RAIRO-Oper. Res. 60 (2026) 2021–2042
https://doi.org/10.1051/ro/2026053

RAIRO Operations Research
www.rairo-ro.org

VISION TRANSFORMER-BASED AUTOMATIC WEATHER DETECTION WITH ADAPTIVE TRANSFER LEARNING USING BAYESIAN OPTIMIZATION AND GENETIC ALGORITHM

Walid Hariri[1,*] and Mohamed Abdelbassat Boumezaid[2]

Abstract. Recognizing weather patterns plays a vital role in numerous areas of daily life, including weather forecasting, transportation, agriculture, and forest management. Machine learning approaches, particularly Convolutional Neural Networks (CNNs), offer improved weather pattern analysis compared to traditional radar systems. However, CNNs struggle to capture multi-level dependencies without increasing model complexity. Recently, Vision Transformers (ViT), adapted from Natural Language Processing, have demonstrated superior performance in capturing global image relationships. This study presents a deep learning approach to accurately detect and classify weather conditions into multiple categories through transfer learning. To tackle the challenge of selecting the optimal number of trainable layers in CNNs and blocks in ViTs, an Adaptive Layer Freezing (ALF) technique is introduced, dynamically modifying the trainable layers to maximize efficiency during transfer learning with the help of Bayesian Optimization (BO). BO, widely recognized for its capability in hyperparameter tuning, ensures the ALF process operates with optimal configurations, enhancing both training efficiency and model accuracy. We compare BO with Genetic Algorithms (GA), a robust metaheuristic inspired by natural selection. Experimental results show that GA achieves slightly lower accuracy than BO. A new architecture based on ViT-L32 is evaluated against two CNN models – EfficientNetB0 and MobileNetV2 – using two weather imaging datasets (WEAPD and WCD), with ViT-L32 achieving the best classification results, attaining 98.28% accuracy for binary and 96.45% for multi-class classifications.

Mathematics Subject Classification. 68T07, 68T20, 68T45.

Received January 30, 2025. Accepted May 8, 2026.

1. Introduction

The ability to recognize weather patterns plays a crucial role in a wide range of applications, such as weather forecasting, monitoring road conditions, planning transportation logistics, managing agricultural practices, and supporting forest conservation efforts. Additionally, autonomous vehicles can utilize advanced deep learning algorithms to accurately detect and interpret outdoor weather conditions, enabling them to make necessary adjustments in real-time to navigate and adapt to rapidly changing environmental factors [1]. Despite its widespread

Keywords. Weather detection, vision transformer, Bayesian optimization, genetic algorithm, CNN.

[1] LABGED Laboratory, Computer Science Department, Badji Mokhtar-Annaba University, P.O. Box 12, Annaba 23000, Algeria.

[2] Computer Science Department, Badji Mokhtar-Annaba University, P.O. Box 12, Annaba 23000, Algeria.

[*]Corresponding author: walid.hariri@univ-annaba.dz

use, radar-based weather detection has notable limitations. The following points highlight the main limitations and challenges.

- **Line-of-sight limitations:** Radar weather detection relies on a direct line-of-sight between the radar antenna and the object being detected, so it cannot detect objects that are obstructed by terrain, buildings, or other structures.
- **Range limitations:** The range of radar weather detection is limited by the transmission power of the radar and the sensitivity of the receiver. The effective range of the radar decreases with increasing altitude, so it may not be able to detect high-altitude weather patterns.
- **False returns:** Ground clutter, such as buildings and trees, can create false returns on radar images, making it difficult to distinguish between actual weather patterns and false echoes.
- **Difficulty in detecting light precipitation:** Light precipitation, such as drizzle or light rain, can be difficult to detect with radar due to its low reflectivity.
- **Limitations in detecting small-scale weather patterns:** Small-scale weather patterns, such as thunderstorms, can be difficult to detect with radar due to its relatively coarse resolution.
- **Interference from other sources:** Other sources of electromagnetic interference, such as cell phone towers and other radar systems, can cause interference and affect the accuracy of radar weather detection.

Despite these limitations, radar weather detection remains an important tool for meteorologists and weather forecasters, providing valuable information about the location, intensity, and movement of precipitation and other weather patterns [2]. However, real-time weather condition classification using radar data is still very challenging compared to using digital cameras, which enable the application of many innovative machine learning techniques.

To overcome the challenges in accurately identifying weather patterns, several approaches are utilized. Cutting-edge imaging and signal processing technologies are used to detect various atmospheric conditions, such as clouds, precipitation, and fog. For example, clutter removal-based weather observation has been introduced in [3] using signal preprocessing and pulse integration in order to obtain an accurate rain observation. Furthermore, satellite images, radar measurements, and numerical models play a crucial role in analyzing and recognizing weather patterns [4]. It finds its success in many applications such as maritime surveillance systems [5], driver assistance system [6], and object detection for autonomous vehicles [7].

Deep learning methods can be employed to identify weather conditions by analyzing and interpreting data from multiple sources, including satellite imagery, radar data, and observations from weather stations [8, 9]. Integrating these diverse datasets with deep learning algorithms enables the precise prediction and classification of various weather phenomena. To support this process, numerous labeled weather datasets have been introduced in the literature. Leveraging these resources in combination with deep learning techniques facilitates the accurate detection of different weather conditions [10].

Nevertheless, like any technology, the application of deep learning for weather detection has its challenges. For instance, deep learning models may exhibit biases depending on the training data, and they often struggle to fully represent the intricate physical processes within the atmosphere. Furthermore, these models are computationally demanding, requiring substantial computational resources for both training and execution. Despite these challenges, deep learning, particularly transfer learning, remains a promising research domain in weather detection. It offers the potential to enhance weather prediction and understanding significantly, leveraging camera-based imagery as an alternative to traditional radar-based methods.

The remainder of this paper is structured as follows: Section 2 reviews related work, while Section 3 details the proposed methodology. Section 4 describes the optimization of trainable layers, followed by the presentation of experimental results in Section 5. Finally, Section 6 concludes the paper.

2. Related works

Recent advances in deep learning, particularly CNNs and vision transformers, have significantly enhanced the ability to automatically recognize weather conditions from visual data. These methods have been increasingly applied across various domains, including environmental monitoring, smart cities, and autonomous vehicles. This section reviews previous studies that have explored image-based weather detection techniques, with an emphasis on CNN-based transfer learning, vision transformer methodologies, and their applications.

CNN-based weather detection with transfer learning: CNNs have proven effective in extracting hierarchical features from visual data, making them well-suited for weather detection tasks. Transfer learning, wherein pre-trained models are fine-tuned for specific tasks, has been particularly impactful in weather classification due to the limited availability of large-scale annotated datasets.

Naufal *et al.* [11] explored transfer learning by fine-tuning pre-trained CNN models, including MobileNetV2, VGG16, DenseNet201, and Xception, to classify six weather conditions: cloudy, rainy, shine, sunrise, snowy, and foggy. Their experiments utilized images collected from two distinct databases, demonstrating the effectiveness of pre-trained models in achieving high classification accuracy with minimal training data. Similarly, Pillai *et al.* [12] investigated the use of EfficientNetB4 on a dataset comprising 6,862 images annotated across 11 distinct classes. While the model demonstrated promising results, its ability to extract highly discriminative features was limited when dealing with challenging weather conditions. This limitation highlights the need for further enhancements, such as incorporating weather-robust feature extraction techniques or leveraging additional preprocessing methods to improve model performance under adverse environmental scenarios.

Further advancements have incorporated domain-specific augmentation techniques and feature fusion strategies. For instance, Minhas *et al.* [13] proposed a custom-built driver simulator designed to simulate a wide range of weather conditions, addressing the challenges of weather classification from real-world images using neural networks. Given the high variability in weather representation across existing datasets, the simulator generates a synthetic dataset that enables more controlled and consistent evaluation. Additionally, the performance of this synthetic dataset was assessed to demonstrate its potential for improving weather classification tasks.

Vision Transformer-based weather detection: ViTs have emerged as a powerful alternative to traditional CNNs, leveraging self-attention mechanisms to effectively capture long-range dependencies and global contextual information [14]. Unlike CNNs, which rely on localized feature extraction, ViTs process entire image patches, enabling them to identify subtle environmental cues and complex contextual relationships crucial for weather detection tasks. This capability makes them particularly well-suited for scenarios where weather conditions exhibit nuanced variations, such as changes in cloud patterns, lighting conditions, or atmospheric haze.

Recent studies have demonstrated the potential of ViTs in addressing the challenges of weather classification, such as dealing with high intra-class variability and distinguishing between visually similar weather conditions. Furthermore, their adaptability to large-scale datasets and compatibility with pre-training on diverse image corpora have further enhanced their robustness and accuracy. As a result, ViTs have become a prominent choice for weather detection, offering superior performance and flexibility compared to traditional deep learning approaches.

Samo *et al.* [15] utilized a vision transformer framework with an adaptive attention mechanism to detect road irregularities and weather conditions. Their method dynamically adjusted pixel-based attention weights, achieving state-of-the-art performance on multiple weather datasets. The use of adaptive attention allowed the model to focus on regions of interest, such as cloudy skies or wet road surfaces, enhancing overall detection accuracy.

Hajri *et al.* [16] extended the application of vision transformers to road accident detection, integrating weather condition recognition into their pipeline. Their multi-head attention mechanism effectively captured complex interactions between weather patterns and traffic anomalies, demonstrating the versatility of ViTs in safety-critical scenarios.

Hybrid architectures: Recent studies have also explored hybrid architectures combining CNNs and vision transformers to leverage the strengths of both paradigms. Chen *et al.* [17] proposed a hybrid CNN-ViT model for weather detection, where CNN layers captured local features while transformer layers modeled global dependencies. This approach achieved superior performance on benchmark datasets compared to standalone CNN or ViT models. The MASK mechanism is employed in this work to deal with the impact of illumination and feature regions on classification performance. Despite the efficiency of this strategy, there are still challenges in optimizing the interaction between the CNN and transformer components. These challenges include balancing computational complexity and ensuring effective feature fusion. These hybrid architectures represent promising directions for improving weather detection systems, particularly in scenarios requiring both local feature sensitivity and global contextual understanding. However, further research is needed to address limitations such as computational efficiency and scalability for real-world applications.

Recent research has explored hybrid architectures incorporating multi-modal fusion to enhance weather detection and environment perception. Zhang *et al.* [18] introduced a framework combining CNNs and vision transformers for robust performance in diverse weather. Similarly, the "TransFusion" model leverages LiDAR and Radar fusion to address foggy weather challenges, as Radar is unaffected by fog. Its architecture includes a Multimodal Rotate Region Proposal Network and Multi-modal Refine Network, enhanced by a Spatial Vision Transformer (SVT) and Cross-Modal Attention Mechanism for feature extraction and fusion. TransFusion achieves challenging results in foggy conditions on the Oxford Radar RobotCar dataset. In addition, Li *et al.* [19] proposed a dual-augmented attention mechanism that combines convolutional self-attention with atrous self-attention modules to capture both low-level and high-level semantic representations from images. These feature vectors are then concatenated and passed through a linear layer to classify weather types effectively.

However, challenges remain, such as the high computational cost of multi-modal fusion and ensuring real-time performance in practical applications. Additionally, effectively integrating data from multiple sensors like LiDAR, Radar, and cameras is inherently complex due to differences in data resolution, noise levels, and temporal synchronization. Ensuring seamless fusion while maintaining robustness across varying weather conditions and sensor reliability adds another layer of difficulty, particularly for deployment in dynamic and resource-constrained environments.

Hyperparameter optimization: Several optimization techniques, including Bayesian Optimization (BO), Genetic Algorithms (GA), and Particle Swarm Optimization (PSO), have been widely adopted to address the complexity of hyperparameter tuning in deep learning models. In the context of weather forecasting and automatic weather detection, these approaches have been effectively applied to optimize model parameters, select relevant meteorological features, and improve predictive performance. Evolutionary and swarm-based methods such as GA and PSO, inspired respectively by natural selection and collective social behavior, are particularly suitable for global search in complex and non-linear environments. For instance, GA has been used for feature selection and atmospheric parameter optimization [20–22], while PSO has been applied to refine model parameters and enhance forecasting accuracy in weather systems [23]. In contrast, BO [24,25] offers a more principled and sample-efficient strategy by constructing a probabilistic surrogate model of the objective function. This makes BO particularly well-suited for computationally expensive deep learning models commonly used in weather analysis, where each evaluation involves significant training cost. As a result, BO has gained increasing attention for optimizing deep architectures in such domains. In this work, we primarily rely on Bayesian Optimization to determine the optimal number of trainable layers, enabling an effective balance between leveraging pre-trained representations and task-specific fine-tuning for weather classification. Additionally, Genetic Algorithms are used for comparison as a representative evolutionary approach widely applied in hyperparameter optimization. GA is particularly suitable for discrete search spaces, such as layer selection, whereas PSO is more naturally designed for continuous optimization and typically requires discretization to handle integer parameters. Therefore, GA provides a more direct and fair baseline for comparison with Bayesian Optimization in this setting.

3. The proposed method

3.1. Motivation and contribution

Convolutional Neural Networks (CNNs) have achieved remarkable success in various image-related tasks, including water detection. However, a significant limitation of CNNs when applied to 2D images is their inherent difficulty in capturing long-range dependencies. These dependencies are crucial for understanding the intricate interplay of visual features across diverse weather conditions, such as the presence of clouds, fog, rain, or sunshine. This limitation has fueled the growing interest in Vision Transformer architectures [26], which excel at modeling long-range dependencies by dividing images into patches and processing them through a sequence of self-attention layers. ViTs have demonstrated superior performance in numerous image classification tasks, including glaucoma detection [27], COVID-19 diagnosis [28], and skin lesion analysis [29, 30].

This study investigates the efficacy of a novel ViT architecture specifically designed for automatic weather detection. We conduct a comparative analysis of its performance against state-of-the-art CNN models. To address the critical challenge of determining the optimal number of layers to freeze during transfer learning, we introduce an Adaptive Layer Freezing (ALF) strategy. This approach leverages techniques rooted in operations research, such as Bayesian Optimization [31], and Genetic Algorithms to systematically identify the optimal number of trainable layers or blocks within both CNNs and ViTs. By adaptively tuning the transfer learning process, ALF enables us to fine-tune the models more effectively for the specific weather classification task, leading to improved accuracy and generalization.

Our contributions can be summarized as follows:

(1) **Development of a novel ViT architecture:** We propose a specialized ViT architecture tailored for efficient and accurate weather detection, leveraging the strengths of ViTs in capturing long-range dependencies.
(2) **Adaptive Layer Freezing (ALF) strategy:** We introduce an ALF strategy that employs Bayesian Optimization to determine the optimal number of trainable layers during transfer learning, enhancing the performance of both CNNs and ViTs for weather classification.
(3) **Comprehensive comparative analysis:** We conduct a thorough evaluation of our proposed ViT architecture and compare its performance against established CNN models. In addition, we compare Bayesian Optimization with Genetic Algorithm for trainable layers optimization, providing valuable insights into the effectiveness of each approach for automatic weather detection.

It is important to clarify the interplay between gradient-based optimization and Bayesian Optimization. While both mechanisms ultimately aim to maximize weather detection accuracy, they operate at distinct levels of the optimization hierarchy: the Vision Transformer's gradient-based learning (*i.e.*, Adam) optimizes the internal model weights to minimize training loss, whereas BO functions at a meta-level to optimize the structural hyperparameters. Specifically, BO determines the optimal number of trainable layers within our Adaptive Layer Freezing strategy, while gradient descent handles the fine-tuning of the weights within those selected layers. Consequently, these processes are complementary rather than redundant, with BO identifying the most effective architecture configuration and gradient-based optimization learning the parameters within that configuration.

3.2. Method overview

In this work, we propose an enhanced Vision Transformer (ViT) model for weather condition classification, fine-tuned using adaptive transfer learning techniques. Our approach builds upon the baseline ViT-L32 architecture, incorporating advanced hyperparameter tuning to optimize model performance for both binary and multi-class classification scenarios. The proposed method is designed to leverage the inherent strengths of Vision Transformers, particularly their robustness to translation and scale variations, making them well-suited for complex weather classification tasks.

To evaluate the effectiveness of the proposed model, we compare its performance against two top-performing convolutional neural networks (CNNs): EfficientNetB0 and MobileNetV2. While EfficientNetB0 is known for its high accuracy and strong performance on diverse classification tasks, MobileNetV2 offers a lightweight alternative for resource-constrained applications. These comparisons highlight the advantages of the ViT model, particularly in scenarios with smaller target datasets where adaptive transfer learning significantly boosts accuracy.

By combining the efficiency of Vision Transformers with the adaptability of transfer learning, this approach provides a robust solution for accurate weather condition classification. The study also examines how ViTs outperform traditional CNNs in handling variations in weather patterns, such as differences in scale, perspective,

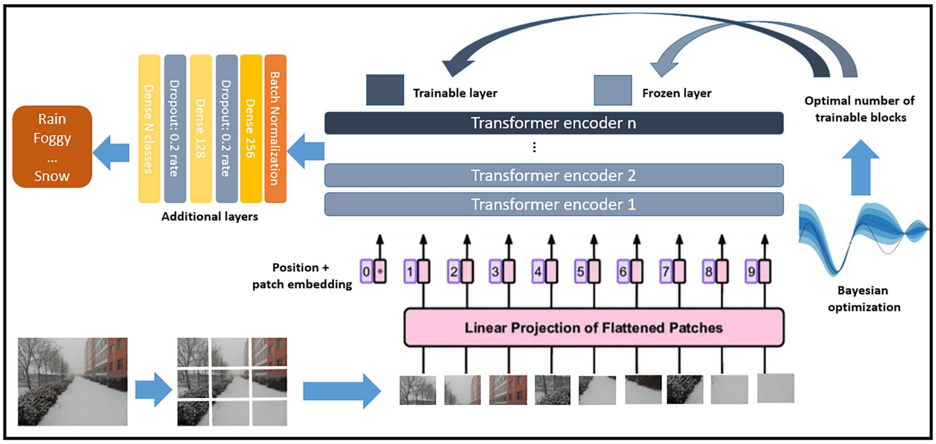


FIGURE 1. Enhanced vision transformer for weather detection.

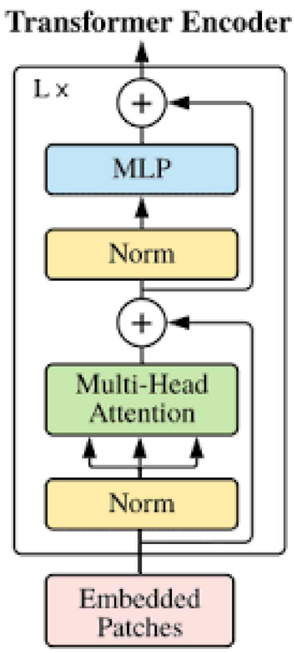


FIGURE 2. Transformer encoder.

and environmental noise, which often pose challenges for conventional methods. Adaptive Layer Freezing is applied to determine the optimal number of trainable layers during the transfer learning process to enhance the classification performance using Bayesian optimization [32]. Figure 1 presents the overview of the proposed method.

3.3. ViT description

Vision Transformers [26] are deep learning models that modify the attention-based transformer framework, which has been widely adopted in natural language processing [33], to enable its application in image classification and pattern recognition, particularly for automatic weather classification. As a result, ViT models have become strong competitors to CNNs in image classification tasks, achieving superior performance. This enhancement in performance can be attributed to the ability of ViTs to capture long-range dependencies effectively through self-attention mechanisms. Consequently, when evaluated on the ImageNet-21k dataset, ViT models outperformed CNN baselines, improving the accuracy from 87.54% to 88.55%. The transformer model takes a sequence of one-dimensional inputs. To process two-dimensional images, we divide an image $x \in \mathbb{R}^{S_1 \times S_2 \times C_h}$ into N patches of fixed size $\boldsymbol{x}_p^{(i)} \in \mathbb{R}^{N \times (P \times P \times C_h)}$ where:

– $i \in \{1, \ldots, N\}$ indicates the position of each patch.
– $S_1 \times S_2$ denotes the image dimensions.
– C_h represents the number of channels (*e.g.*, $C_h = 3$ for RGB images).
– $P \times P$ defines the spatial resolution of each patch (*e.g.*, 32×32 in the case of ViT-L32).
– The total number of patches, N, is computed as: $N = \frac{S_1 \times S_2}{P \times P}$.

The subsequent step involves the linear projection of the flattened patches. Initially, these patches are flattened and then mapped to a D-dimensional space through a trainable linear projection layer, as shown in equation (1). This process produces patch embeddings as output. To this sequence of embedded patches, we append a learnable embedding, denoted as $Z_0^0 = x_{\text{class}}$, similar to the [class] token used in BERT. In order to preserve the positional information, position embeddings are incorporated into each patch embedding.

$$\mathbf{z}_0 = \left[\boldsymbol{x}_{\text{class}}\,; \boldsymbol{x}_p^1\mathbf{E}; \boldsymbol{x}_p^2\mathbf{E}; \boldsymbol{x}_p^3\mathbf{E}; \ldots; \boldsymbol{x}_p^N\mathbf{E}\right] + \mathbf{E}_{\text{postion}} \tag{1}$$

where, $\quad \mathbf{E} \in \mathbb{R}^{(P \times P \times C_h) \times D}, \mathbf{E}_{postion} \in \mathbb{R}^{(N+1) \times D}$.

Finally, the resulting sequence is fed to the transformer encoder block, where each block contains three different layers:

– **Layer normalization (LN)**: it is added before each block because there are disabled new dependencies between the training images. Accordingly, the training time and overall performance are improved.
– **Multi-Head Self-Attention Layer (MHSA)**: this layer plays a crucial role as the central component of the transformer block. It consists of h self-attention mechanisms, each referred to as a "HEAD." The self-attention mechanism takes three input vectors: the value vector with a dimension of d_v, along with the query and key vectors, each having a dimension of d_k. The output matrix is computed as follows: first, the dot product of the query with all keys is calculated, then each result is divided by $\sqrt{d_k}$, and finally, the softmax function is applied to generate the corresponding weights for the values (Fig. 2). The Self-Attention function is mathematically expressed in equation (2).

$$\mathrm{SA}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{2}$$

MHSA consists of concatenating h self-attention operations in parallel. The obtained result is then projected linearly to achieve the final output. The MHSA function is defined in equation (3):

$$\mathrm{MHSA}(Q, K, V) = \mathrm{Concat}(\mathrm{SA}_1, \ldots \mathrm{SA}_h) \times W^O \tag{3}$$

where, Q, K, V form a matrix of set of: queries, keys and values respectively and W^O is a projection matrix that allows concatenating the self-attention outputs.
– **Multi-Layer Perceptrons Layer (MLP)**: This layer is made up of two layers, each with a Gaussian Error Linear Unit (GELU).

In this paper, we conduct our experiments using ViT-L32, compared to two CNN baselines (MobileNetV2 and EfficientNETB0) for automatic weather classification. ViT-L32 is a deeper variant of ViT models belongs to ViT-Large family and contains 24 blocks with a total of 307 million parameters. As shown in Figure 1, the following additional layers are used: Batch Normalization, Dense 256, Dense 128, and Dropout (0.2 or 0.3). The size of the final Dense layer depends on the number of output classes corresponding to the weather conditions. These added layers aim to adapt the model to the output classes and help prevent overfitting.

3.4. MobileNetV2 fine-tuning

MobileNetV2 [34] enhances MobileNet's performance using inverted residuals and linear bottlenecks. Inverted residuals improve the efficiency of ReLU activations and help retain more information post-activation. Crucially, the final activation within each bottleneck must be linear to maintain this preserved information. To train the model, we froze several layers while keeping the rest trainable. For the classification task, we added a 2D convolutional layer with a kernel size of 5, a dropout layer with a rate of 0.2, and a global average pooling layer to avoid overfitting. The output was adjusted according to the number of output classes using a dense layer followed by a softmax activation function. We applied ALF to optimize the number of trainable layers during the transfer learning process.

3.5. EfficientNetB0 fine-tuning

The EfficientNetB0 model [35] introduces a new scaling method that uniformly scales all dimensions of depth, width, and resolution using a compound coefficient. EfficientNetB0 achieves state-of-the-art accuracy while being more computationally efficient. The model's design incorporates a combination of depthwise separable convolutions and efficient swish activations, optimizing the network for both speed and accuracy. Fine-tuning EfficientNetB0 involves adjusting these parameters to leverage its balance between efficiency and performance, allowing for superior accuracy with reduced computational overhead. We applied the same configuration of MobileNetV2 to train the model, experimenting with different numbers of trainable layers using ALF optimization.

Softmax is an activation function, not a loss, it results into a vector in the range of [0, 1], the sum of all values is equal to 1, where each value represents a class probability.

After adding the class weight method through the Categorical Cross-Entropy loss, equation (2) is updated and it is given by equation (1):

$$\mathrm{CCE} = -\sum_{i=1}^{C} W_i . t_i \log(f(s)_i). \tag{4}$$

Where W_i represents the weight of each class i.

4. Optimization of trainable layers

This section details the theoretical foundations of Bayesian Optimization and Genetic Algorithms within the context of model fine-tuning. We describe how these optimization strategies – combining a model-based approach (Bayesian Optimization) and a metaheuristic method (Genetic Algorithms) – navigate the discrete search space of trainable layers to identify configurations that optimize the trade-off between predictive accuracy and computational efficiency.

4.1. Bayesian optimization applied to ViT-L32

Bayesian Optimization (BO) [31] is a sequential optimization technique designed for problems where evaluating the objective function is expensive. It has been successfully applied to adaptive experimental design [36], multi-objective [37], and multi-task [38] optimization. BO uses a surrogate model (*e.g.*, a Gaussian Process) to approximate the relationship between inputs (*e.g.*, number of trainable layers in ViT-L32) and outputs (*e.g.*, validation accuracy). Rather than exhaustively testing all configurations, BO selects promising candidates to evaluate based on an acquisition function, balancing *exploration* (trying new configurations) and *exploitation* (refining known high-performance configurations).

For the ViT-L32 model, which is computationally expensive to fine-tune due to its large number of parameters and multiple transformer layers, BO offers several advantages over exhaustive search methods. Exhaustive grid or random search across all possible configurations (*e.g.*, training 1, 2, ..., 24 blocks) is inefficient. BO reduces the number of configurations to test by focusing on the most promising candidates, and dynamically adjusts the search space based on observed performance. This approach leads to higher performance and efficiency in finding near-optimal configurations with fewer evaluations.

The steps involved in applying Bayesian Optimization to ViT-L32 are as follows:

– **Step 1: Define the optimization problem:**
  - Input Variable: Number of trainable layers (x), ranging from 1 to 24 (since ViT-L32 has 24 transformer blocks).
  - Objective Function: Validation accuracy $f(x)$ after fine-tuning the model with x trainable layers.
  - Constraints: Training time or computational budget.

  The optimization problem can be formally expressed as:

$$x^* = \arg\max_{x \in \mathcal{X}} f(x), \tag{5}$$

  where $\mathcal{X} = \{1, 2, \ldots, 24\}$ represents the search space.

– **Step 2: Initialize the surrogate model:**
  We model the objective function using a Gaussian Process (GP), where the *prior* over $f(x)$ is given by:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')). \tag{6}$$

  Here, $m(x)$ is the mean function, often set to zero, and $k(x, x')$ is the covariance function (*e.g.*, the squared exponential kernel):

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{(x - x')^2}{2l^2}\right), \tag{7}$$

  where σ_f^2 is the signal variance and l is the length scale.

– **Step 3: Evaluate initial configurations:**
  We select an initial set of points $\{x_1, x_2, \ldots, x_n\}$ and evaluate $f(x)$ at these points to initialize the surrogate model.

– **Step 4: Use the acquisition function:**
  The next configuration x_{t+1} to evaluate is chosen using an acquisition function $\alpha(x)$. Common choices include:
  - Expected Improvement (EI):

$$\alpha_{\mathrm{EI}}(x) = \mathbb{E}[\max(f(x) - f(x^+), 0)], \tag{8}$$

    where $f(x^+)$ is the best observed accuracy so far.
  - Upper Confidence Bound (UCB):

$$\alpha_{\mathrm{UCB}}(x) = \mu(x) + \kappa\sigma(x), \tag{9}$$

    where $\mu(x)$ and $\sigma(x)$ are the mean and standard deviation of the GP *posterior*, and κ controls exploration-exploitation trade-off.
  - Probability of Improvement (PI):

$$\alpha_{\mathrm{PI}}(x) = \Phi\left(\frac{\mu(x) - f(x^+)}{\sigma(x)}\right), \tag{10}$$

    where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution.

– **Step 5: Train and evaluate the model:**
  The model is fine-tuned using the selected number of trainable layers x_{t+1}, and the validation accuracy $f(x_{t+1})$ is measured. The surrogate model is updated accordingly.

– **Step 6: Repeat:**
  The process iterates by selecting new configurations using the updated surrogate model and acquisition function until the computational budget is exhausted or accuracy converges.

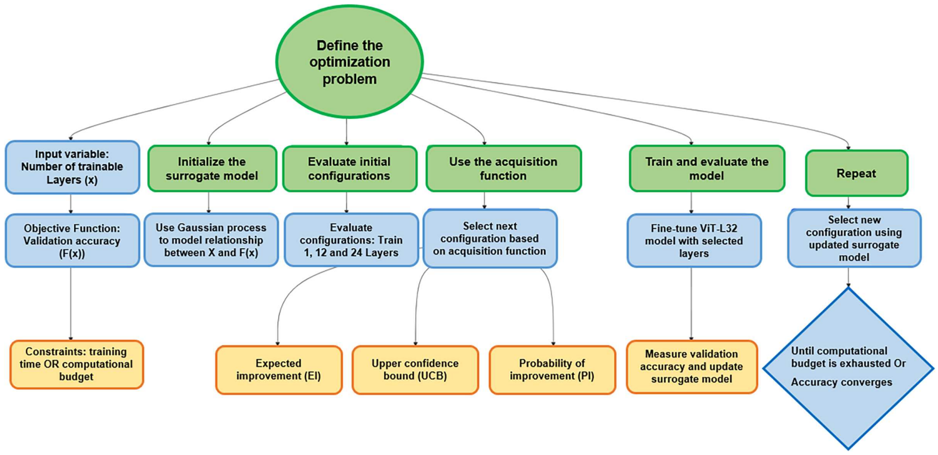


FIGURE 3. Flowchart for Bayesian optimization applied to ViT-L32 fine-tuning for weather classification.

For instance, in a step-by-step application:

(1) **Initialization:** Fine-tune ViT-L32 with 1, 12, and 24 trainable layers. Record validation accuracy: $f(1) = 78\%$, $f(12) = 82\%$, $f(24) = 80\%$.
(2) **Surrogate model:** Fit a Gaussian Process to these points. The GP predicts accuracy for untested configurations (*e.g.*, 6 or 18 trainable layers) with associated uncertainty.
(3) **Acquisition function:** EI selects the next configuration (*e.g.*, 18 trainable layers) as it has the highest expected improvement.
(4) **Iteration:** Fine-tune with 18 trainable layers, achieve $f(18) = 83\%$, and update the surrogate model.
(5) Repeat the process until a stopping criterion is met.

The benefits of BO for ViT-L32 include:

- **Computational efficiency:** BO reduces the number of configurations tested, saving computational time compared to grid or random search.
- **Flexibility:** It incorporates constraints like limited training time or acceptable computational cost.
- **Performance gains:** BO identifies high-performing configurations by leveraging the surrogate model.
- **Uncertainty management:** BO quantifies uncertainty, enabling it to explore untested configurations systematically.

Bayesian Optimization is an efficient technique for fine-tuning large models like ViT-L32. By intelligently selecting the number of trainable layers, BO balances computational cost with the goal of maximizing validation accuracy. Experiments show that BO often converges to the optimal configuration (*e.g.*, 18 trainable layers) with significantly fewer evaluations compared to exhaustive search methods, making it particularly valuable for large-scale models and resource-constrained environments. Figure 3 presents the flowchart that summarizes the Bayesian Optimization process applied to fine-tune ViT-L32.

4.2. Genetic algorithm applied to ViT-L32

Genetic Algorithm [39] is a population-based metaheuristic inspired by the principles of natural selection and genetics. Unlike Bayesian Optimization, which relies on a sequential surrogate model to guide the search, GA explores the search space through a population of candidate solutions evolving over successive generations. This population-based strategy enables parallel exploration and makes GA particularly suitable for non-linear and discrete optimization problems, such as selecting the number of trainable layers in deep architectures. In the context of ViT-L32 fine-tuning for automatic weather detection, GA is employed to optimize the number of trainable transformer blocks within a transfer learning framework. This parameter directly impacts the trade-off between leveraging pre-trained knowledge and adapting the model to domain-specific features. The implementation of GA follows the steps below:

- **Step 1: Problem encoding:** Each candidate solution is encoded as a chromosome $x \in \{1, 2, \ldots, 24\}$, where x represents the number of trainable transformer blocks in ViT-L32. The objective function $f(x)$ corresponds to the validation accuracy obtained after fine-tuning.
- **Step 2: Initialization:** An initial population $P_0 = \{x_1, x_2, \ldots, x_N\}$ is randomly generated, ensuring diversity in the explored configurations, where each x_i is a random layer count.
- **Step 3: Fitness evaluation:** Each individual is evaluated by fine-tuning the model with x_i trainable layers and computing its validation performance:

$$\text{Fitness}(x_i) = f(x_i). \tag{11}$$

- **Step 4: Selection:** Individuals with higher fitness are more likely to be selected as parents using strategies such as *Tournament Selection* or *Roulette Wheel Selection*, promoting the propagation of high-quality solutions.
- **Step 5: Crossover and mutation:** New offspring are generated using genetic operators:
  - **Crossover:** Combines two parent solutions to produce new candidate configurations.
  - **Mutation:** Randomly altering a layer count with probability p_m, enabling exploration and reducing the risk of premature convergence.
- **Step 6: Iteration and convergence:** The population is iteratively updated over generations until a stopping criterion, based on validation accuracy, is satisfied.

The main advantage of GA lies in its strong global search capability and its ability to escape local optima through stochastic exploration. However, this comes at the cost of higher computational complexity, as multiple candidate solutions must be evaluated at each generation, each requiring full or partial model fine-tuning. Despite this limitation, GA provides a robust baseline for optimizing deep model configurations and serves as a valuable comparison framework against more sample-efficient methods such as Bayesian Optimization.

5. Experimental results

In this section, we present the evaluation protocol, the tested databases, and the results obtained using both binary and multi-class classifications. We applied 10-fold cross-validation to fine-tune the proposed CNN and Vision Transformer models. The Adam optimizer was used with a learning rate of 0.0001 and a batch size of 32. Dropout values of 0.3 and 0.5 were considered as hyperparameters. To regularize the models, early stopping was implemented with a patience of 6 epochs. The total number of epochs was set to 20.

5.1. Databases

Weather Phenomen database (WEAPD): consists of 11 meteorological occurrences and the related categories and quantity of pictures are as follows: hail, snow, lightning, rainbow, rain, dew, sandstorm, rime, frost, glaze and fog/smog (see Fig. 4). All images are in the JPG format [40].

The WEAPD dataset [40], available on Harvard Dataverse, consists of images representing 11 distinct weather conditions. These categories, along with their respective image counts, include snow, hail, lightning, rain, rainbow, dew, frost, rime, glaze, sandstorm, and fog or smog. All images are stored in JPG format. It is designed to be used for training and testing computer algorithms that can automatically detect and classify different types of weather events from images. for a total of over 6,000 images in the dataset. The WEAPD can be used to develop and evaluate computer vision algorithms for weather detection and analysis, which can have

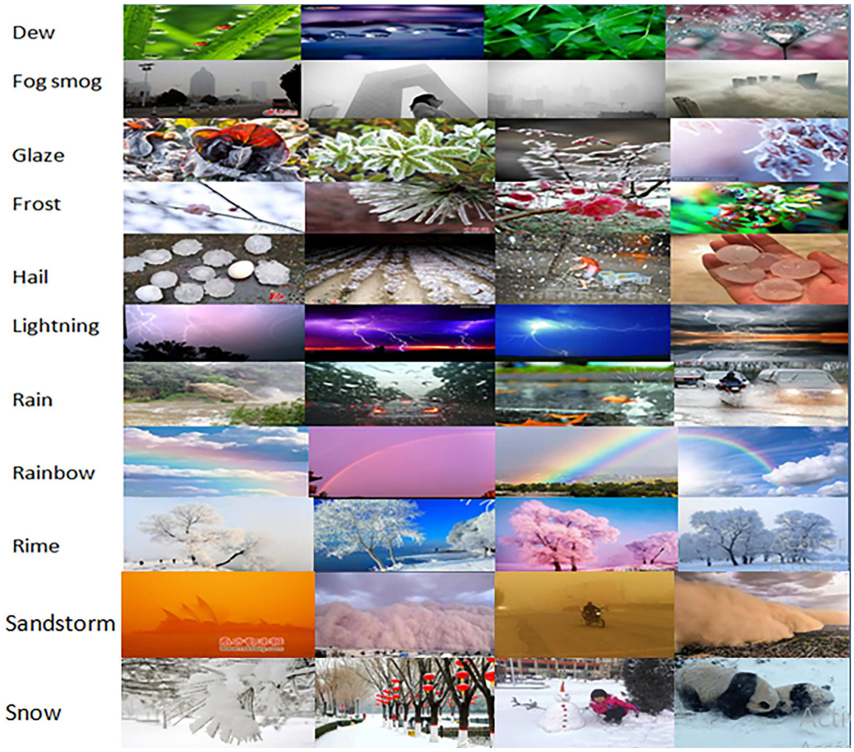


FIGURE 4. Images samples of 11 weather conditions from WEAPD database.

applications in fields such as meteorology, agriculture, transportation, and disaster response. Eleven weather conditions from WEAPD dataset are shown in Figure 4.

Weather Classification Dataset (WCD): is downloaded from Kaggle [41], it consists of 1530 images of 5 different weather conditions: cloudy, rainy, shine, sunrise, and foggy, with a training set of 1080 images and a test set of 450 images. Each RGB image has a resolution of 128x128 pixels. The dataset can be used to train machine learning models for weather classification tasks and computer vision applications. Five weather conditions from WCD are shown in Figure 5.

The Weather Classification dataset [41] downloaded from Kaggle consists of 1530 images of 5 different weather conditions: "cloudy," "rainy," "shine," "sunrise," and "foggy." Each image has a resolution of 128x128 pixels and is in RGB format. The dataset is divided into a training set of 1080 images and a test set of 450 images. The dataset can be used to train machine learning models for weather classification tasks, such as identifying the current weather conditions in a given image. The dataset can also be used for image recognition research, computer vision applications, and other related fields. Five weather conditions from Weather Classification Dataset are shown in Figure 5.

5.2. Binary classification

We evaluated several binary classifications using the WEAPD dataset, including (snow, sandstorm) and (lightning, hail). The confusion matrices shown in Figures 6, 7 and 8 show that the first classification is notably more challenging, achieving accuracies of 95.05% 96.19%, and 98.28% with MobileNetV2, EfficientNetB0 and ViT-L32, respectively.

In contrast, the latter two achieve perfect classification accuracy of 100% (see Fig. 8B). This outcome is attributed to the distinct sharp edges in lightning images and the textural differences in hail and snow images.

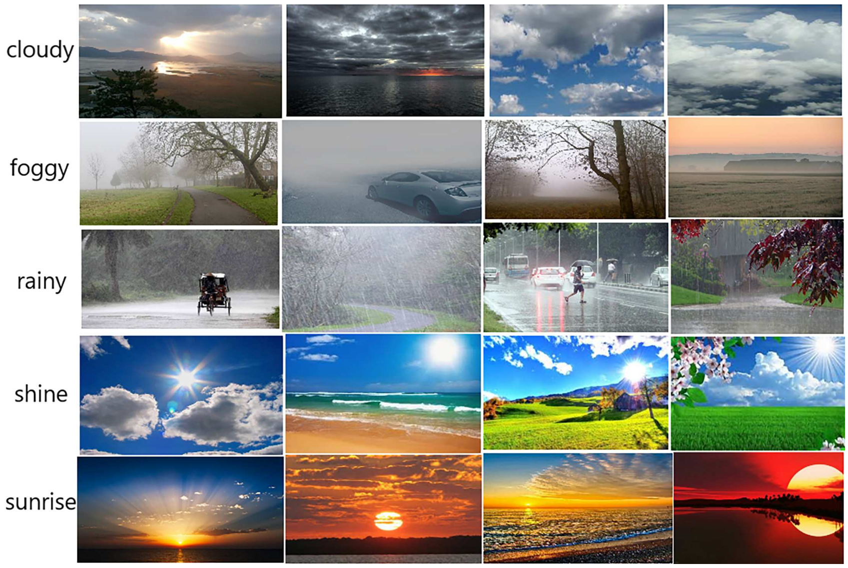


FIGURE 5. Five weather conditions from WCD dataset.

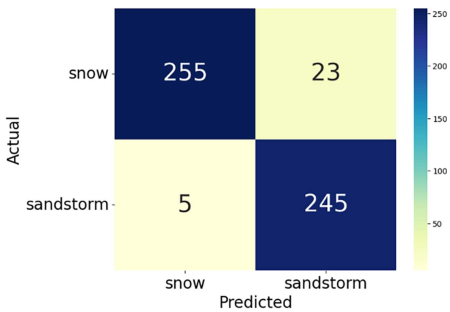


FIGURE 6. Confusion matrix of MobileNetV2 using WEAPD dataset with binary classification: snow, sandstorm.

The high contrast and well-defined features in these categories facilitate their accurate differentiation by the model.

However, the classification of snow and sandstorm proves more challenging due to the similarity in their appearance. Both classes exhibit overlapping visual characteristics, such as low contrast and fine granular textures, which can lead to misclassification (8 snow images and 1 sandstorm image with ViT-L32; 17 snow images and 3 sandstorm images with EfficientNetB0). The presence of swirling patterns and diffused lighting in sandstorm images further contributes to the complexity of distinguishing them from snow-covered landscapes. Additionally, environmental conditions like fog or mist may blur the differences between these two categories, adding another layer of difficulty to the classification task.

5.3. Multi-class classification

In the following, we present the obtained results for multi-class classification using both datasets, leveraging the hyper-parameters detailed above. We analyze the impact of different model architectures, including CNN and ViT, and assess their classification performance across various classes. Furthermore, we examine the influence of

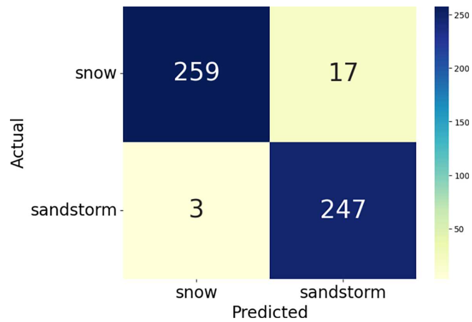
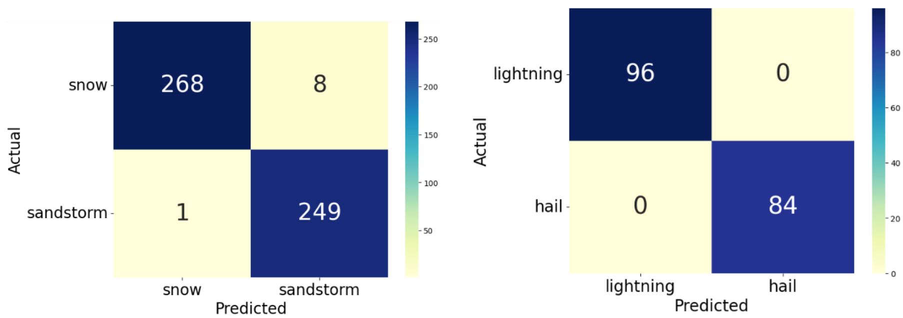


FIGURE 7. Confusion matrix of EfficientNetB0 using WEAPD dataset with binary classification: snow, sandstorm.



(A) Snow, Sandstorm (B) Lightning,Hail

FIGURE 8. Confusion matrix of ViT-L32 using WEAPD dataset with binary classification: snow and sandstorm, lightning and hail.

Bayesian Optimization on improving accuracy and efficiency. The results provide insights into how these models perform under different configurations and highlight the advantages of using BO for enhancing classification performance.

5.3.1. WEAPD dataset

To further investigate the robustness of each model in dealing with various weather conditions, we considered five classes from WEAPD dataset including: hail, glaze, snow, sandstorm, and dew. Figure 9 shows the confusion matrix for the ViT-L32 model. From this figure, it is evident that "snow" and "sandstorm" are classified more accurately followed by "hail", while "dew" shows the lowest classification rate. This discrepancy can be attributed to the indistinct texture of "dew" and "glaze," which makes them prone to confusion with other classes. Using five classes from WEAPD dataset (comprising Hail, Glaze, Snow, Sandstorm, and Dew), where the GA-optimized ViT-L32 yielded slightly lower performance metrics compared to its BO-optimized counterpart (96.20% *vs.* 96.45%).

5.3.2. WCD dataset

This section presents the obtained results when dealing multi-class classification scenarios on WCD dataset using 3 classes: "cloudy", "sunrise", and "shine". Figures 10 and 11 present the confusion matrices of ViT-L32 and EfficientNetB0 with a performance of 94.44% and 96.66% respectively, where "shine" has been perfectly classified compared to the other classes. This result is obtained due to the distinct visual features and patterns

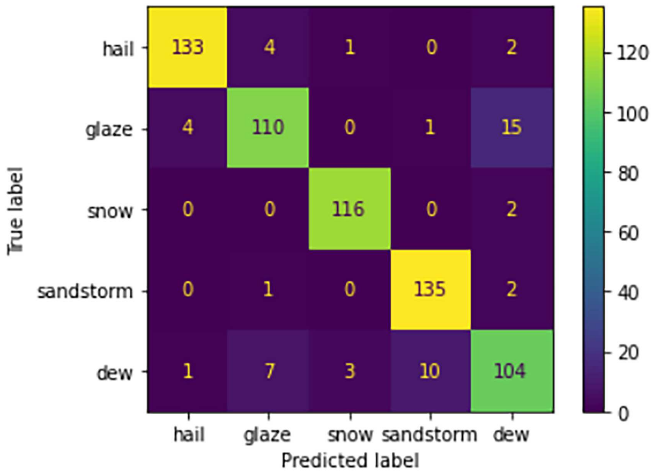


FIGURE 9. Confusion matrix of ViT-L32 using 5 classes.

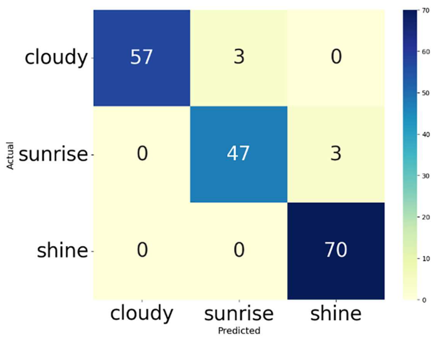


FIGURE 10. Confusion matrix of ViT-L32 using three classes from WCD: cloudy, sunrise, shine.

associated with "shine" that are effectively captured by both ViT-L32 and EfficientNetB0 models. ViT-L32 was superior to EfficientNet due to its capability to process the weather images as sequences of patches, enabling comprehensive context capture through self-attention mechanisms.

Table 1 presents a summary of the results obtained on the WEAPD and WCD datasets for both binary and multi-class classifications. This table also highlights the impact of Bayesian Optimization on the performance of CNN and ViT models.

From the comparison, it is evident that BO enhances the accuracy of all models in both binary and multi-class classification tasks. For instance, in the "snow, sandstorm" classification, ViT-L32 achieved an accuracy of 97.98% without BO, which improved to 98.28% with BO. Similarly, for the "Cloudy, Shine, Sunrise" classes, the accuracy of EfficientNet-B0 increased from 93.66% to 94.44% after applying BO. A significant improvement was also observed for the same model in the classification of "Hail, Glaze, Snow, Sandstorm, and Dew," where accuracy increased from 91.89% to 92.37%. Furthermore, for the challenging five-class classification task (Hail, Glaze, Snow, Sandstorm, and Dew), ViT-L32 achieved 96.45% with BO compared to 96.03% without BO.

These results demonstrate the effectiveness of BO in enhancing model performance while potentially reducing power consumption. Additionally, it is noteworthy that ViT-L32 consistently outperforms CNN models across all settings, further emphasizing its superiority in classification tasks.

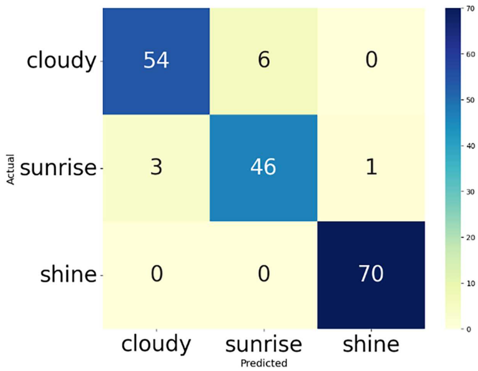


FIGURE 11. Confusion matrix of EfficientNetB0 using three classes from WCD: cloudy, sunrise, shine.

TABLE 1. Performance comparison on WEAPD and WCD. "+BO" indicates that Bayesian Optimization is used, whereas "−BO" signifies that it is not applied to the model.

Dataset	Classes	Model	Accuracy
WEAPD	Hail, Lightening	MobileNetV2+BO	100%
WEAPD	Hail, Snow	EfficientNetB0+BO	100%
WEAPD	Hail, Snow	ViT-L32+B0	100%
WEAPD	Snow, Sandstorm	MobileNetV2+BO	95.05%
WEAPD	Snow, Sandstorm	EfficientNetB0+BO	96.19%
WEAPD	Snow, Sandstorm	ViT-L32-BO	97.98%
WEAPD	Snow, Sandstorm	ViT-L32+BO	**98.28%**
WCD	Cloudy-Shine-Sunrise	ViT-L32-BO	96.05%
WCD	Cloudy-Shine-Sunrise	ViT-L32+BO	**96.66%**
WCD	Cloudy-Shine-Sunrise	EfficientNetB0-BO	93.66%
WCD	Cloudy-Shine-Sunrise	EfficientNetB0+BO	**94.44%**
WCD	Cloudy-Shine-Sunrise	ViT-L32+GA	**96.15%**
WEAPD	Hail, Glaze, Snow, Sandstorm, Dew	EfficientNetB0-BO	91.89%
WEAPD	Hail, Glaze, Snow, Sandstorm, Dew	EfficientNetB0+BO	**92.37%**
WEAPD	Hail, Glaze, Snow, Sandstorm, Dew	ViT-L32-BO	96.03%
WEAPD	Hail, Glaze, Snow, Sandstorm, Dew	ViT-L32+BO	**96.45%**
WEAPD	Hail, Glaze, Snow, Sandstorm, Dew	ViT-L32+GA	**96.20%**

A comparative study with some state-of-the-art methods using the same datasets is presented in Table 2. Our fine-tuned ViT-L32 achieved higher classification This performance gain can be attributed to the inherent advantages of ViTs over CNNs. ViTs leverage self-attention mechanisms, enabling them to capture long-range dependencies more effectively than the local receptive fields of convolutions, further enhanced by adaptive transfer learning provided by the ALF process to find the optimal number of trainable transformer encoder blocks and trainable layers of CNN using the Bayesian optimization. This adaptive approach optimizes the model's capacity for the specific classification task, leading to improved generalization and reduced overfitting.

A similar trend to WEAPD dataset occurs with the WCD dataset using three classes (Cloudy, Shine, and Sunrise), the ViT-L32 model coupled with Genetic Algorithm optimization achieved an accuracy of 96.15%. This is slightly lower than the 96.66% achieved by the same model using Bayesian Optimization, yet it remains

TABLE 2. Performance comparison of the proposed method with the state-of-the art on WEAPD and WCD datasets.

Method	Model	Number of classes	Accuracy
Li *et al.* [19]	VGG-16	5 classes (WEAPD)	86.14%
Li *et al.* [19]	AlexNet	5 classes (WEAPD)	85.44%
Ghaleb *et al.* [42]	CNN +SVM	5 classes (WCD)	94%
Liu *et al.* [43]	CNN	5 classes (WEAPD)	94%
Albelwi [44]	DenseNet-121	3 classes (RFS Dataset)	95.9%
Our	EfficientNetB0+BO	3 classes (WEAPD)	**94.44%**
Our	Vit-L32+BO	Binary (WEAPD)	**100%**
Our	ViT-L32+BO	3 classes (WCD)	**96.66%**
Our	ViT-L32+BO	5 classes (WEAPD)	**96.45%**

superior to the performance of the non-optimized ViT-L32. Notably, the ViT-L32 consistently outperforms the CNN model baselines.

5.4. Optimizing trainable layers using Bayesian optimization

In this section, we optimize the number of trainable layers in each pre-trained model using BO. The optimization process determines the ideal number of layers to fine-tune while freezing the rest, ensuring an optimal balance between model performance and computational efficiency. Only the original ImageNet pre-trained layers are considered, excluding any additional layers introduced earlier. We apply a consistent adaptive layer freezing strategy across all experiments and evaluate the models on the test dataset, presenting the corresponding results. The results presented are obtained either after the model's accuracy has converged or when the computational budget has been exhausted, as illustrated in the Bayesian optimization flowchart in Figure 3. During the optimization process, a probabilistic model is built to estimate the objective function, which in this case is to improve the accuracy of weather classification. Each iteration involves selecting the next point to evaluate based on the trade-off between exploration and exploitation. The model's performance is then assessed, and the results are incorporated to update the probabilistic model. This process continues until either the desired accuracy is achieved or the computational budget for evaluations is fully utilized.

By analyzing the results of the study, we can notice that the optimal number of trainable parameters varies across different models due to their unique architectures. For instance, MobileNetV2 achieves the best result with 44 trainable layers, while EfficientNetB0 attains the highest accuracy with 51 trainable layers (Fig. 14). In the case of ViT-L32, the model outperforms its accuracy with only 22 trainable layers compared to 30, 40, and 60 layers as shown in Figure 13. Fine-tuning ViT-L32 with a small number of trainable blocks adjusts upper layers or the classification head, refining semantic representations and class-specific features. In contrast, fine-tuning with a large number of trainable blocks extends this adjustment to multiple transformer layers or the entire model, adapting lower-level details and reshaping the entire representation hierarchy to fit new weather datasets.

Figure 12 illustrates that the MobileNetV2 architecture requires a greater number of trainable layers to achieve convergence. This is primarily attributed to its lightweight and efficient design, which trades off depth and complexity for speed and lower computational requirements. In contrast, as shown in Figure 13, models with deeper architectures, such as ViT-L32, demand fewer trainable layers to converge due to their inherent ability to capture complex feature hierarchies more effectively. However, despite requiring fewer layers, these deeper architectures can sometimes sacrifice computational efficiency, particularly when dealing with smaller datasets or less complex problems. ViT-L32 continues to outperform CNN-based models such as MobileNetV2 in terms of accuracy. This advantage can be attributed to its self-attention mechanism, which allows it to model long-range dependencies and capture global context more effectively than traditional convolutional layers in

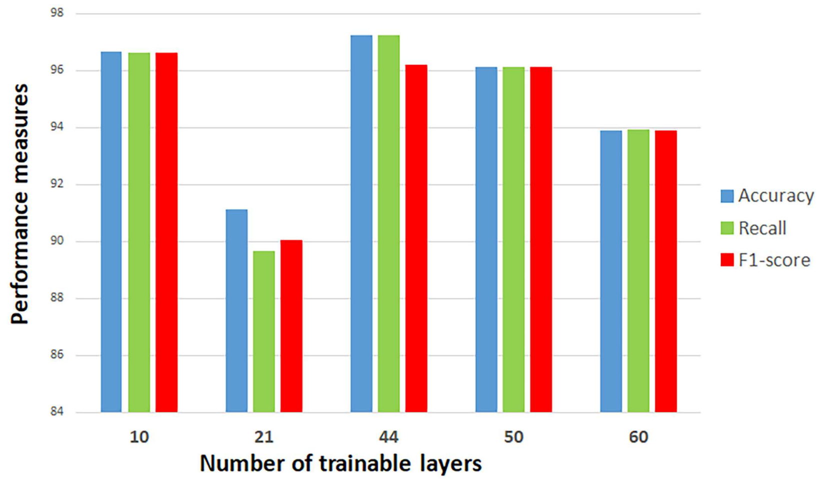


FIGURE 12. Fine-tuning results of three classes "Cloudy, Shine and Sunrise" using MobileNetV2 model.

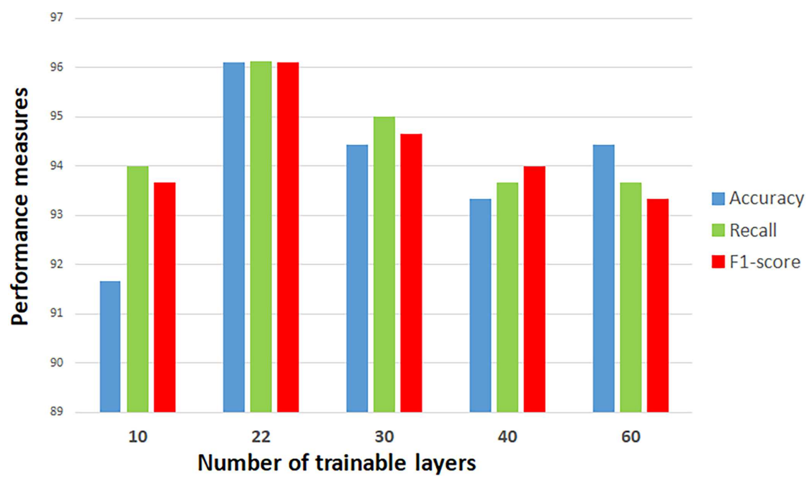


FIGURE 13. Three-class classification (Cloudy, Shine, Sunrise) using ViT-L32 model and BO-based adaptive layer freezing.

CNNs. Self-attention enables ViT-L32 to focus on relevant regions in an input image dynamically, leading to better performance on diverse and complex visual tasks.

Furthermore, the use of Bayesian Optimization has significantly enhanced the efficiency of both ViT and CNN models by fine-tuning hyperparameters in a way that optimizes computational power and reduces time consumption. This optimization process has proven particularly beneficial for models like ViT, where the training process is resource-intensive due to their large parameter space. By identifying optimal configurations, Bayesian Optimization ensures that these models achieve high performance while mitigating computational overhead. This efficiency is particularly beneficial for real-time applications like self-driving cars, where rapid decision-making and minimal latency are crucial. As a result, these optimized models can be deployed in resource-constrained environments, such as edge devices and mobile systems, without compromising accuracy (Fig. 14).

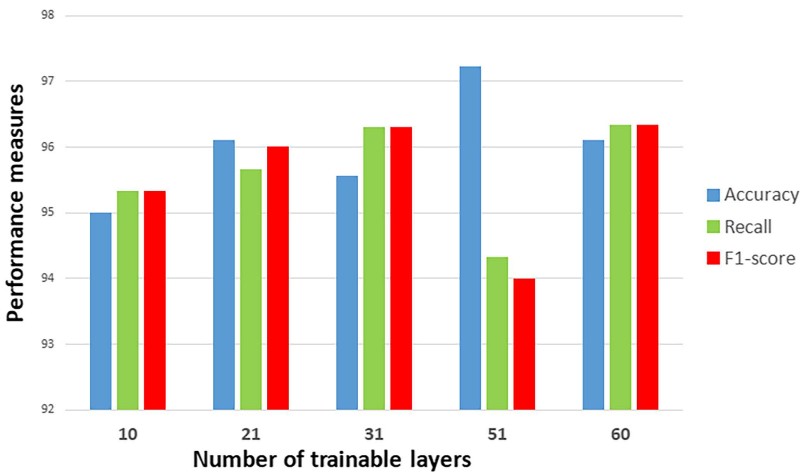


FIGURE 14. Fine-tuning results of three classes "Cloudy, Shine and Sunrise" using EfficientNetB0 model.

5.5. Comparative analysis of optimization strategies

The experimental results obtained from the WCD and WEAPD datasets demonstrate that the ViT-L32 architecture, when combined with metaheuristic optimization for trainable layer selection, consistently outperforms standard CNN baselines for automatic weather detection. This superiority is largely attributed to the self-attention mechanism within the Vision Transformer, which excels at capturing global atmospheric contexts – such as the diffuse lighting of a *Sunrise* or the sweeping textures of a *Sandstorm* – that localized convolutional filters often struggle to encode. A consistent performance trend is observed across both benchmarks. On the WCD dataset, BO achieved 96.66% accuracy compared to 96.15% for the GA. A similar pattern emerged with the more complex 5-class WEAPD dataset (comprising *Hail, Glaze, Snow, Sandstorm, and Dew*), where the BO-optimized counterpart again yielded superior performance metrics over the GA-optimized model (96.45% *vs.* 96.20%). These findings suggest that the high-dimensional search space for layer freezing in large-scale models is more efficiently navigated through probabilistic modeling rather than stochastic evolution.

– **Search precision:** While GA explores the search space through crossover and mutation, it lacks a formal memory of the objective function's landscape. In contrast, BO utilizes a Gaussian Process surrogate model to determine which specific transformer blocks contribute most to feature discriminability. This allows BO to surgically unfreeze layers that capture fine-grained meteorological details, such as the crystalline structure of *Snow* versus the liquid sheen of *Glaze*, while maintaining frozen weights for redundant layers to mitigate overfitting.
– **Convergence and stability:** As complexity increased with the 5-class WEAPD dataset, the performance gap between BO and GA persisted. This indicates that as classification boundaries become more nuanced, the mathematical efficiency of BO in identifying the optimal depth for fine-tuning becomes a critical factor in maintaining high predictive accuracy.

The success of these optimized models confirms that not all layers in a pre-trained Vision Transformer are equally relevant for specialized tasks like meteorological classification. By selectively unfreezing layers, the model adapts its positional embedding and attention heads to the specific spatial frequencies found in weather imagery, ensuring a robust detection performance.

6. Conclusion

In this article, a method for detecting weather conditions using Vision Transformers to capture both global and local dependencies in images has been proposed, enhancing top-performing CNNs. We demonstrated that the fine-tuned ViT-L32 outperforms EfficientNetB0 and MobileNetV2 in binary and multi-class classification on challenging datasets using adaptive transfer learning without overfitting. This superior performance is attributed to ViT's ability to process images as sequences of patches, capturing comprehensive contextual information and modeling complex dependencies through self-attention mechanisms.

Moreover, the introduction of adaptive layer freezing significantly enhanced our models' performance by effectively optimizing the number of trainable layers using Bayesian optimization. Although MobileNetV2 is more computationally efficient due to its lightweight architecture, it requires more trainable layers to converge effectively. On the other hand, ViT-L32 achieves superior accuracy, due to its self-attention mechanism, even with fewer trainable layers. BO integration makes these models more practical by balancing accuracy, resources, and training time, and effectively optimizes pre-trained model hyperparameters within a transfer learning framework.

For comparison purposes, we also evaluated Genetic Algorithms for optimizing the number of trainable layers. GA was selected as a representative population-based evolutionary method due to its widespread use in deep learning optimization and its effectiveness in discrete search spaces. In contrast, PSO, although efficient for continuous optimization problems, is less naturally suited to discrete parameter tuning such as layer selection without additional adaptations. The experimental results show that Bayesian Optimization consistently outperforms GA in terms of convergence efficiency and final predictive performance, confirming its suitability for this task. A similar discipline is expected when dealing with the fine-tuned CNNs. Combining CNN's local features with ViT's holistic view offers a promising path for advancing weather pattern recognition. Expanding data sources and testing scalability on larger datasets can further improve accuracy and real-world applicability.

Data availability statement

The weather classification dataset associated with this article is available on Kaggle at the following URL: https://www.kaggle.com/datasets/vijaygiitk/multiclass-weather-dataset.
The WEAPD dataset associated with this article is available at the following URL: https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/M8JQCR.

References

[1] Q.A. Al-Haija, M. Gharaibeh and A. Odeh, Detection in adverse weather conditions for autonomous vehicles via deep learning. *AI* **3** (2022) 303–317.

[2] D.-H. Paek, S.-H. Kong and K.T. Wijaya, K-radar: 4D radar object detection for autonomous driving in various weather conditions. *Adv. Neural Inf. Process. Syst.* **35** (2022) 3819–3829.

[3] A. Awaludin, T. Sinatra, G.A. Nugroho and F. Nauval, Clutter removal improvement of marine radar for weather observation, in AIP Conference Proceedings, Vol. 2366. AIP Publishing (2021).

[4] K. Hood, S. Torres and R. Palmer, Automatic detection of wind turbine clutter for weather radars. *J. Atmos. Ocean. Technol.* **27** (2010) 1868–1880.

[5] M. Chen, J. Sun, K. Aida and A. Takefusa, Weather-aware object detection method for maritime surveillance systems. *Future Gener. Comput. Syst.* **151** (2024) 111–123.

[6] M.Z. Shahzad, M.A. Hanif and M. Shafique, Robust ADAS: enhancing robustness of machine learning-based advanced driver assistance systems for adverse weather. Preprint arXiv:2407.02581 (2024).

[7] N.U.A. Tahir, Z. Zhang, M. Asim, J. Chen and M. ELAffendi, Object detection in autonomous vehicles under adverse weather: a review of traditional and deep learning approaches. *Algorithms* **17** (2024) 103.

[8] J. Cuomo and V. Chandrasekar, Use of deep learning for weather radar nowcasting. *J. Atmos. Ocean. Technol.* **38** (2021) 1641–1656.

[9] C. Pereira, R.P. Cruz, J.N. Fernandes, J.R. Pinto and J.S. Cardoso, Weather and meteorological optical range classification for autonomous driving. *IEEE Trans. Intell. Vehicles* **9** (2024).

[10] H. Gupta, O. Kotlyar, H. Andreasson and A.J. Lilienthal, Robust object detection in challenging weather conditions, in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (2024) 7523–7532.

[11] M.F. Naufal and S.F. Kusuma, Weather image classification using convolutional neural network with transfer learning, in AIP Conference Proceedings, Vol. 2470. AIP Publishing LLC (2022) 050004.

[12] R. Pillai, N. Sharma and R. Gupta, Fine-tuned EfficientNetB4 transfer learning model for weather classification, in 2023 3rd Asian Conference On Innovation In Technology (ASIANCON). IEEE (2023) 1–6.

[13] S. Minhas, Z. Khanam, S. Ehsan, K. McDonald-Maier and A. Hernández-Sabaté, Weather classification by utilizing synthetic data. *Sensors* **22** (2022) 3193.

[14] S. Khan, M. Naseer, M. Hayat, S.W. Zamir, F.S. Khan and M. Shah, Transformers in vision: a survey, *ACM Comput. Surv. (CSUR)* **54** (2022) 1–41.

[15] M. Samo, J.M. Mafeni Mase and G. Figueredo, Deep learning with attention mechanisms for road weather detection. *Sensors* **23** (2023) 798.

[16] F. Hajri and H. Fradi, Vision transformers for road accident detection from dashboard cameras, in 2022 18th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE (2022) 1–8.

[17] S. Chen, T. Shu, H. Zhao and Y.Y. Tang, Mask-CNN-transformer for real-time multi-label weather recognition. *Knowl. Based Syst.* **278** (2023) 110881.

[18] C. Zhang, H. Wang, Y. Cai, L. Chen and Y. Li, Transfusion: multi-modal robust fusion for 3D object detection in foggy weather based on spatial vision transformer. *IEEE Trans. Intell. Transp. Syst.* **25** (2024).

[19] J. Li and X. Luo, A study of weather-image classification combining VIT and a dual enhanced-attention module. *Electronics* **12** (2023) 1213.

[20] F.A. Shah, A.B.E. Meraj, M. Roman, M. Anwar, A. Zaib, S.S. Shams and F. Hussain, Enhancing weather forecasting accuracy: a machine learning approach using genetic algorithm and random forest. *Glob. Res. J. Nat. Sci. Technol.* **3** (2026).

[21] P. Houtekamer, B. He, D. Jacques, R. McTaggart-Cowan, L. Separovic, P.A. Vaillancourt, A. Zadra and X. Deng, Use of a genetic algorithm to optimize a numerical weather prediction system. *Mon. Weather Rev.* **149** (2021) 1089–1104.

[22] L. Oana and A. Spataru, Use of genetic algorithms in numerical weather prediction, in 2016 18th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC). IEEE (2016) 456–461.

[23] R. Rajkumar, A.J. Albert and D. Chandrakala, Weather forecasting using fuzzy neural network (FNN) and hierarchy particle swarm optimization algorithm (HPSO). *Indian J. Sci. Technol.* **8** (2015) 1.

[24] A.H. Victoria and G. Maragatham, Automatic tuning of hyperparameters using Bayesian optimization. *Evol. Syst.* **12** (2021) 217–223.

[25] H. Cho, Y. Kim, E. Lee, D. Choi, Y. Lee and W. Rhee, Basic enhancement strategies when using Bayesian optimization for hyperparameter tuning of deep neural networks. *IEEE Access* **8** (2020) 52 588–52 608.

[26] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit and N. Houlsby, An image is worth 16x16 words: transformers for image recognition at scale. Preprint arXiv:2010.11929 (2020).

[27] I.-E. Haouli, W. Hariri and H. Seridi-Bouchelaghem, Exploring vision transformers for automated glaucoma disease diagnosis in fundus images, in 2023 International Conference on Decision Aid Sciences and Applications (DASA). IEEE (2023) 520–524.

[28] I. Haouli, W. Hariri and H. Seridi-Bouchelaghem, Covid-attention: efficient covid19 detection using pre-trained deep models based on vision transformers and x-ray images. *Int. J. Artif. Intell. Tools* **32** (2023) 2350046.

[29] A. Mezina and R. Burget, Detection of post-covid-19-related pulmonary diseases in x-ray images using vision transformer-based neural network. *Biomed. Signal Process. Control* **87** (2024) 105380.

[30] G. Yolcu Oztel, Vision transformer and CNN-based skin lesion analysis: classification of monkeypox. *Multimed. Tools Appl.* **83** (2024) 71 909–71 923.

[31] X. Wang, Y. Jin, S. Schmitt and M. Olhofer, Recent advances in Bayesian optimization. *ACM Comput. Surv.* **55** (2023) 1–36.

[32] R. Garnett, Bayesian Optimization. Cambridge University Press (2023).

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser and I. Polosukhin, Attention is all you need, in Advances in Neural Information Processing Systems (2017) 5998–6008.

[34] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, MobileNetV2: inverted residuals and linear bottlenecks, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018) 4510–4520.

[35] M. Tan and Q. Le, EfficientNet: rethinking model scaling for convolutional neural networks, in International Conference on Machine Learning. PMLR (2019) 6105–6114.

[36] S. Greenhill, S. Rana, S. Gupta, P. Vellanki and S. Venkatesh, Bayesian optimization for adaptive experimental design: a review. *IEEE Access* **8** (2020) 13 937–13 948.

[37] M. Laumanns and J. Ocenasek, Bayesian optimization algorithms for multi-objective optimization, in International Conference on Parallel Problem Solving from Nature. Springer, Berlin (2002) 298–307.

[38] K. Swersky, J. Snoek and R.P. Adams, Multi-task Bayesian optimization, in Advances in Neural Information Processing Systems, Vol. 26 (2013).

[39] A. Lambora, K. Gupta and K. Chopra, Genetic algorithm-a literature review, in 2019 International Conference on Machine Learning, Big data, Cloud and Parallel Computing (COMITCon). IEEE (2019) 380–384.

[40] H. Xiao, Weather phenomenon database (WEAPD) (2021). Available: https://doi.org/10.7910/DVN/M8JQCR.

[41] V. Gupta, Weather classification dataset. Accessed: 2023-05-16, https://www.kaggle.com/datasets/vijaygiitk/multiclass-weather-dataset.

[42] M.S. Ghaleb, H. Moushier, H. Shedeed and M. Tolba, Weather classification using fusion of convolutional neural networks and traditional classification methods. *Int. J. Intell. Computing Inf. Sci.* **22** (2022) 84–96.

[43] W. Liu, M. Hua, Z. Deng, Y. Huang, C. Hu, S. Song, L. Gao, C. Liu, L. Xiong and X. Xia, A systematic survey of control techniques and applications: from autonomous vehicles to connected and automated vehicles. Preprint arXiv:2303.05665 (2023).

[44] S.A. Albelwi, Deep architecture based on DenseNet-121 model for weather image recognition. *Int. J. Adv. Comput. Sci. Appl.* **13** (2022) 559–565.