electronics

MDPI

Article

Efficient Weather Perception via a Lightweight Network with Multi-Scale Feature Learning, Channel Attention, and Soft Voting

Che-Cheng Chang, Po-Ting Wu, Ting-Yu Tsai and Jhe-Wei Lin *

Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan; checchang@fcu.edu.tw (C.-C.C.)

* Correspondence: jhewlin@fcu.edu.tw; Tel.: +886-4-24517250 (ext. 3758)

Abstract

Autonomous driving technology is advancing rapidly, particularly in vision-based approaches that use cameras to perceive the driving environment, which is the most human-like perception method. However, one of the key challenges that smart vehicles face is adapting to various weather conditions, which can significantly impact visual perception and vehicular control strategies. The ideal design for the latter is to dynamically adjust in real time to ensure safe and efficient driving, taking into account the prevailing weather conditions. In this study, we propose a lightweight weather perception model that incorporates multi-scale feature learning, channel attention mechanisms, and a soft voting ensemble strategy. This enables the model to capture various visual patterns, emphasize critical information, and integrate predictions across multiple modules for improved robustness. Benchmark comparisons are conducted using several well-known deep learning networks, including EfficientNet-B0, ResNet50, SqueezeNet, MobileNetV3-Large, MobileNetV3-Small, and LSKNet. Finally, using both public datasets and real-world video recordings from roads in Taiwan, our model demonstrates superior computational efficiency while maintaining high predictive accuracy. For example, our model achieves 98.07% classification accuracy with only 0.4 million parameters and 0.19 GFLOPs, surpassing several well-known CNNs in computational efficiency. Compared with EfficientNet-B0, which has a similar accuracy (98.37%) but requires over ten times more parameters and four times more FLOPs, our model offers a much lighter and faster alternative.

Keywords: autonomous driving; weather perception; multi-scale feature learning; channel attention; soft voting

Check for updates

Academic Editor: George A. Tsihrintzis

Received: 17 November 2025
Revised: 13 December 2025
Accepted: 17 December 2025
Published: 19 December 2025

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Weather perception plays an important role in several advanced applications, including Autonomous Vehicles (AVs), Advanced Driver Assistance Systems (ADASs), drones, and so on. Accurate weather sensing enables these systems to adapt to various environmental conditions and improve safety and operational efficiency. For instance, the stopping distance of a vehicle is influenced by both the thinking distance and the braking distance. Both are affected by the weather. Adverse conditions can degrade environmental perception, lengthen the thinking distance, alter vehicle dynamics, and increase the braking distance. This shows the critical importance of effective weather perception techniques. Traditionally, weather perception has relied on basic binary classifications, e.g., detecting

rain to activate wipers, determining daylight to control headlights, and so forth. While simple binary sensors support these functions, they fall short of the complex requirements for modern smart vehicles. Additionally, weather data acquired through external communication only provides general forecasts for an area, which lacks real-time and localized perception accuracy.

In a notable study [1], the authors propose a deep learning-based concept for weather classification to meet the needs of modern smart vehicles. Their method classifies six distinct weather conditions: shine, sunrise, cloudy, rainy, snowy, and sandy. These categories cover a broad range of environmental variability that we may meet. Particularly, the authors use a combined dataset of two public and widely used weather image datasets, MCWRD2018 [2] and DAWN2020 [3], to evaluate their models' effectiveness. Merging these datasets ensures diverse weather conditions and enhances their models' generalization capabilities. Next, to achieve high accuracy in weather classification, they employ the transfer learning technique [4–6]. This allows models to utilize knowledge gained from large-scale datasets and apply it to specific tasks with limited data, thus improving training efficiency. The core of their approach involves the development built upon three well-known convolutional neural network (CNN) architectures: SqueezeNet, ResNet-50, and EfficientNet-B0. These architectures offer their corresponding balance between computational complexity and classification performance. They are all well-suited for deployment in real-world systems where resource constraints and inference speed are critical considerations. Finally, in [1], their models show strong performance across all six weather classes through several experiments and highlight the potential for practical use in modern smart vehicles.

This paper gives two critical enhancements over the work presented in [1]. First, while [1] focuses on only six weather classes and excludes the fog category, we extend the classification scope by incorporating all seven weather classes in the combined MCWRD2018 and DAWN2020 datasets [2,3]. Second, we introduce a lightweight weather perception model, which incorporates multi-scale feature learning, channel attention mechanisms, and a soft voting ensemble strategy to enable it to capture various visual patterns, emphasize critical information, and integrate predictions across multiple modules for improved robustness. Also, it has fewer trainable parameters and floating-point operations (FLOPs), while maintaining strong performance. In other words, our model achieves lower computational complexity without sacrificing inference accuracy. The experiments in this paper evaluate three well-known models discussed in [1], along with MobileNetV3-Large, MobileNetV3-Small, and LSKNet. Note that the latter three models are additionally considered in this study to provide a more complete analysis. For fairness, all model comparisons in this study are conducted utilizing versions without pretraining on external datasets. Moreover, using both public datasets and real-world video recordings from roads in Taiwan, our model demonstrates superior computational efficiency while maintaining high predictive accuracy.

The remainder of this paper is organized as follows. Section 2 provides the fundamental concepts that form the basis of this study. Section 3 details the proposed methodologies, including the experimental procedures and our network architecture. In Section 4, we present and analyze the experimental results. Finally, the last section concludes this paper and discusses some potential future directions.

2. Preliminary

In [7], learning scene-level 3D representations by pre-training on easier-to-collect shape datasets is proposed. It aggregates multiple shapes into pseudo-scenes and utilizes multi-scale high-resolution backbones, applying a point–point contrastive loss to bridge the gap between shape-level and scene-level domains. In [8], the authors propose a weather-

robust 3D detection framework that leverages radar signals to prompt a pre-trained LiDAR model and applies multi-level knowledge distillation, enabling the LiDAR-only model to inherit radar-invariant features. These works demonstrate that weather perception can be achieved through various sensing data. In [9], the model targets multi-label weather recognition, employing a CNN backbone combined with a Transformer strategy to model label dependencies. Because it incorporates Transformer blocks and multi-label correlation modeling, its computational cost is substantially higher than ours, making it difficult to realize in a simple embedding system. In contrast to works in [10,11] (the former employs a variety of synthetic weather degradations and denoising strategies to make object detection robust under adverse conditions, while the latter relies on synthetic rain generation (analytical, neural style transfer, and CycleGAN) to enhance transformer-based detection performance in rainy scenes; both focus on maintaining task performance while ignoring or compensating for weather effects), our objective is fundamentally different. Rather than treating weather as a nuisance factor to be removed, our model explicitly aims to recognize the weather itself, as accurate weather identification is essential for generating appropriate and safe vehicle motion commands in autonomous driving scenarios. Hence, in this paper, we focus specifically on camera-based sensing and the design of CNN architectures for weather perception. The following provides the fundamental concepts that form the basis of this study.

2.1. Convolutional Neural Network and Standard Designs

A convolutional neural network is a class of deep learning models widely used for image classification tasks. They can extract spatial features from images through convolutional layers, which apply filters to detect patterns, e.g., edges, shapes, textures, and so on. These features are then processed by pooling layers and activation functions, reducing dimensionality while preserving essential features. Eventually, fully connected layers map the extracted features to the target classes [6,12–14]. CNNs are well-known for their high accuracy in object and scene recognition. They can learn hierarchical representations directly from raw pixel data without manual feature engineering, which has made them the foundation of modern computer vision.

Here, we introduce the well-known CNN architectures adopted in our experiments, each featuring distinct design characteristics, including the three models discussed in [1], along with MobileNetV3-Large and MobileNetV3-Small, which are additionally included in this study to provide a more complete analysis.

- EfficientNet-b0 is the foundational model in the EfficientNet series. Instead of scaling depth, width, or resolution independently as in conventional CNNs, it applies a compound scaling strategy to adjust all dimensions in a balanced way. This concept makes EfficientNet-b0 well-suited for practical deployments where both computational efficiency and accuracy are essential, providing a significant trade-off among speed, model size, and predictive performance [15].
- ResNet-50 is a 50-layer deep convolutional neural network from the Residual Network series. Its key innovation lies in shortcut (skip) connections that form residual learning to effectively address the vanishing gradient issue in very deep networks. By allowing layers to learn residual mappings, ResNet-50 can train deeper architectures more reliably and efficiently [16].
- SqueezeNet is a lightweight CNN architecture that achieves high accuracy while drastically reducing the number of model parameters. Its structure minimizes memory usage and computational demands, which makes it suitable for resource-constrained environments, e.g., Internet of Things (IoT) applications, embedded devices, and so on [17].

- MobileNetV3 is also a lightweight CNN architecture for mobile and edge devices. It builds upon the inverted residual blocks of MobileNetV2, Neural Architecture Search (NAS) techniques from EfficientNet, Squeeze-and-Excitation (SE) modules, and the efficient h-swish activation function [18]. MobileNetV3 has two main variants: MobileNetV3-Large and MobileNetV3-Small. The Large version is designed for tasks requiring higher accuracy. It uses deeper and wider layers with SE modules to enhance channel attention. In contrast, the Small version is optimized for real-time and ultra-low power applications. With a lighter architecture, it runs significantly faster with lower latency. Thus, MobileNetV3 provides flexible options depending on the deployment scenario.
- LSKNet addresses the challenge of detecting small and diverse objects in remote sensing images [19]. The key idea is to use large and adaptive convolutional kernels that dynamically adjust the receptive field according to different regions and object types, effectively capturing both fine details and global contextual information. This work highlights the potential of dynamically designed receptive fields for the unique characteristics of remote sensing imagery, offering a new perspective for future model architectures in aerial object detection.

2.2. Critical Components in Our Model

Our weather perception model incorporates multi-scale feature learning, channel attention mechanisms, and a soft voting ensemble strategy to capture various visual patterns, emphasize critical information, and integrate predictions across multiple modules for improved robustness. The following are the abstracts for each component.

- Multi-scale feature learning integrates features extracted at different scales via dense blocks with various sizes. It aggregates low-level and high-level information through dense and skip connections, while a bottleneck layer compresses the concatenated feature maps to reduce redundancy. This design allows the network to simultaneously capture fine details and broader contextual features to achieve balance [20].
- The channel attention mechanism used here is Squeeze-and-Excitation, which aims to enhance the representational power of CNNs by adaptively recalibrating channel-wise feature responses. Instead of treating all feature channels equally, SE blocks explicitly model inter-channel dependencies. This is achieved through two operations: a squeeze step aggregating global spatial information into compact channel descriptors via global average pooling and an excitation step employing a lightweight gating mechanism to capture channel relationships and generate adaptive weights. These weights are then used to rescale the feature maps, strengthen informative channels, and suppress less useful ones [15].
- Soft voting is an ensemble strategy in which each classifier contributes a probability distribution over all classes. Instead of simply voting for one class, every model provides its estimated probability for each possible category. These probability values are then aggregated, although weighted combinations can also be used when specific models are considered more reliable. After aggregation, the class with the highest combined probability is selected as the final prediction. On the other hand, compared with hard voting, which only counts the discrete class decisions of each classifier, soft voting preserves information about each model's confidence level. This makes the ensemble more stable and typically more accurate. Soft voting can be applied not only across multiple independent classifiers but also within a single CNN architecture. When a CNN contains multiple branches, each branch can be treated as an individual learner that produces its own class-probability output, e.g., multi-scale feature extraction, multi-path feature extraction, and so on. By aggregating these branch-level

probability distributions through soft voting, the model effectively integrates complementary information captured by different branches, leading to improved robustness and overall classification performance [21,22].

3. Our Method

This section presents our methodology, including the architecture of our proposed convolutional neural network as well as the training and testing procedures.

3.1. Architecture of Our Proposed Convolutional Neural Network

The overall architecture is composed of three major modules shown in Figures 1–3. These modules are integrated to enhance feature representation and improve the final decision process. The complete framework is summarized in Figure 4.

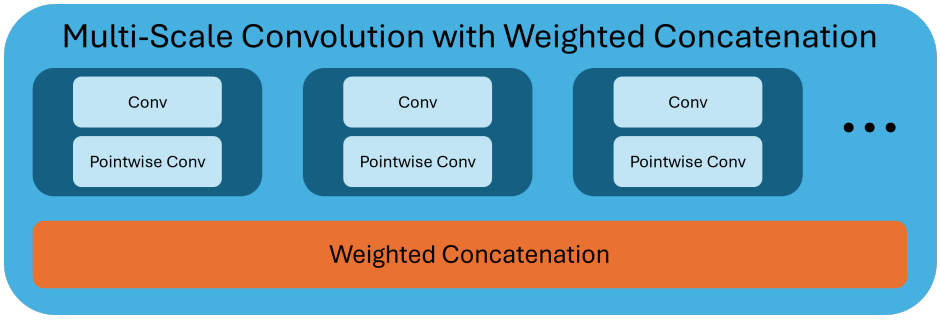

Figure 1. The multi-scale convolution with weighted concatenation module.

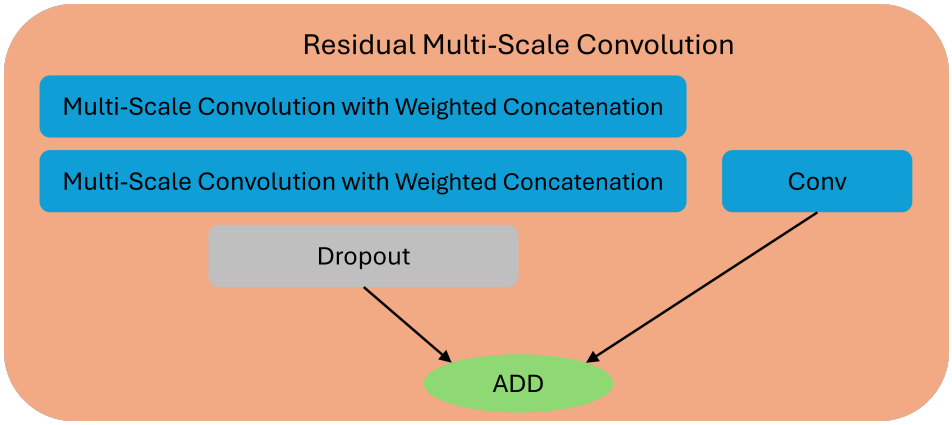

Figure 2. The residual multi-scale convolution module.

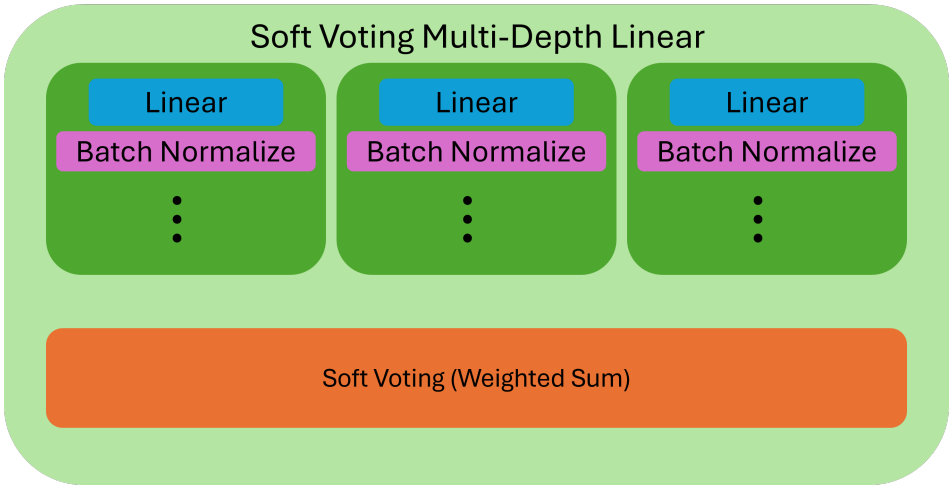

Figure 3. The soft voting multi-depth linear module.

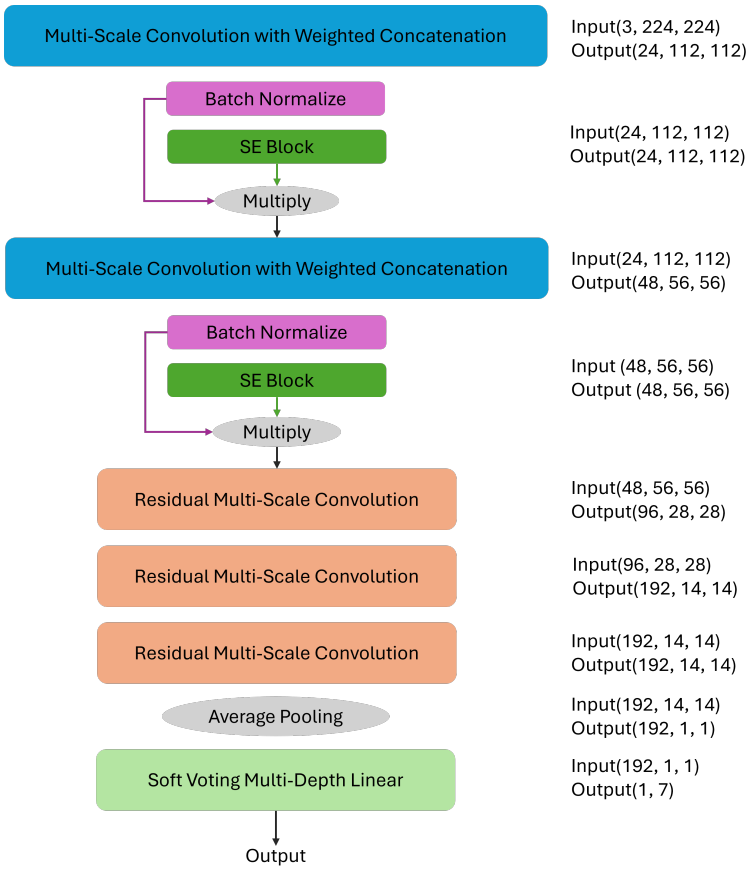

Figure 4. The overall architecture.

First, in Figure 1, the multi-scale convolution module consists of several parallel convolutional branches. Each branch applies a standard convolution followed by a pointwise convolution. Using different kernel sizes enables the extraction of multi-scale features, capturing both local and global spatial patterns, e.g., 1×1, 3×3, 5×5, and 7×7. The outputs of all branches are then fused through a weighted concatenation mechanism, which performs a weighted combination of branch outputs using weights learned automatically during training. This ensemble-like fusion enhances robustness and enables the model to emphasize informative feature scales adaptively. Note that, unlike standard convolution, which jointly processes spatial and channel dimensions, pointwise convolution operates only across channels, making it an efficient mechanism for channel mixing, dimensionality reduction, and feature refinement [23].

Next, the residual multi-scale convolution module (Figure 2) is constructed by stacking multiple multi-scale convolution blocks while incorporating residual connections and dropout layers. The residual structure helps the gradient vanishing problem and facilitates deeper feature learning, whereas the dropout operation effectively reduces overfitting. Through these designs, the module ensures efficient feature propagation and stability in deeper representations.

In Figure 3, the soft voting multi-depth linear module contains several parallel fully connected branches with different depths. Each branch consists of linear layers followed by batch normalization, enabling the extraction of features at multiple levels of abstraction. Their outputs are then combined through soft voting, which calculates a weighted sum of the branch outputs. This mechanism enables the model to automatically determine the appropriate emphasis to place on each branch, resulting in a more robust and balanced decision that reflects both shallow and deep feature transformations.

Finally, in Figure 4, the complete architecture begins with an input image of the size $3 \times 224 \times 224$, where the three channels correspond to the RGB (Red–Green–Blue) color

space and 224 × 224 denotes the spatial resolution. After initial preprocessing, the network interprets this input as a feature map of size $C \times H \times W$, where C denotes the channel dimension and H and W represent the spatial width and height, respectively. The input is first processed by two multi-scale convolution modules, each followed by batch normalization and an SE block to recalibrate channel-wise responses. These operations preserve the spatial dimensions while enriching the channel representation. The resulting feature maps are then passed through several residual multi-scale convolution modules, which progressively strengthen hierarchical feature learning. Afterward, an average pooling layer compresses the spatial dimensions into a compact feature vector. This vector is forwarded to the soft voting multi-depth linear module, which integrates multiple levels of linear transformations through a learned weighted combination of parallel branches, ultimately producing a final output of the size 1×7, corresponding to the seven weather categories.

3.2. Training and Testing Phases

The pseudocode is provided in Algorithm 1. First, a dataset, *data*, derived from combining MCWRD2018 and DAWN2020 in the input phase, is initially utilized as the sole input source (lines 1–2). Next, in the variable phase, several variables are defined for the pseudocode: *augmenting*, *training*, and *testing* serve as the three datasets for subsequent steps, *model* denotes the proposed deep learning model, and *benchmarks* represent the five baseline models (lines 3–8).

In preprocessing, *augmenting* is generated through data augmentation from *data*. It is then partitioned into two subsets, with 80% allocated to training for training and 20% to testing for evaluation (lines 9–12). During the training phase, both *model* and *benchmarks* are trained using *training* (lines 13–15). These models are assessed on *testing* in the evaluation phase (lines 16–18). To guarantee fairness in performance comparison, the contents of the training and testing datasets are kept mutually exclusive [4,5].

Algorithm 1: Training and testing phases for weather perception

1 **Input phase:**
2 *data*: an integration of the MCWRD2018 and DAWN2020 datasets;
3 **Variable phase:**
4 *augmenting*: a dataset;
5 *training*: a dataset;
6 *testing*: a dataset;
7 *model*: our deep learning model;
8 *benchmarks*: all comparisons (EfficientNet-B0, ResNet50, SqueezeNet, MobileNetV3-Large, MobileNetV3-Small, and LSKNet);
9 **Initial phase:**
10 *augmenting* ← **augment**(*data*);
11 *training* ← **split**(*augmenting*, 80%);
12 *testing* ← **split**(*augmenting*, 20%);
13 **Training phase:**
14 **train** *model* using *training*;
15 **train** *benchmarks* using *training*;
16 **Evaluating phase:**
17 **evaluate** *model* using *testing*;
18 **evaluate** *benchmarks* using *testing*;

4. Experimental Results

The experimental configuration is summarized as follows. Specifically, we describe the choice of datasets, augmentation techniques, data partitioning strategies, model input resolution, hyperparameters, and benchmarks. These settings were carefully designed to ensure the experimental results' reliability and reproducibility.

- Datasets: MCWRD2018 and DAWN2020 were employed as the original datasets (images illustrating the seven distinct weather categories are shown in Figure 5).
- The real-world dataset consists of dashcam video recordings from roads in Taiwan under four different weather conditions. From each video sequence, 50 frames were uniformly sampled to construct the final image set.
- Training and testing split: The original (unaugmented) dataset was first randomly divided into mutually exclusive training (80%) and testing (20%) subsets. This ensured that no image or its variants appeared in both subsets.
- Data augmentation: After splitting the original dataset into mutually exclusive training and testing subsets, augmentation was applied only within each subset to avoid data leakage. Each image was further augmented by horizontal flipping and by cropping 10% from each of the four sides.
- Input resolution: $3 \times 224 \times 224$.
- Output resolution: 1×7.
- Epochs: 300.
- Batch size: 16.
- Learning rate: 0.0001.
- Loss function: CrossEntropyLoss.
- Optimizer: Adam.
- Implementation: Python 3.12 [24] with the PyTorch 2.6.0 library [25].
- Benckmarks: EfficientNet-B0, ResNet50, SqueezeNet, MobileNetV3-Large, MobileNetV3-Small, and LSKNet.



Figure 5. Images illustrating the seven distinct weather categories.

Table 1 compares our model with several widely adopted convolutional neural networks, including EfficientNet-B0, ResNet50, SqueezeNet, MobileNetV3-Large, MobileNetV3-Small, and LSKNet. According to the results, our model achieves its best performance in epoch 276 with an accuracy of 98.07%. Although EfficientNet-B0 achieves the highest accuracy of 98.37%, it requires a model size of 15.62 MB, 4.02M parameters, and 0.82 GFLOPs, which makes it considerably more computationally demanding than ours. Next, ResNet50 achieves 96.52% accuracy but at the cost of 90.04 MB in size, 23.52 million parameters, and 8.26 GFLOPs, making it the heaviest among all compared models. SqueezeNet exhibits relatively low complexity, but its accuracy (96.98%) re-

mains notably lower than ours. MobileNetV3-Large achieves the same accuracy as our model (98.07%), but it requires more computational resources, making it less efficient for lightweight deployment. In contrast, MobileNetV3-Small offers a compact architecture but with slightly reduced accuracy (97.45%). Finally, LSKNet achieves 97.83% accuracy with a large model size, indicating that while it performs well, it comes with significantly higher computational overhead. These results collectively demonstrate that our model achieves a superior trade-off between accuracy, model size, parameter count, and computational efficiency, making it particularly competitive for real-time or resource-constrained applications.

Although all models are evaluated, their confusion matrices display broadly similar trends. Across all architectures, most categories are classified with high accuracy, and the few misclassifications consistently appear among visually identical conditions. For example, fog is occasionally mistaken for rain or snow, and snow is sometimes confused with rain. These are also reflected in our model's confusion matrix (Figure 6), where cloudy, sunrise, and shine achieve nearly perfect recognition, while minor errors occur mainly in fog and rain. Compared with other models, such as EfficientNet, MobileNetV3, ResNet50, and SqueezeNet, which show slightly higher counts in fog, rain, and sand, our model consistently produces fewer misclassifications. This indicates that our architecture not only follows the overall trend observed across models but also achieves the most accurate and stable performance.

Table 1. A comparison across all experimental models.

Model Name	Best Epoch	Accuracy(%)	Model Size (MB)	Parameters (M)	GFLOPs
Our model	276	98.07	1.6	0.4	0.19
EfficientNet-B0	262	98.37	15.62	4.02	0.82
ResNet50	267	96.52	90.04	23.52	8.26
SqueezeNet	235	96.98	4.81	1.26	1.67
MobileNet V3-Large	299	98.07	16.27	4.21	0.46
MobileNet V3-Small	198	97.45	5.95	1.53	0.12
LSKNet	246	97.83	65.87	17.16	7.06

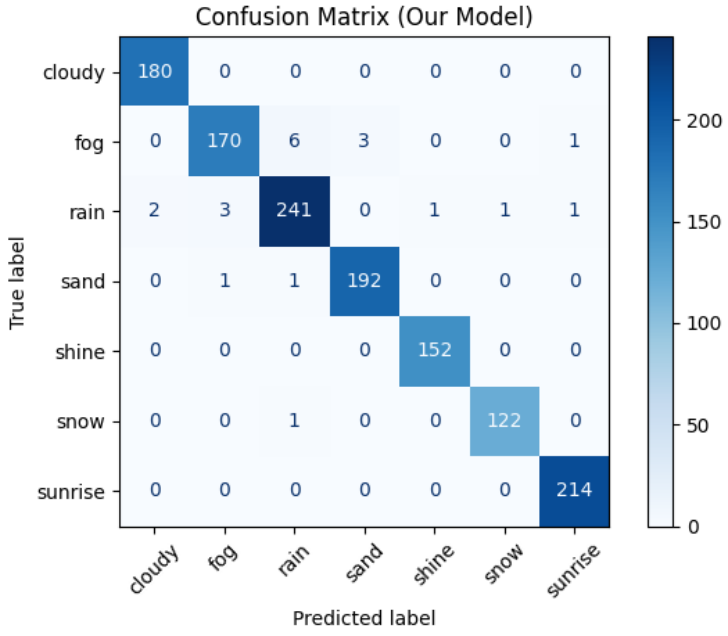

Figure 6. Confusion matrix of our model across seven weather categories.

Next, the results presented in Table 2 report the confidence levels of all models under different weather scenarios in MCWRD2018 and DAWN2020, including cloudy, fog, rain, sand, shine, snow, and sunrise. Most models exhibit strong robustness and stable confidence across various environmental conditions. Focusing on our model, it consistently demonstrates high confidence across all weather types, particularly achieving 99.82% under both cloudy and sunrise conditions. Although our model performs competitively overall, it exhibits slightly lower confidence in specific categories, such as fog and rain, compared to EfficientNet-B0 and LSKNet. This primarily arises from two factors. First, fog, rain, and snow share visually similar low-contrast textures, making them among the most challenging weather conditions for all models. Some larger networks naturally benefit from larger representational capacity. Second, EfficientNet-B0 and LSKNet contain substantially more parameters, enabling them to capture subtle visual cues that lightweight architectures may not model fully. Despite these minor differences, our model still maintains a favorable trade-off between accuracy, model size, and computational efficiency, achieving competitive confidence levels while significantly reducing computational demand.

Additionally, real-world video recordings collected from roads in Taiwan are used for evaluation. Some examples are shown in Figure 7, and the corresponding statistics of confidence levels are summarized in Table 3. As expected, all models exhibit lower confidence due to the domain shift between training and evaluation data. Even under this condition, our model still achieves higher confidence across all tested conditions, demonstrating strong robustness. Although some models show slightly higher confidence in specific weather types, notably rain, these cases generally involve architectures with significantly larger parameter counts, which provide stronger capacity for handling complex real-world textures. In contrast, our lightweight model maintains competitive confidence while preserving superior computational efficiency.

Table 2. A comparison of confidence levels across all experimental models.

Performance Metric	**Confidence (%)**						
Model Name	**Cloudy**	**Fog**	**Rain**	**Sand**	**Shine**	**Snow**	**Sunrise**
Our model	99.82%	91.79%	96.39%	97.92%	98.97%	98.33%	99.82%
EfficientNet-B0	99.61%	93.34%	97.03%	98.49%	99.73%	99.67%	100.00%
ResNet50	97.45%	88.54%	93.87%	95.15%	97.68%	99.35%	99.99%
SqueezeNet	99.73%	89.21%	93.87%	95.06%	95.25%	97.85%	99.84%
MobileNet V3-Large	97.68%	92.01%	96.93%	98.53%	99.94%	99.15%	99.96%
MobileNet V3-Small	97.51%	89.28%	95.57%	98.41%	98.79%	97.70%	99.58%
LSKNet	100%	98.21%	94.96%	96.97%	98.71%	99.16%	98.62%

Table 3. A comparison of confidence levels across all experimental models on our Taiwan dataset.

Performance Metric	**Confidence (%)**			
Model Name	**Cloudy**	**Rain**	**Shine**	**Sunrise**
Our model	99.98%	98.01%	98.05%	99.81%
EfficientNet-B0	73.61%	98.96%	76.11%	90.48%
ResNet50	89.44%	98.78%	94.22%	88.73%
SqueezeNet	96.87%	98.87%	72.78%	76.73%
MobileNet V3-Large	85.43%	99.21%	83.87%	86.69%
MobileNet V3-Small	78.49%	81.97%	90.73%	72.12%
LSKNet	93.55%	96.85%	79.33%	93.71%

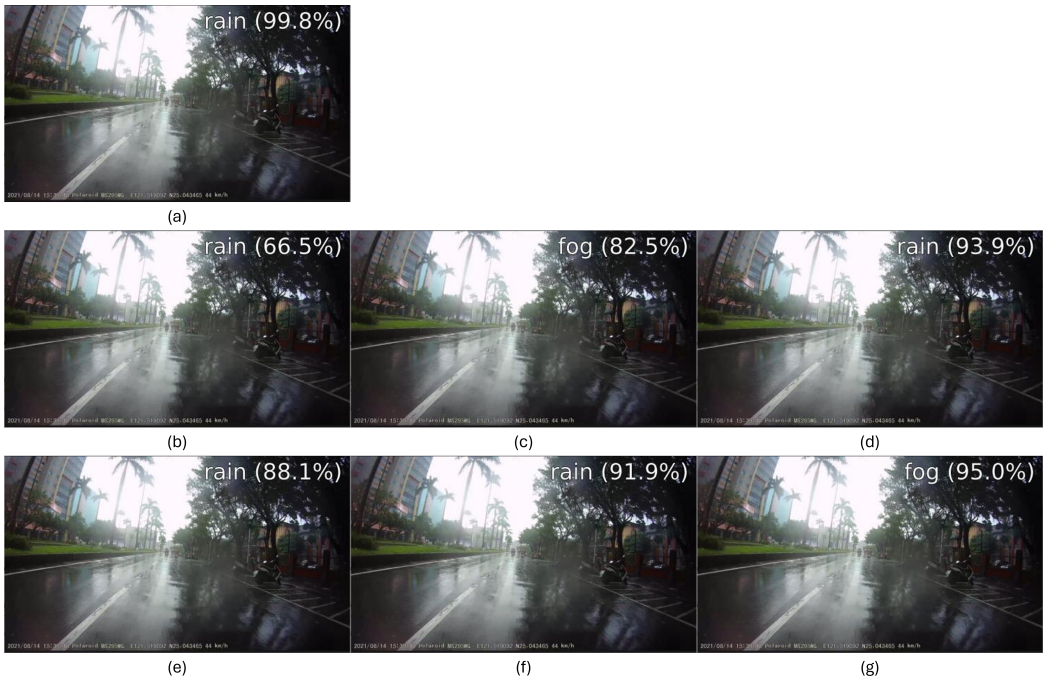

Figure 7. Comparison of recognition results under rainy scenes for (**a**) our model, (**b**) EfficientNet-B0, (**c**) ResNet50, (**d**) SqueezeNet, (**e**) MobileNetV3-Large, (**f**) MobileNetV3-Small, and (**g**) LSKNet.

5. Conclusions and Future Work

This work introduces a lightweight weather perception model combining multi-scale feature learning, channel attention, and soft voting. Experiments on public datasets and real-world recordings demonstrate that our model achieves 98.07% accuracy with only 0.4 million parameters and 0.19 GFLOPs, outperforming several well-known CNNs in terms of efficiency while maintaining competitive accuracy. Compared to larger architectures (EfficientNet-B0, ResNet-50, MobileNetV3-Large, and LSKNet), which benefit from greater parameter capacity but require substantially higher computational costs, our model provides a more favorable trade-off between performance and complexity. Meanwhile, compared to other lightweight baselines (SqueezeNet and MobileNetV3-Small), our model consistently delivers higher accuracy and confidence levels. These comparisons highlight the effectiveness of our architectural choices and demonstrate that lightweight designs can achieve strong performance without sacrificing real-time applicability.

In future work, we plan to enrich the evaluation scope by incorporating datasets under rarer and more challenging environmental conditions from various areas, such as occlusion, nighttime, and composite or extreme-weather scenarios. This will enable a more comprehensive validation of our model's robustness and generalization ability. To further enhance real-world applicability, we will continue to optimize the model for embedded and edge devices, aiming to improve both computational efficiency and real-time performance. Additionally, we are interested in adopting generative approaches to synthesize diverse and realistic data samples, thereby expanding dataset coverage and improving model resilience across varied application domains.

Author Contributions: Conceptualization, C.-C.C., P.-T.W. and T.-Y.T.; methodology, C.-C.C., P.-T.W., T.-Y.T. and J.-W.L.; software, P.-T.W. and T.-Y.T.; validation, C.-C.C., P.-T.W. and T.-Y.T.; formal analysis, C.-C.C., P.-T.W. and T.-Y.T.; writing—original draft preparation, C.-C.C. and J.-W.L.; writing—review and editing, C.-C.C. and J.-W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the National Science and Technology Council, Taiwan, R.O.C., under grant 112-2221-E-035-062-MY3.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data will be made available on reasonable request from the corresponding author.

Conflicts of Interest: The authors declare that they have no conflict of interest to this work.

References

1. Al-Haija, Q.A.; Gharaibeh, M.; Odeh, A. Detection in Adverse Weather Conditions for Autonomous Vehicles via Deep Learning. *AI* **2022**, *3*, 303–317. [CrossRef]
2. Multi-Class Weather Dataset for Image Classification—Mendeley Data. Available online: https://data.mendeley.com/datasets/4drtyfjtfy/1 (accessed on 7 December 2025).
3. DAWN—Mendeley Data. Available online: https://data.mendeley.com/datasets/766ygrbt8y/3 (accessed on 7 December 2025).
4. Hwang, K. *Cloud Computing for Machine Learning and Cognitive Applications*; The MIT Press: Cambridge, MA, USA, 2017.
5. Hwang, K.; Chen, M. *Big-Data Analytics for Cloud, IoT and Cognitive Computing*; Wiley: Hoboken, NJ, USA , 2017.
6. Ooi, Y.-M.; Chang, C.-C.; Su, Y.-M.; Chang, C.-M. Vision-Based UAV Localization on Various Viewpoints. *IEEE Access* **2025**, *13*, 38317–38324. [CrossRef]
7. Feng, T.; Wang, W.; Quan, R.; Yang, Y. Shape2Scene: 3D Scene Representation Learning Through Pre-training on Shape Data. In Proceedings of the Computer Vision—ECCV 2024: 18th European Conference, Milan, Italy, 29 September–4 October 2024.
8. Chae, Y.; Kim, H.; Oh, C.; Kim, M.; Yoon, K.-J. LiDAR-Based All-Weather 3D Object Detection via Prompting and Distilling 4D Radar. In Proceedings of the Computer Vision—ECCV 2024: 18th European Conference, Milan, Italy, 29 September–4 October 2024.
9. Chen, S.; Shu, T.; Zhao, H.; Tang, Y.Y. MASK-CNN-Transformer for Real-Time Multi-Label Weather Recognition. *Knowl.-Based Syst.* **2023**, *278*, 110881. [CrossRef]
10. Gupta, H.; Kotlyar, O.; Andreasson, H.; Lilienthal, A.J. Robust Object Detection in Challenging Weather Conditions. In Proceedings of the 2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2024.
11. Hao, C.-Y.; Chen, Y.-C.; Chen, T.-T.; Lai, T.-H.; Chou, T.-Y.; Ning, F.-S.; Chen, M.-H. Synthetic Data-Driven Real-Time Detection Transformer Object Detection in Raining Weather Conditions. *Appl. Sci.* **2024**, *14*, 4910. [CrossRef]
12. Abbass, M.J.; Lis, R.; Awais, M.; Nguyen, T.X. Convolutional Long Short-Term Memory (ConvLSTM)-Based Prediction of Voltage Stability in a Microgrid. *Energies* **2025**, *17*, 1999. [CrossRef]
13. Hosna, A.; Merry, E.; Gyalmo, J.; Alom, Z.; Aung, Z.; Azim, M. Transfer Learning: A Friendly Introduction. *J. Big Data* **2022,** *9*, 102. [CrossRef] [PubMed]
14. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A Comprehensive Survey on Transfer Learning. *Proc. IEEE* **2021**, *109*, 43–76. [CrossRef]
15. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019.
16. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, Nevada, USA, 27–30 June 2016.
17. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. *arXiv* **2016**, arXiv:1602.07360.
18. Howard, A.; Sandler, M.; Chen, B.; Wang, W.; Chen, L.-C.; Tan, M. Searching for MobileNetV3. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
19. Li, Y.; Hou, Q.; Zheng, Z.; Cheng, M.-M.; Yang, J.; Li, X. Large Selective Kernel Network for Remote Sensing Object Detection. In Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–3 October 2023.
20. Lee, Y.; Jun, D.; Kim, B.-G.; Lee, H. Enhanced Single Image Super Resolution Method Using Lightweight Multi-Scale Channel Dense Network. *Sensors* **2021**, *21*, 3351. [CrossRef] [PubMed]
21. Rokach, L. Ensemble-based classifiers. *Artif. Intell. Rev.* **2010**, *33*, 1–39. [CrossRef]
22. Zhou, Z.-H. *Ensemble Methods: Foundations and Algorithms*; CRC: Boca Raton, FL, USA, 2012.
23. Liang, P.F.; Tian, Z.; Dong, M.; Cheng, S.; Sun, L.; Li, H.; Chen, Y.; Zhang, G. Efficient Neural Network Using Pointwise Convolution Kernels with Linear Phase Constraint. *Neurocomputing* **2021**, *423*, 572–579. [CrossRef]

24. Welcome to Python.org. Available online: https://www.python.org/ (accessed on 7 December 2025).
25. PyTorch. Available online: https://pytorch.org/ (accessed on 7 December 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.