized where applicable.

References

1. Z. Zhu and H. Zhao, "A survey of deep RL and IL for autonomous driving policy learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14043-14065, Sept. 2022, doi: 10.1109/TITS.2021.3134702.
2. C. Bernardeschi, G. Lami, F. Merola, and F. Rossi, "Verifying robustness of neural networks in vision-based end-to-end autonomous driving," *IEEE Access*, vol. 13, pp. 71688-71704, 2025, doi: 10.1109/ACCESS.2025.3563120.
3. X. Zhao, "Deep learning based visual perception and decision-making technology for autonomous vehicles," *Applied and Computational Engineering*, vol. 33, pp. 191-200, Feb. 2024, doi: 10.54254/2755-2721/33/20230265.
4. Y. Zhang, A. Carballo, H. Yang, and K. Takeda, "Perception and sensing for autonomous vehicles under adverse weather conditions: A survey," *Sensing*, vol. 196, pp. 146-177, Feb. 2023, doi: 10.1016/j.isprsjprs.2022.12.021.
5. M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proc. 2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 3213-3223, doi: 10.1109/CVPR.2016.350.
6. C. Sakaridis, D. Dai, and L. Van Gool, "ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding," in *Proc. of 2021 IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, Montreal, QC, Canada, Oct. 10-17, 2021, pp. 10745-10755, doi: 10.1109/ICCV48922.2021.01059.
7. X. Fei, S. Wu, J. Miao, G. Wang, and L. Sun, "Lightweight-VGG: A fast deep learning architecture based on dimensionality reduction and nonlinear enhancement for hyperspectral image classification," *Remote Sens.,* vol. 16, no. 2, Jan. 2024, Art. no. 259, doi: 10.3390/rs16020259.

8. A. Saini, K. Guleria, and S. Sharma, "An efficient TinyVGG model for sugarcane leaf disease classification in precision agriculture," in *Proc. 2024 2nd World Conf. on Communication & Computing (WCONF)*, Raipur, India, pp. 1-6, doi: 10.1109/WCONF61366.2024.10692002.
9. G. López, P. D. Ramírez, E. Vega, F. Pizarro, J. Toro, and C. Parra, "WeldVGG: A VGG-inspired deep learning model for weld defect classification from radiographic images with visual interpretability," *Sensors,* vol. 25, no. 19, Oct. 2025, Art. no. 6183, doi: https://doi.org/10.3390/s25196183.
10. M. Imani and D. Cerra, "Triple graph convolutional network for hyperspectral image feature fusion and classification," *Remote Sens.*, vol. 17, no. 9, May 2025, Art. no. 1623, doi:10.3390/rs17091623.
11. J. Zhang, "A lightweight CNN for efficient image classification based on knowledge distillation and structured pruning," in *Proc. of 2025 4th Int. Conf. on Electronic Information Technology (EIT)*, Chengdu, China, pp. 786-789, doi: 10.1109/EIT67313.2025.11232424.
12. R. S, A. S. Bharadwaj, D. S. K., M. S. Khadabadi, and A. Jayaprakash, "Digital implementation of the softmax activation function and the inverse softmax function," in *Proc. 2022 4th Int. Conf. on Circuits, Control, Communication and Computing (I4C)*, Bangalore, India, pp. 64-67, doi: 10.1109/I4C57141.2022.10057747.
13. R. Connor, A. Dearle, B. Claydon, and L. Vadicamo, "Correlations of cross-entropy loss in machine learning," *Entropy*, vol. 26, no. 6, Jun. 2024, Art. no. 491, doi: 10.3390/e26060491.
14. V. Shatravin, D. Shashev, and S. Shidlovskiy, "Sigmoid activation implementation for neural networks hardware accelerators based on reconfigurable computing environments for low-power intelligent systems," *Appl. Sci.*, vol. 12, no. 10, May 2022, Art. no. 5216, doi: 10.3390/app12105216.
15. J. Terven, D.-M. Cordova-Esparza, J.-A. Romero-González, A. Ramírez-Pedraza, and E. A. Chávez-Urbiola, "A comprehensive survey of loss functions and metrics in deep learning," *Artificial Intelligence Review,* vol. 58, no. 7, Art. no. 195, Apr. 2025, doi: 10.1007/s10462-025-11198-7.
16. D. Zhang *et al.*, "A dual-branch framework integrating the segment anything model and semantic-aware network for high-resolution cropland extraction," *Remote Sens.*, vol. 17, no. 20, Oct. 2025, Art. no. 3424, doi: 10.3390/rs17203424.
17. M. Reyad, A. M. Sarhan, and M. Arafa, "A modified Adam algorithm for deep neural network optimization," *Neural Computing and Applications,* vol. 35, no. 23, pp. 17095–17112, Apr. 2023, doi: 10.1007/s00521-023-08568-z.
18. S. Chang and J. C. Principe, "Explaining deep and ResNet architecture choices with information flow," in *Proc. 2022 Int. Joint Conf. on Neural Networks (IJCNN)*, Padua, Italy, pp. 1-6, doi: 10.1109/IJCNN55064.2022.9892065.
19. E. Hassan, S. Abu Ghazalah, N. El-Rashidy, T. Abd El-Hafeez, and M. Y. Shams, "DenseNet model with attention mechanisms for robust date fruit image classification," *International Journal of Computational Intelligence*, vol. 18, no. 1, Sep. 2025, Art. no. 228, doi: 10.1007/s44196-025-00809-4.
20. X. Liu, G. Karagoz, and N. Meratnia, "Analyzing the impact of data augmentation on the explainability of deep learning-based medical image classification," *Mach. Learn. Knowl. Extr.*, vol. 7, no. 1, Dec. 2024, Art. no. 1, doi: 10.3390/make7010001.
21. P. Drdák and P. Lehoczký, "Sliding RandAugment: Dynamic dual-window data augmentation for efficient traffic sign recognition," in *Proc. 2025 Int. Symp. ELMAR*, Zadar, Croatia, pp. 57-60, doi: 10.1109/ELMAR66948.2025.11194024.
22. A. Brehmer, J. Egger, and J. Kleesiek, "Evaluating data augmentation strategies for robust cross-dataset generalization in wound classification," in *Proc. 2025 IEEE 22nd Int. Symp. on Biomedical Imaging (ISBI)*, Houston, TX, USA, pp. 1-4, doi: 10.1109/ISBI60581.2025.10980725.
23. M. F. T. Sim, C. C. Anabeza, M. K. Cabatuan, J. R. B. D. Rosario, and A. A. Bandala, "MobileNetV2-based mobile application for brain tumor classification," in *Proc. 2022 IEEE 14th Int. Conf. on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, Boracay Island, Philippines, pp. 1-6, doi: 10.1109/HNICEM57413.2022.10109489.
24. C. Galabuzi, H. Abdullah, N. Ahmad, and H. M. Kaidi, "EfficientNet-based deep learning neural network for accurate plant disease detection," in *Proc. 2024 5th Int. Conf. on Smart Sensors and Application (ICSSA)*, Penang, Malaysia, pp. 1-6, doi: 10.1109/ICSSA62312.2024.10788558.
25. A. N. Yumang, J. M. Baguisi, B. R. S. Buenaventura, and C. C. Paglinawan, "Detection of black sigatoka disease on banana leaves using ShuffleNet v2 CNN architecture in comparison to SVM and KNN techniques," in *Proc. 2023 15th Int. Conf. on Computer and Automation Engineering (ICCAE)*, Sydney, Australia, pp. 281-286, doi: 10.1109/ICCAE56788.2023.10111367.