Developing Robust AI Techniques in Computer Vision

Thesis for the Degree of Doctor of Philosophy (PhD)

by

Marcell Beregi-Kovács

Supervisor:

Dr. András Hajdu

University of Debrecen
Doctoral Council for Natural Sciences and Engineering
Doctoral School of Mathematical and Computational Sciences

Debrecen, 2025

Hereby I declare that I prepared this thesis within the Doctoral Council for Natural Sciences and Engineering, Doctoral School of Mathematical and Computational Sciences, University of Debrecen in order to obtain a PhD Degree in Mathematics at the University of Debrecen.

The results published in the thesis are not reported in any other PhD thesis.

Debrecen, ... 2025. .....
signature of the candidate

Hereby I confirm that Marcell Beregi-Kovács candidate conducted his studies with my supervision within the Doctoral School of Mathematical and Computational Sciences between 2018 and 2025. The independent studies and research work of the candidate significantly contributed to the results published in the thesis.
I also declare that the results published in the thesis are not reported in any other theses.
I support the acceptance of the thesis.

Debrecen, ... 2025. .....
signature of the supervisor

Developing Robust AI Techniques in Computer Vision

Dissertation submitted in partial fulfilment of the requirements for the doctoral (PhD) degree in mathematics and computing

Written by Marcell Beregi-Kovács, certified Applied Mathematician

Prepared in the framework of
Doctoral School of Mathematical and Computational Sciences

Dissertation advisor: Dr. András Hajdu

The official opponents of the dissertation:

Dr. _____ _____
Dr. _____ _____

The evaluation committee:

Chairperson: Dr. _____ _____

Members: Dr. _____ _____
Dr. _____ _____
Dr. _____ _____
Dr. _____ _____

The date of the dissertation defence: _____ 2025.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Artificial intelligence (AI) and machine learning (ML) have undergone a profound transformation over the past decades, establishing themselves as critical technologies across a wide range of application domains. Within AI, computer vision has emerged as one of the most dynamic and practically impactful fields, providing the foundation for advances in areas such as autonomous driving, medical diagnostics, environmental monitoring, and augmented reality. As visual data becomes increasingly complex and abundant, and as sensing systems operate in more diverse and uncontrolled environments, there is a growing demand for methods that can operate robustly under adverse conditions, incomplete observations, or structural variations.

Among the many open challenges in computer vision, two fundamental aspects stand out: the availability of realistic and representative training data and the reliability and generalizability of prediction models. Data-driven models, especially deep neural networks, are known to achieve impressive performance when trained on high-quality datasets. However, in real-world scenarios, such datasets often lack coverage of rare events, adverse weather conditions, or other edge cases. This motivates the need for advanced simulation techniques that can augment existing data with physically plausible variations, particularly in the context of safety-critical applications.

On the other hand, the performance of deep classifiers, including convolutional neural networks (CNNs), is heavily influenced by their sensitivity to the training distribution. Despite their expressive power, individual CNNs often exhibit limited robustness when deployed across diverse environments or domains. To address this limitation, ensemble

learning offers a principled way to aggregate multiple models and improve both predictive performance and uncertainty estimation. However, the effectiveness of ensemble methods critically depends on the diversity of the constituent models, a factor that is often overlooked or not optimized in sufficient detail in standard settings.

This dissertation addresses two independent but equally pressing problems situated within this broader context:

First, we propose a new strategy for increasing the diversity and robustness of CNN ensembles. While ensemble learning is a well-established tool in statistical learning theory, its practical application in deep learning is still evolving. In particular, we introduce a loss function augmented with a Pearson correlation-based penalty term that discourages correlated predictions among ensemble members. This term is designed to penalize jointly made incorrect predictions while preserving agreement on correct decisions. By doing so, we guide the optimization process toward learning complementary decision boundaries. The result is an ensemble that is not only more accurate, but also more resilient to out-of-distribution inputs. Furthermore, our method facilitates the construction of lightweight ensembles with minimal computational overhead, making it suitable for resource-constrained deployment scenarios.

Second, we investigate the development of a physically based algorithm for simulating synthetic fog in outdoor images by discretizing the Radiative Transfer Equation (RTE). The accurate modeling of visibility degradation caused by atmospheric scattering is crucial for developing autonomous vehicles and robotics perception systems. Unlike traditional fog augmentation techniques that assume homogeneous fog or rely on heuristic image blending, our method is grounded in first-principles physics. It captures spatial inhomogeneity, depth-dependent attenuation, and anisotropic scattering, thereby enabling the generation of fog effects that are both perceptually plausible and physically interpretable. Although the computational cost of the complete RTE solution is high, we propose an optimized tensor-based implementation that balances realism with efficiency. This enables the generation of large-scale, high-quality datasets for training and evaluation under degraded visibility conditions.

The central scientific challenges addressed in this work include:

- Improving the diversity and robustness of CNN ensembles by explicitly penalizing correlated outputs during training and ana-

lyzing the trade-off between accuracy and diversity.

- Overcoming the limitations of traditional fog simulation models by developing a discretized, learnable version of the RTE suitable for image-space synthesis of non-homogeneous fog.

- Demonstrating the effectiveness of the proposed methods on multiple datasets, including both natural, medical and real-world imagery, and providing a comprehensive empirical evaluation using appropriate metrics.

- Bridging the gap between physical modeling and machine learning by integrating parameter estimation into the fog simulation pipeline via differentiable loss functions.

The contributions of this dissertation reflect a dual commitment: improving algorithmic reliability in classification tasks via ensemble diversity and enhancing the realism of synthetic data via physically grounded simulation. Though distinct in nature, these directions converge in their relevance to practical computer vision systems that must operate under uncertainty and environmental variation.

The dissertation is organized as follows:

- Chapter 2 reviews the necessary theoretical background in deep learning, ensemble methods, and radiative transfer theory, emphasizing the bias–variance trade-off and model uncertainty.

- Chapter 3 presents our correlation-penalized ensemble training framework, including its mathematical formulation, theoretical justification, and empirical validation across various datasets.

- Chapter 4 details the physically based fog generation pipeline, including the discretization of the RTE, the numerical solver, and parameter learning via gradient-based optimization.

- Chapter 5 summarizes the main findings, discusses their implications for robust computer vision, and outlines directions for future research including real-time adaptation and multimodal integration.

Chapter 2

Background and Notation

2.1 General Notation and Learning Paradigms

We denote the dataset by $\mathcal{D}$, whose elements depend on the machine learning paradigm under consideration:

- **Unsupervised learning:** $\mathcal{D} = \{x^{(i)}\}_{i=1}^{m}$, $x^{(i)} \in \mathbb{R}^n$, where no external labels are provided.
- **Supervised learning:** $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$, where $x^{(i)}$ is the input and $y^{(i)}$ is the corresponding label of the i-th example.
- **Self-supervised learning:** $\mathcal{D} = \{x^{(i)}\}_{i=1}^{m}$, where labels are automatically derived from the structure of the data itself (e.g., via contrastive objectives or context prediction).

In practice, the dataset $\mathcal{D}$ is typically divided into three disjoint subsets:

$$\mathcal{D} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{val}} \cup \mathcal{D}_{\text{test}}, \quad \mathcal{D}_{\text{train}} \cap \mathcal{D}_{\text{val}} = \mathcal{D}_{\text{train}} \cap \mathcal{D}_{\text{test}} = \mathcal{D}_{\text{val}} \cap \mathcal{D}_{\text{test}} = \emptyset.$$

- $\mathcal{D}_{\text{train}}$: Used to learn the model parameters by minimizing the training loss.
- $\mathcal{D}_{\text{val}}$: Used for model selection and hyperparameter tuning; it monitors generalization during training.

- $\mathcal{D}_{\text{test}}$: Used only once after training to evaluate final model performance on unseen data.

This split helps to ensure that the model generalizes well and is not overfitted to the training set.

In supervised settings, the goal is to approximate an unknown function $f : \mathcal{X} \to \mathcal{Y}$ with a model h_Θ parameterized by Θ, such that $h_\Theta(x^{(i)}) \approx y^{(i)}$ (Sometimes $h_\Theta(x^{(i)})$ is referred by $\hat{y}^{(i)}$).

Common supervised tasks:

- **Regression:** $y^{(i)} \in \mathbb{R}^n$ or a continuous space.
- **Classification:** $y^{(i)} \in \{1, \ldots, L\}$, where L is the number of classes.

The model's performance is usually measured via a loss function $\mathcal{L}(y, \hat{y})$ such as:

- **Mean squared error (MSE):** $\mathcal{L}(y, \hat{y}) = \|y - \hat{y}\|^2$
- **Cross-entropy (CE):** $\mathcal{L}(y, \hat{y}) = -\sum_{c=1}^{L} y_c \log \hat{y}_c$

2.1.1 Bias–Variance Trade-Off

A critical concept in generalization is the bias-variance trade-off:

- *Bias:* Error due to simplifying assumptions (e.g., underfitting).
- *Variance:* Error due to sensitivity to training data (e.g., overfitting).

A good learning algorithm balances these components. Mathematically, for a learned hypothesis h and target function f, the expected prediction error can be decomposed as:

$$\mathbb{E}_{\mathcal{D}}[(h(x) - f(x))^2] = \text{Bias}^2 + \text{Variance} + \text{Irreducible noise}.$$

This decomposition is central in evaluating model capacity and generalization performance.

2.2 Core Concepts in Deep Learning

Deep neural networks consist of layers of parameterized transformations. CNNs are specialized for image processing.

2.2.1 CNN Architecture Components

CNNs are composed of several key building blocks, each contributing to hierarchical feature extraction and classification.

- **Convolutional Layers:** These layers apply learnable filters (also called kernels) $W \in \mathbb{R}^{k \times k \times c_{\text{in}} \times c_{\text{out}}}$ to local spatial regions of the input X. The 2D discrete convolution (without bias) at position (i, j) for channel m is given by:

$$Z_m(i,j) := (X * W)_m(i,j) = \sum_{u=0}^{k-1} \sum_{v=0}^{k-1} \sum_{c=1}^{c_{\text{in}}} W_{u,v,c,m} \cdot X_{i+u,j+v,c}.$$

  Two important hyperparameters control the convolution behavior:

  - **Stride** s: The step size with which the kernel moves over the input. The larger stride reduces the spatial resolution.
  - **Padding** p: Zero-padding around the input to preserve dimensions. With padding p and stride s, the output size n_{out} for input size n_{in} and kernel size k is:

$$n_{\text{out}} = \left\lfloor \frac{n_{\text{in}} - k + 2p}{s} \right\rfloor + 1.$$

- **Pooling Layers:** These perform spatial downsampling on Z to reduce dimensionality and increase translation invariance. The most common types are:

  - **Max Pooling:** Takes the maximum value over a region:

$$P(i,j) = \max_{(u,v) \in R_{i,j}} Z(u,v),$$

    where $R_{i,j}$ denotes the pooling region around position (i, j).

– **Average Pooling:** Computes the average over the region:

$$P(i,j) = \frac{1}{|R_{i,j}|} \sum_{(u,v)\in R_{i,j}} Z(u,v).$$

- **Fully Connected (Dense) Layers:** Each neuron in a fully connected layer receives input from all neurons in the previous layer. Given an input vector $x \in \mathbb{R}^n$, a weight matrix $W \in \mathbb{R}^{m\times n}$, and a bias vector $b \in \mathbb{R}^m$, the output is computed as:

$$y = \sigma(Wx + b),$$

where $\sigma(\cdot)$ denotes a non-linear activation function such as ReLU or sigmoid. The output vector $y \in \mathbb{R}^m$ serves as the input to the next layer. Fully connected layers are commonly used in the final stages of classification networks, but they may also appear in intermediate layers to project features to different dimensions.

- **Activation Functions:** Nonlinearities applied after convolution or fully connected layers. Common examples include:
  - **ReLU (Rectified Linear Unit):**

$$\mathrm{ReLU}(x) = \max(0, x).$$

  - **Leaky ReLU:**

$$\mathrm{LeakyReLU}(x) = \begin{cases} x & \text{if } x \geq 0, \\ \alpha x & \text{if } x < 0, \end{cases}$$

    where $\alpha \in (0,1)$ is a small constant (e.g., $\alpha = 0.01$).
  - **Sigmoid:**

$$\sigma(x) = \frac{1}{1+e^{-x}}, \quad \sigma(x) \in (0,1).$$

  - **Tanh (Hyperbolic Tangent):**

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad \tanh(x) \in (-1,1).$$

– **Softplus (Smooth ReLU):**

$$\text{Softplus}(x) = \ln(1 + e^x).$$

- **Softmax Output:** Used in the final classification layer to convert logits z_i into a probability distribution over classes:

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}.$$

It ensures that $\sum_i \text{Softmax}(z_i) = 1$, making it suitable for multi-class classification.

2.2.2 Model Optimization

The goal of training a neural network is to find a parameter vector $\Theta \in \mathbb{R}^d$ that minimizes a loss function $\mathcal{L}(\Theta, \mathcal{D})$, given a dataset $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^m$.

In full-batch gradient descent, the parameters are updated iteratively as follows:

$$\Theta \leftarrow \Theta - \alpha \nabla_\Theta \mathcal{L}(\Theta, \mathcal{D}),$$

where $\alpha > 0$ is the learning rate and $\nabla_\Theta \mathcal{L}$ denotes the gradient of the loss with respect to the parameters. However, evaluating the gradient over the entire dataset can be computationally expensive and memory-intensive.

In practice, **Stochastic Gradient Descent (SGD)** and its variants are preferred. These methods operate on *mini-batches* $\mathcal{B} \subset \mathcal{D}$ of size $B \ll m$, yielding the update rule:

$$\Theta \leftarrow \Theta - \alpha \nabla_\Theta \mathcal{L}(\Theta, \mathcal{B}),$$

where the loss is approximated over the mini-batch:

$$\mathcal{L}(\Theta, \mathcal{B}) = \frac{1}{B} \sum_{(x,y) \in \mathcal{B}} \mathcal{L}(h_\Theta(x), y),$$

and ℓ is a sample-wise loss function, e.g., cross-entropy or MSE.

To improve convergence and stability, adaptive methods dynamically adjust the learning rate for each parameter:

- **RMSProp:** Maintains a moving average of squared gradients to normalize parameter updates. The update rule is:

$$v_t \leftarrow \beta v_{t-1} + (1-\beta)(\nabla_\Theta \mathcal{L})^2,$$
$$\Theta \leftarrow \Theta - \alpha \cdot \frac{\nabla_\Theta \mathcal{L}}{\sqrt{v_t} + \varepsilon},$$

  where $\beta \in [0, 1)$ is the decay rate, and ε is a small constant for numerical stability.

- **Adam:** Combines momentum and adaptive learning rates using estimates of both first and second moments:

$$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1)\nabla_\Theta \mathcal{L},$$
$$v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2)(\nabla_\Theta \mathcal{L})^2,$$
$$\Theta \leftarrow \Theta - \alpha \cdot \frac{m_t}{\sqrt{v_t} + \varepsilon},$$

  where $\beta_1, \beta_2 \in [0, 1)$ are decay rates and ε is a small constant for numerical stability.

This process is repeated for multiple *epochs*, each consisting of several mini-batch updates. Batches are typically sampled randomly at each epoch to avoid overfitting to data order and to ensure better generalization.

2.2.3 Architectures

Popular CNN architectures have evolved significantly over the past decade, each introducing key innovations that contributed to breakthroughs in image classification and other computer vision tasks.

- **AlexNet** [1]: Marked a major breakthrough by winning the ImageNet 2012 competition by a large margin. Its key innovations included the use of ReLU activations (instead of sigmoid/tanh), overlapping max-pooling, and GPU-accelerated training, which demonstrated the feasibility of large-scale deep learning.

- **VGG** [2]: Proposed a very deep architecture using only 3×3 convolutional filters and 2×2 max-pooling. The simplicity of the network and the uniform design of the layers facilitated generalization and transfer learning. It highlighted that depth alone, if properly structured, can lead to performance improvements.

- **ResNet** [3]: Introduced *residual connections*, which allows the training of extremely deep networks (e.g., 152 layers) without vanishing gradients. The residual block computes $F(x) + x$ instead of just $F(x)$, enabling the network to learn the residual mappings. This innovation laid the foundation for many subsequent architectures.

- **MobileNet** [4]: Optimized for mobile and embedded devices, MobileNet introduced *depth-wise separable convolutions*, which factorize standard convolutions into depth-wise and point-wise steps. This drastically reduces the number of parameters and computations while preserving competitive accuracy.

Beyond classical CNNs, recent years have seen the rise of Vision Transformers (ViTs) [5] and hybrid CNN-Tuner models [6] as competitive alternatives in computer vision. ViTs treat images as sequences of patches and apply self-attention mechanisms to capture long-range dependencies, often achieving state-of-the-art results in classification, detection, and segmentation tasks. Although they typically require large-scale datasets and computational resources for practical training, their ability to model global context represents a significant advantage over purely convolutional approaches.

Building on this idea, hierarchical variants such as the Swin Transformer [7] further improved scalability and efficiency by introducing shifted window attention, making them competitive in a wide range of vision tasks. In parallel, hybrid architectures combine CNN-based feature extractors with Transformer layers to leverage the strengths of both paradigms: the inductive biases and efficiency of convolutions with the flexibility and global receptive field of self-attention. Representative examples include Bottleneck Transformers [8], CoAtNet [9], and CMT [10], which consistently demonstrates improved accuracy–efficiency trade-offs and strong generalization in domains ranging from natural images to medical imaging and remote sensing.

Together, these developments reflect the evolution of vision architectures from convolution-only designs toward attention-based and hybrid models, expanding the toolbox for building expressive and efficient classifiers.

Although this dissertation focuses primarily on CNN ensembles, these emerging architectures provide promising directions for future research. In particular, the principles of diversity promotion through

correlation penalization, introduced in this work, could also be extended to ensembles composed of ViTs or CNN-Transformer hybrids, potentially amplifying their robustness in safety-critical applications.

2.3 Ensemble Learning: Principles and Measures

Ensemble learning aims to improve predictive performance and robustness by combining multiple base models $\{h_i\}_{i=1}^{N}$. The effectiveness of an ensemble stems not only from the strength of the individual learners, but critically from their *diversity*—i.e., the ability to make different errors.

2.3.1 Aggregation Strategies

Given input $x \in \mathbb{R}^d$, the ensemble prediction $F_{ens}(x)$ is obtained by aggregating the outputs $\{h_i(x)\}$. Common strategies include:

- **Simple Averaging:**

$$F_{ens}(x) = \frac{1}{N} \sum_{i=1}^{N} h_i(x),$$

  used primarily for regression or soft classification outputs (e.g., softmax scores).

- **Weighted Averaging:**

$$F_{ens}(x) = \sum_{i=1}^{N} w_i h_i(x), \quad \text{with } \sum_{i=1}^{N} w_i = 1,\ w_i \geq 0,$$

  where weights may be learned or derived from validation accuracy.

- **Majority Voting:**

$$F_{ens}(x) = \arg\max_{y} \sum_{i=1}^{N} \chi_y(h_i(x)),$$

applicable to classification with discrete outputs and $\chi_y(\cdot)$ is the indicator function defined as

$$\chi_y(h_i(x)) = \begin{cases} 1, & \text{if } h_i(x) = y, \\ 0, & \text{otherwise.} \end{cases}$$

- **Stacking:** A meta-learner (e.g., MLP or SVM) is trained on the predictions $\{h_i(x)\}$ to compute the final output.
- **Bagging (Bootstrap Aggregating):** Trains each model on a different bootstrap sample. Aggregation typically uses averaging or majority voting. Bagging reduces variance and increases stability.
- **Boosting:** Models are trained sequentially, each correcting the errors of the previous one. The final prediction is a weighted sum. Boosting reduces bias but is sensitive to noise.

2.3.2 Training Paradigms

- **Offline Ensembles:** Each h_i is trained independently. Diversity is typically induced via random initialization, different architectures, or data augmentation.
- **Online Ensembles:** All h_i are trained jointly within a unified architecture. This allows for direct enforcement of ensemble-level objectives, such as accuracy and diversity, through a shared loss:

$$\mathcal{L} = \mathcal{L}_{\text{main}} + \lambda \cdot \mathcal{L}_{\text{diversity}},$$

where $\lambda \in \mathbb{R}^+$ balances task accuracy and output decorrelation.

2.3.3 Diversity Metrics

Ensemble success relies on individual accuracy and prediction diversity. Several metrics quantify diversity:

- **Disagreement Rate:**

$$\text{Dis}(h_i, h_j) = \frac{1}{m} \sum_{k=1}^{m} \mathbb{I}[h_i(x_k) \neq h_j(x_k)],$$

where $\mathbb{I}$ is the indicator function and m is the number of samples.

- **Pearson Correlation:**

$$\varrho(F_i, F_j) = \frac{\mathrm{Cov}(F_i, F_j)}{\sigma_{F_i}\sigma_{F_j}},$$

  where $F_i \in \mathbb{R}^K$ are softmax outputs. Used as a penalty term in this dissertation.

- **Ambiguity:**

$$A(x) = \frac{1}{N}\sum_{i=1}^{N}\left(h_i(x) - \bar{h}(x)\right)^2, \quad \bar{h}(x) = \frac{1}{N}\sum_{i=1}^{N} h_i(x),$$

  which measures variance among outputs at a given input.

2.3.4 Diversity Promotion Strategies

- **Implicit:** Via data augmentation, architecture variation, or different initialization seeds.

- **Explicit:** Diversity is enforced during training using specialized loss terms:

  - Negative Correlation Learning (NCL) [11]
  - KL-divergence between outputs
  - Pearson correlation-based penalty (used in this work)

2.4 Radiative Transfer in Computer Vision

The Radiative Transfer Equation (RTE) [12, 13] provides a physically grounded framework for modeling the propagation of light in participating media such as fog, haze, or smoke. It captures key physical processes including absorption, scattering, and emission, and plays a crucial role in simulating realistic atmospheric effects in computer vision and graphics.

In the general time-dependent, spectral form, the radiance function L depends on spatial position $x \in \mathbb{R}^3$, direction $\sigma \in \mathbb{S}^2$, time $t \in \mathbb{R}$, and wavelength $\lambda \in \mathbb{R}_{>0}$. The full integro-differential form of the RTE is:

$$\begin{aligned}\frac{1}{c}\frac{\partial L}{\partial t}(x,\sigma,t,\lambda) + \langle \nabla_x L(x,\sigma,t,\lambda), \sigma\rangle = \\ - K(x,t,\lambda)L(x,\sigma,t,\lambda) + S(x,\sigma,t,\lambda) \\ + K_s(x,t,\lambda)\int_{\mathbb{S}^2} \phi(\sigma,\omega,t,\lambda)L(x,\omega,t,\lambda)\, dS(\omega)\end{aligned} \tag{2.1}$$

where:

- $K(x,t,\lambda)$ is the extinction coefficient (sum of absorption and scattering),
- $K_s(x,t,\lambda)$ is the scattering coefficient,
- $\phi(\sigma,\omega,t,\lambda)$ is the phase function describing angular redistribution,
- $S(x,\sigma,t,\lambda)$ is the emission source term,
- c is the speed of light in the medium.

The general form of the RTE in a monochromatic, stationary and emission-free setting is given by:

$$\frac{\partial L(r,\sigma)}{\partial r} = -K(r,\sigma)L(r,\sigma) + K_s(r,\sigma)\int_{\mathbb{S}^2} L(r,\omega)\,\phi(\omega,\sigma)\,d\omega,$$

where:

- $L(r,\sigma)$ denotes the radiance at position r in direction σ,
- $K(r,\sigma)$ is the extinction coefficient (comprising both absorption and scattering),
- $K_s(r,\sigma)$ is the scattering coefficient,
- $\phi(\omega,\sigma)$ is the phase function describing angular scattering probability from direction ω to σ,
- and $\mathbb{S}^2$ denotes the unit sphere representing all possible directions.

This formulation allows modeling of light attenuation (via K), as well as in-scattering contributions from all directions (via the integral term). Importantly, the RTE is capable of describing highly anisotropic and inhomogeneous media, which makes it particularly well-suited for simulating realistic fog.

2.4.1 Special Case: Koschmieder's Law

In homogeneous fog with isotropic scattering:

$$L(d) = L_0 e^{-Kd} + L_{\text{air}}(1 - e^{-Kd}).$$

This corresponds to a zeroth-order approximation of RTE.
For implementation and dataset synthesis, see Chapter 4.

Symbol	Meaning
$\mathcal{D}$	Entire dataset
$\mathcal{D}_{\text{train}}$	Training set
$\mathcal{D}_{\text{val}}$	Validation set
$\mathcal{D}_{\text{test}}$	Test set
$x^{(i)}$	Input feature vector for the i-th sample
$y^{(i)}$	True label for the i-th sample
$\hat{y}^{(i)}$	Predicted label for the i-th sample
f	Ground truth target function
h, h_Θ	Model / hypothesis parameterized by Θ
Θ	Model parameters
$\mathcal{L}(y, \hat{y})$	General loss function
ℓ	Sample-wise loss function (e.g., MSE, CE)
α	Learning rate
L	Number of output classes
L_{CE}	Cross-entropy loss
L_{corr}	Correlation-based penalty term
ϱ	Pearson correlation coefficient
A	Prediction ambiguity (diversity metric)
$c_{\text{in}}, c_{\text{out}}$	Number of input/output channels
$n_{\text{in}}, n_{\text{out}}$	Input/output spatial dimensions
$\chi_y(\cdot)$	Indicator function for class y
$\phi(\omega, \sigma)$	Phase function (angular scattering)
K or β	Extinction coefficient
K_s	Scattering coefficient
$S(x, \sigma, t, \lambda)$	Emission source term in RTE
$L(x, \omega)$	Radiance at point x in direction ω
L_0	Initial radiance from the object
L_{air}	Background airlight in Koschmieder's model

Table 2.1: Extended summary of key symbols used in this thesis.

Chapter 3

Increasing CNN Ensemble Diversity via Correlation Penalty

3.1 Motivation

CNNs have become dominant tools in digital image processing, achieving state-of-the-art performance in tasks such as classification, detection, localization, and segmentation [14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]. Their success is largely due to their capacity to learn hierarchical feature representations from data through convolutional filters. Nevertheless, the performance of a single CNN model can be limited by overfitting, architectural biases, and variance in decision boundaries. To overcome these challenges, ensemble learning has emerged as a powerful strategy to improve generalization and robustness.

The idea of combining multiple learners into an ensemble dates back to early foundational work by Hansen [25], who demonstrated that an ensemble of neural networks can outperform individual models. Since then, ensemble techniques have gained prominence across multiple machine learning domains including regression [26, 11], classification [27, 28, 29], semantic segmentation [30], and metric learning [31].

However, building an effective ensemble is not merely about aggregating predictions. It depends critically on two factors: (i) the accuracy of individual models and (ii) the diversity among them. Without sufficient diversity, ensemble members tend to make similar errors, lim-

iting the potential performance gains.

Several strategies have been proposed to achieve output diversity. Classical methods involve training models with different initializations, architectures [32, 33, 34], data subsets, or through variations in preprocessing [35, 28]. Others rely on cyclic learning rate schedules, such as Snapshot Ensembles [36, 37, 38], or introduce architectural manipulations such as layer splitting [39]. Although these methods are effective, they are often applied in a manner *offline*: models are trained independently, and their outputs are combined post hoc. This setting makes it difficult to optimize ensemble diversity during training directly.

Aggregation strategies themselves can be categorized as simple averaging [32], weighted voting [34, 26], majority voting [39, 35, 40], or meta-learning-based fusion via support vector machines, random forests, or multilayer perceptrons [27, 41]. While these methods enable output fusion, they do not address diversity explicitly.

A more promising approach lies in *online* ensembles, where the aggregation is integrated into the network architecture. Through a joint fully connected layer or trainable fusion module, CNNs can be trained simultaneously, allowing the entire ensemble to learn collaboratively [42, 43, 31, 44]. Such architectures are computationally more efficient and open up new opportunities for loss-level diversity constraints.

Among loss-level strategies, *Negative Correlation Learning* (NCL) [11] stands out. NCL introduces a penalty term into the loss function that encourages the base learners to differ in their predictions. Though initially proposed for regression with multilayer perceptrons (MLPs), it has since been adapted to convolutional networks for crowd counting and image super-resolution [42], and for classification tasks using variants of AdaBoost [45]. More recent efforts also employ cosine similarity [43], KL-divergence, or mutual information to encourage output decorrelation.

Despite these advances, most methods either assume Mean Squared Error (MSE) loss or lack fine-grained control over how correlation affects correct versus incorrect classifications. This motivates our proposed method, which integrates a Pearson correlation-based penalty term into the cross-entropy loss. The key idea is to penalize strongly correlated incorrect predictions, while allowing members to agree on correct ones. This provides a more nuanced and task-appropriate formulation of diversity, particularly suitable for classification problems.

In our earlier work [46], we explored ensemble structures using sev-

eral well-known CNNs (AlexNet [47], GoogLeNet [48], ResNet [49], and VGGNet [50]) and showed how statistical aggregation could improve performance without expanding the training set. However, these ensembles were loosely coupled and trained independently.

Later, we proposed an architectural fusion framework that connects member CNNs through a shared fully connected layer [51], enabling joint training and backpropagation through the ensemble. This dissertation further enhances this architecture by adding a Pearson-correlation-based regularization term to the loss function. This term promotes functional diversity by penalizing redundant output behavior, particularly for incorrect predictions.

Through comprehensive experimentation on both natural and clinical image datasets, we demonstrate that the proposed method:

- Improves ensemble classification accuracy across tasks.
- Reduces the frequency of jointly made misclassifications.
- Provides a scalable and architecture-agnostic ensemble training framework.

The architecture and loss-based training methodology described in this chapter, aimed at promoting diversity in CNN ensembles via Pearson correlation penalization, were originally published in our study "Composing Diverse Ensembles of Convolutional Neural Networks by Penalization" [52]. The detailed analysis and experimental results presented here are derived from that publication, with additional context and critical interpretation adapted for inclusion in this dissertation.

3.2 Learning Methodology and Network Architecture

In recent years, numerous CNN architectures have been developed for natural image classification, including GoogLeNet [48], AlexNet [47], ResNet [49], VGGNet [50], DenseNet [53], and MobileNet [54], among others. Many of these architectures are publicly available as pre-trained models trained on the large-scale ImageNet [55] dataset, consisting of approximately 1.28 million labeled images across 1,000 categories . This makes them ideal candidates for transfer learning

[56], where we fine-tune pre-trained weights using task-specific data. Alternatively, if sufficient labeled data are available, models may be trained from scratch by initializing the network weights randomly.

This study considers both approaches: transfer learning and training from random initialization. Multiple CNNs are integrated into a unified ensemble network, producing a directed acyclic computational graph where the final layers are fused. Specifically, we concatenate the outputs of the member networks via a fully-connected ($\mathcal{FC}$) layer. This fusion strategy allows us to jointly exploit the representational power of multiple CNNs while enabling the model to learn how to combine their outputs effectively.

The core challenge in ensemble learning is to increase the classification performance by leveraging the strengths of individual models while mitigating their correlated weaknesses. A critical scenario arises when multiple members of the ensemble misclassify the same sample in the same way, leading to strongly correlated errors and reduced ensemble effectiveness. This commonly occurs because the networks are trained on the same data and optimized using the same objective function.

To address this issue, we propose a methodological innovation that jointly optimizes the CNN ensemble while explicitly encouraging diversity among the members, particularly in their misclassification patterns. We achieve this by augmenting the conventional categorical cross-entropy loss function with a novel penalty term that measures the pairwise correlations of the ensemble outputs and the target one-hot vector. This encourages the ensemble members to make different errors, thereby increasing the robustness and generalization capacity of the final model.

In practice, we construct the ensemble by interconnecting well-known CNNs — pre-trained or randomly initialized — via a shared $\mathcal{FC}$ layer, followed by a softmax activation. The parameters of the fusion layer and the CNNs are updated simultaneously using backpropagation. The loss function $\mathcal{L}$ for training takes the following general form:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \cdot \mathcal{L}_{\text{corr}},$$

where $\mathcal{L}_{\text{CE}}$ denotes the standard categorical cross-entropy loss and $\mathcal{L}_{\text{corr}}$ represents the correlation penalty that discourages redundant errors among CNNs. The hyperparameter $\lambda > 0$ controls the trade-off

between accuracy and diversity.

Cross-entropy is selected as the base loss since it is widely used in classification tasks and serves as a component in several hybrid loss functions, particularly in object classification [57, 58, 59, 60]. Nonetheless, our method is flexible and supports alternative loss formulations as well.

By incorporating this diversity-oriented penalty directly into the training objective, we ensure that the final ensemble not only performs well in terms of accuracy but also exhibits improved robustness through diversified member behavior. This forms the basis of this dissertation's methodological contribution, setting the stage for the detailed formulation of the Pearson correlation-based penalty in the next section.

3.2.1 The Fusion of the Member Networks

To implement the ensemble structure as a single trainable neural network, we propose a specific architectural fusion of N CNNs. Instead of treating each CNN as an independent module followed by post-hoc prediction fusion, we integrate them at the architectural level, resulting in an end-to-end trainable system.

In the proposed framework, we begin by modifying each individual CNN. Their final softmax layers are removed, and their original fully connected (FC) output layers are replaced with new *task-specific FC layers*, whose output dimensionality equals the number of target classes, denoted by L. These task-specific FC layers transform the learned features into class logits, which are subsequently normalized into probability vectors.

The outputs of all member CNNs are concatenated and passed through an additional trainable *fusion fully connected layer* (hereafter referred to as the *fusion FC*), as illustrated in Figure 3.1. While the task-specific FC layers generate predictions for each individual CNN, the fusion FC is designed to learn an adaptive combination of these predictions. Formally, it applies a weighted linear transformation to the concatenated outputs and produces an aggregated representation, which is then normalized by a final softmax layer to yield the ensemble probability distribution. This design enables the fusion FC to emphasize or down-weight individual networks depending on their reliability, providing a more flexible and learnable alternative to naive averaging

or majority voting.

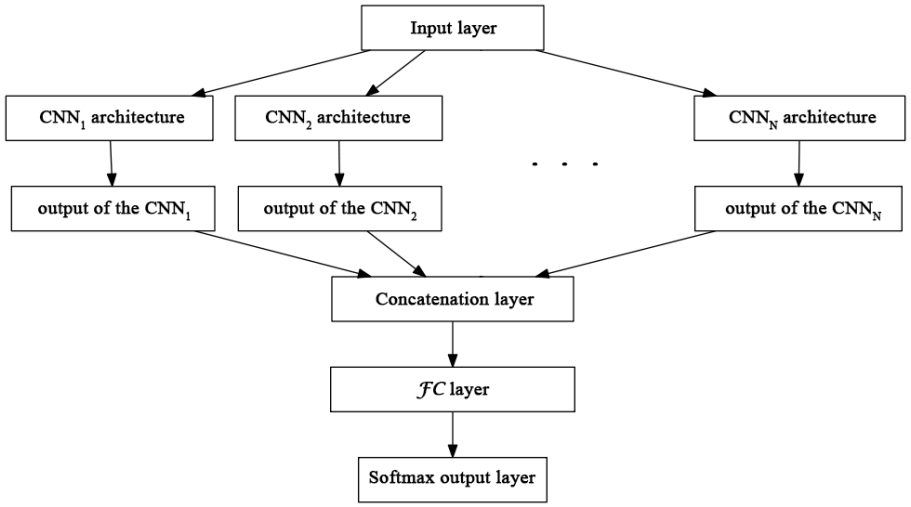

Figure 3.1: Architecture of the proposed ensemble of CNNs. Task-specific FC layers are used to produce class-level outputs for each CNN, which are concatenated and passed through a fusion FC layer to obtain the ensemble prediction.

Let the training dataset be denoted by:

$$\{x^{(n)}, y^{(n)}\}, \quad n = 1, \ldots, M,$$

where $x^{(n)}$ is the n-th input sample and $y^{(n)} \in \mathbb{R}^L$ is its corresponding one-hot encoded class label. Here, M denotes the total number of training examples and L the number of distinct classes, typically with $L \ll M$.

Let $h_i^{(n)}$ denote the output of the task-specific FC layer of the i-th CNN for input $x^{(n)}$, producing a probability vector over the L classes. These vectors are concatenated and passed through the fusion FC, which computes a weighted combination of the CNN outputs:

$$\tilde{F}_{\text{ens}}^{(n)} = \sum_{i=1}^{N} A_i h_i^{(n)},$$

where each A_i is a learnable weight matrix of size $L \times L$ associated with the i-th CNN. The matrices A_i are initialized close to scaled identity

matrices:

$$A_i = \begin{pmatrix} \frac{1}{N} & \varepsilon & \cdots & \varepsilon \\ \varepsilon & \frac{1}{N} & \cdots & \varepsilon \\ \vdots & \vdots & \ddots & \vdots \\ \varepsilon & \varepsilon & \cdots & \frac{1}{N} \end{pmatrix}, \tag{3.1}$$

where $\varepsilon \approx 0$. This initialization approximates a simple arithmetic mean of the CNN outputs. However, since the A_i matrices are trainable parameters, they are updated during training through backpropagation to learn an optimal combination of the base learners' outputs.

To ensure that the ensemble prediction represents a valid probability distribution over the classes, a final softmax layer is applied:

$$F_{\text{ens}}^{(n)} = \text{softmax}\left(\tilde{F}_{\text{ens}}^{(n)}\right).$$

This fusion mechanism enables the model to jointly learn both the CNN parameters and their optimal combination strategy in a fully end-to-end fashion. It also provides a flexible structure that supports diversity enforcement, which is addressed in the following section through the introduction of a Pearson correlation-based regularization term.

3.2.2 Loss Function: Diversity-Penalized Training Objective

To enhance the diversity of the ensemble and reduce the occurrence of jointly made incorrect predictions, we extend the traditional classification loss with a novel correlation-based penalty term. This term, grounded in the Pearson correlation coefficient, is designed to penalize correlated misclassifications while tolerating agreement on correct predictions. The goal is to achieve higher ensemble accuracy by explicitly encouraging diverse decision boundaries among the member CNNs.

As discussed earlier, we denote the original classification loss by $\mathcal{L}_{\text{CE}}$, which in our case is the categorical cross-entropy. This loss encourages each individual network to learn to classify input samples correctly.

Pearson Correlation Coefficient. Let $X = (X_1, \ldots, X_K)$ and $Y = (Y_1, \ldots, Y_K)$ be K-dimensional real vectors. The Pearson correlation

coefficient $\varrho(X, Y)$ between X and Y is defined as:

$$\begin{aligned}\varrho(X,Y) &= \frac{1}{K-1}\sum_{i=1}^{K}\left(\frac{X_i-\overline{X}}{\sigma_X}\right)\left(\frac{Y_i-\overline{Y}}{\sigma_Y}\right)\\ &= \frac{\sum_{i=1}^{K}(X_i-\overline{X})(Y_i-\overline{Y})}{\sqrt{\sum_{i=1}^{K}(X_i-\overline{X})^2\sum_{i=1}^{K}(Y_i-\overline{Y})^2}},\end{aligned}$$

where $\overline{X}$ and $\overline{Y}$ denote the means, and σ_X, σ_Y the standard deviations of the vectors X and Y, respectively.

Penalty Matrix. For each training sample $x^{(n)}$ with one-hot label vector $y^{(n)}$, let $h_i^{(n)}$ denote the softmax output of the i-th CNN. Define the $(N+1)\times(N+1)$ symmetric correlation matrix $C^{(n)}$ as:

$$C^{(n)} = \begin{pmatrix} \varrho(h_1^{(n)},h_1^{(n)}) & \cdots & \varrho(h_1^{(n)},h_N^{(n)}) & -\varrho(h_1^{(n)},y^{(n)}) \\ \varrho(h_2^{(n)},h_1^{(n)}) & \cdots & \varrho(h_2^{(n)},h_N^{(n)}) & -\varrho(h_2^{(n)},y^{(n)}) \\ \vdots & \ddots & \vdots & \vdots \\ \varrho(h_N^{(n)},h_1^{(n)}) & \cdots & \varrho(h_N^{(n)},h_N^{(n)}) & -\varrho(h_N^{(n)},y^{(n)}) \\ -\varrho(y^{(n)},h_1^{(n)}) & \cdots & -\varrho(y^{(n)},h_N^{(n)}) & \varrho(y^{(n)},y^{(n)}) \end{pmatrix}.$$

Based on $C^{(n)}$, we define the correlation-based penalty $\mathcal{L}_{\text{corr}}$ over the full training set of M samples as:

$$\mathcal{L}_{\text{corr}} = \frac{1}{M}\sum_{n=1}^{M}\left[\sum_{i=1}^{N}\sum_{j=i}^{N}C_{ij}^{(n)} + N\sum_{i=1}^{N+1}C_{i,N+1}^{(n)}\right]. \tag{3.2}$$

The first term in brackets accumulates the upper triangular entries of the $N\times N$ block of $C^{(n)}$, quantifying mutual correlation among the CNNs. The second term encourages each CNN to align with the target label vector $y^{(n)}$, by penalizing negative or weak correlations (since the correlation terms enter with a minus sign). Note that the diagonal values $C_{ii}^{(n)} = 1$ are constant and serve as regularization anchors.

The formulation in Equation (3.2) is justified by the following proposition, which asserts that the penalty term $\mathcal{L}_{\text{corr}}$ grows with the level of correlated error:

Proposition 1 *Including the penalty term* (3.2) *provides a loss function having increasing values according to the order of the following cases:*

- *all the experts (member networks) classify the n-th training sample correctly,*
- *some of the experts do not assign the n-th training sample to the true class, but their outputs are different classes,*
- *some of the experts classify the n-th training sample in the same false class,*
- *all experts assign the n-th training sample to false classes, but these classes differ,*
- *all experts assign the n-th training sample to the same false class.*

Proof. Considering only one input data item, let us investigate the value of the corresponding part of the penalty term (i.e., only one term of the outer sum in (3.2), so here we omit the upper index n). Denote by $\mathcal{L}_{\text{corr}0}$ the value in the perfect classification case with $h_i = d$ for all $i = 1, \ldots, N$, i.e., when all experts assign the correct class to the given sample with probability 1. Then all the correlation coefficients are equal to 1, and

$$\mathcal{L}_{\text{corr}0} = \frac{N(N+1)}{2} - N^2 + N = -\frac{N}{2}(N-3).$$

Now, perturbing one of the h_i vectors (e.g. h_1) we have $\mu_1 := \varrho(h_1, d) < 1$ and

$$\varrho(h_1, h_j) = \mu_1 < 1, \quad j = 2, \ldots, N.$$

Denoting by $\mathcal{L}_{\text{corr}1}$ the current penalty term we obtain that

$$\begin{aligned}
\mathcal{L}_{\text{corr}1} &= \varrho(h_1, h_1) + \sum_{i=2}^{N} \varrho(h_1, h_i) - N\varrho(h_1, d) \\
&+ \sum_{i=2}^{N}\sum_{j=i}^{N} \varrho(h_i, h_j) - N\sum_{i=2}^{N} \varrho(h_i, d) + N\varrho(d, d) \\
&= 1 + (N-1)\mu_1 - N\mu_1 + \frac{(N-1)N}{2} - N(N-1) + N \\
&= -\frac{N}{2}(N-3) + 1 - \mu_1 = \mathcal{L}_{\text{corr}0} + 1 - \mu_1 > \mathcal{L}_{\text{corr}0},
\end{aligned}$$

which shows the farther is h_1 from the perfect case, the larger is the penalty.

If we perturb the next vector (h_2), too, then $\mu_2 := \varrho(h_2, d) < 1$ and

$$\varrho(h_2, h_j) = \mu_2 < 1, \quad j = 3, \ldots, N.$$

The values $\varrho(h_1, h_j)$ remain unchanged for $j = 3, \ldots, N$, and introducing the variable $\mu_{12} := \varrho(h_1, h_2)$, the penalty term $\mathcal{L}_{\text{corr}2}$ can be written as

$$\begin{aligned}
\mathcal{L}_{\text{corr}2} &= \varrho(h_1, h_1) + \sum_{i=3}^{N} \varrho(h_1, h_i) - N\varrho(h_1, d) + \varrho(h_2, h_2) \\
&+ \sum_{i=3}^{N} \varrho(h_2, h_i) - N\varrho(h_2, d) \\
&+ \varrho(h_1, h_2) + \sum_{i=3}^{N} \sum_{j=i}^{N} \varrho(h_i, h_j) - \sum_{i=3}^{N} \varrho(h_i, d) + N\varrho(d, d) \\
&= 1 + (N-2)\mu_1 - N\mu_1 + 1 + (N-2)\mu_2 - N\mu_2 \\
&+ \mu_{12} + \frac{(N-2)(N-1)}{2} - (N-2)N + N \\
&= \mathcal{L}_{\text{corr}1} + (1 - \mu_2) + (1 + \mu_{12} - \mu_1 - \mu_2).
\end{aligned}$$

If $\mu_{12} = 1$, which means $h_1 = h_2$, then $\mathcal{L}_{\text{corr}2} > \mathcal{L}_{\text{corr}1}$. When $\mu_{12} = 1$, or $\mu_{12} \approx 1$, i.e., the h_1, h_2 vectors are highly correlated, then the penalty depends on the fact whether these vectors are close to d, or not. In the first case (when they are not the perfect one-hot vector, but close to it) $\mu_1 \approx 1$ and $\mu_2 \approx 1$, so $\mathcal{L}_{\text{corr}2} \approx \mathcal{L}_{\text{corr}1}$. If h_1 and h_2 are farther from d (eventually they miss the correct class), then $\mu_1 << 1$ and $\mu_2 << 1$, the penalty is larger.

In general, let us denote by $\mathcal{L}_{\text{corr}k}$ the value of the penalty term when the first k vectors are not perfect one-hot ones. Then for every $i \in \{1, \ldots, k\}$

$$\mu_i := \varrho(h_i, d) = \varrho(h_i, h_j) < 1, \quad j = i+1, \ldots, N.$$

If we perturb the next vector(h_{k+1}), too, then only the terms

$$\begin{aligned}
\sum_{i=1}^{N} \varrho(h_{k+1}, h_i) - N\varrho(h_{k+1}, d) &= \sum_{i=1}^{k} \varrho(h_{k+1}, h_i) + \varrho(h_{k+1}, h_{k+1}) \\
&+ \sum_{i=k+2}^{N} \varrho(h_{k+1}, h_i) - N\varrho(h_{k+1}, d)
\end{aligned}$$

change in $\mathcal{L}_{\text{corr}k}$. The value of $\sum_{i=1}^{k} \varrho(h_{k+1}, h_i)$ is $\sum_{i=1}^{k} \mu_i$ and $\sum_{i=1}^{k} \mu_{i,k+1}$ in $\mathcal{L}_{\text{corr}k}$ and in $\mathcal{L}_{\text{corr}k+1}$, respectively, where $\mu_{i,k+1} := \varrho(h_i, h_{k+1}) < 1$ for $i = 1, \ldots, k$. $\varrho(h_{k+1}, h_{k+1}) = 1$ in both cases, while for $i = k+2, \ldots, N$ the correlation $\varrho(h_{k+1}, h_i)$ is 1 in $\mathcal{L}_{\text{corr}k}$ and $\mu_{k+1} := \varrho(h_{k+1}, d)$ in $\mathcal{L}_{\text{corr}k+1}$.

Hence, we obtain that

$$\begin{aligned}
\mathcal{L}_{\text{corr}k+1} &= \mathcal{L}_{\text{corr}k} + \sum_{i=1}^{k} \mu_{i,k+1} - \sum_{i=1}^{k} \mu_i \\
&+ (N - (k+1))\mu_{k+1} - (N - (k+1)) - N\mu_{k+1} + N \\
&= \mathcal{L}_{\text{corr}k} + (k+1) + \sum_{i=1}^{k} \mu_{i,k+1} - \sum_{i=1}^{k} \mu_i - (k+1)\mu_{k+1} \\
&= \mathcal{L}_{\text{corr}k} + (1 - \mu_{k+1}) + \sum_{i=1}^{k} \left[1 + \mu_{i,k+1} - \mu_i - \mu_{k+1}\right].
\end{aligned}$$

Similarly as before, if $\mu_{i,k+1} \approx 1$, for some i, i.e. h_{k+1} is close to h_i, and they are close to d as well, then $1 + \mu_{i,k+1} - \mu_i - \mu_{k+1} \approx 0$, so it does not change the penalty significantly. If the vectors h_i and h_{k+1} are close to each other, but farther from d, then $\mu_i << 1$ and $\mu_{k+1} << 1$, hence $1 + \mu_{i,k+1} - \mu_i - \mu_{k+1} > 0$, the penalty is larger. □

The complete loss function employed during training is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{corr}}, \tag{3.3}$$

where $\lambda \geq 0$ controls the influence of the correlation-based penalty term. Larger values of λ promote greater diversity among ensemble members, but if set too high, may adversely affect classification accuracy. Therefore, λ should be selected through hyperparameter tuning.

To facilitate efficient training, we derive the gradient of $\mathcal{L}_{\text{corr}}$ with respect to each softmax output component $h_{k,m}^{(n)}$ of the k-th CNN:

$$\frac{\partial \mathcal{L}_{\text{corr}}}{\partial h_{k,m}^{(n)}} = \frac{1}{M} \sum_{n=1}^{M} \left[\sum_{\substack{j=1 \\ j \neq k}}^{N} \frac{\partial C_{kj}^{(n)}}{\partial h_{k,m}^{(n)}} + N \frac{\partial C_{k,N+1}^{(n)}}{\partial h_{k,m}^{(n)}} \right].$$

The partial derivative of the Pearson correlation coefficient with

respect to X_m is given by:

$$\frac{\partial \varrho(X,Y)}{\partial X_m} = \frac{(Y_m - \overline{Y}) - \frac{\sum_{i=1}^{K}(X_i-\overline{X})(Y_i-\overline{Y})}{\sum_{i=1}^{K}(X_i-\overline{X})^2}(X_m - \overline{X})}{\sqrt{\sum_{i=1}^{K}(X_i-\overline{X})^2 \sum_{i=1}^{K}(Y_i-\overline{Y})^2}}.$$

These derivatives are implemented in the gradient flow to allow end-to-end training of the ensemble network with the proposed diversity-promoting objective.

When justifying the choice of correlation metric, it's important to highlight that the Pearson correlation coefficient offers several practical advantages in the context of ensemble training. First, it is computationally efficient and differentiable, making it well-suited for integration into gradient-based optimization pipelines. Second, it directly measures linear dependency between the softmax output vectors of ensemble members, which is often sufficient to capture redundancy in incorrect classifications. Its values are also easily interpretable within a bounded range $[-1, 1]$, providing a stable regularization signal across datasets and architectures.

Nonetheless, the Pearson correlation is limited to linear relationships. In principle, non-linear dependencies among ensemble members could be better captured using alternative measures, such as Spearman's rank correlation, mutual information, or KL-divergence. These metrics, however, are computationally more expensive and less straightforward to differentiate, which may hinder their scalability to large ensembles. Moreover, empirical evidence from our experiments suggests that linear correlation is already a strong and sufficient proxy for output redundancy. Investigating non-linear correlation penalties remains an interesting direction for future research, especially in scenarios where ensemble members exhibit complex non-linear interactions.

3.3 Architectures, Datasets and Evaluation Metrics

To comprehensively assess the efficacy of the proposed Pearson correlation-based ensemble regularization, we performed systematic experiments across multiple domains. This section details the architectural

setup of neural networks, the data sets used for evaluation, and the performance metrics used for quantitative and qualitative evaluation.

3.3.1 CNN Ensemble Architectures

The ensemble architecture is constructed by fusing the outputs of N CNNs via a shared $\mathcal{FC}$ layer, followed by a final softmax classifier. Each CNN serves as a backbone feature extractor and is adapted to the current classification task by replacing its original classification head with a custom $\mathcal{FC}$ layer that outputs L-dimensional logits, corresponding to the number of target classes.

The motivation behind using multiple CNNs in ensemble configuration stems from leveraging their complementary strengths and improving their individual classification performance. In particular, the backbone networks selected for this study: AlexNet[61], VGG16[50], ResNet50[49], MobileNetV2[54] and GoogLeNet Inception-v3 [48] have all demonstrated strong performance in both natural and medical image classification tasks [62].

Backbone Architectures.

- **AlexNet** consists of 5 convolutional layers, some of which are followed by max-pooling, and 3 fully connected layers, with the penultimate $\mathcal{FC}$ layer containing 4,096 neurons.
- **VGG16** comprises 13 convolutional layers using 3×3 kernels and is followed by 3 fully connected layers. Its architectural simplicity combined with depth makes it a robust feature extractor.
- **MobileNetV2** utilizes an inverted residual structure and lightweight depthwise separable convolutions. It starts with a convolutional layer of 32 filters and includes 19 bottleneck residual blocks.
- **ResNet50** includes 48 convolutional layers, one max-pooling, and one average-pooling layer. It employs residual identity mappings to mitigate vanishing gradients in deep architectures.
- **GoogLeNet Inception-v3** adopts parallel inception modules for multi-scale feature extraction and has been shown to be particularly effective in dermatological image classification.

Each CNN was either randomly initialized or fine-tuned from the pre-trained ImageNet weights. To construct the ensemble, the original final $\mathcal{FC}$ layers were removed and replaced with new $\mathcal{FC}$ layers that corresponded to the number of output classes. Then these were concatenated and passed through a joint fusion layer, which learns to integrate the predictions. The final classification is obtained by applying a softmax activation to the output of the fusion layer.

Two types of ensembles were studied:

- **Homogeneous Ensembles:** Composed of multiple instances of the same CNN architecture (e.g., 3×AlexNet, 3×VGG16), initialized independently. This setup is useful to observe how the penalty term $\lambda\mathcal{L}_{\text{corr}}$ affects diversity even when architectural variety is absent.

- **Heterogeneous Ensembles:** Combine different CNN architectures (e.g., AlexNet + ResNet50 + MobileNetV2 or AlexNet + VGG16 + Inception-v3). These are naturally more diverse due to architectural differences and are expected to benefit further from the penalty term.

All ensemble architectures were implemented in TensorFlow/Keras and trained end-to-end using backpropagation. The training objective includes the categorical cross-entropy loss and the proposed Pearson correlation-based penalty term, weighted by the hyperparameter λ. As detailed in Section 3.2.1, this formulation encourages the ensemble members to learn decorrelated representations while maintaining task-specific discriminative power.

Altogether, four ensemble configurations were constructed to enable comprehensive evaluation (see Figure 3.2). These include both homogeneous and heterogeneous setups, with and without the correlation penalty.

As we shall see in Section 3.4, these configurations enable a comparative analysis of how architectural diversity and penalization interact to influence overall classification accuracy and robustness.

3.3.2 Datasets and Preprocessing

To evaluate the effectiveness of our proposed ensemble training framework with the Pearson correlation-based penalty, we conducted a comprehensive set of experiments on four publicly available image datasets.

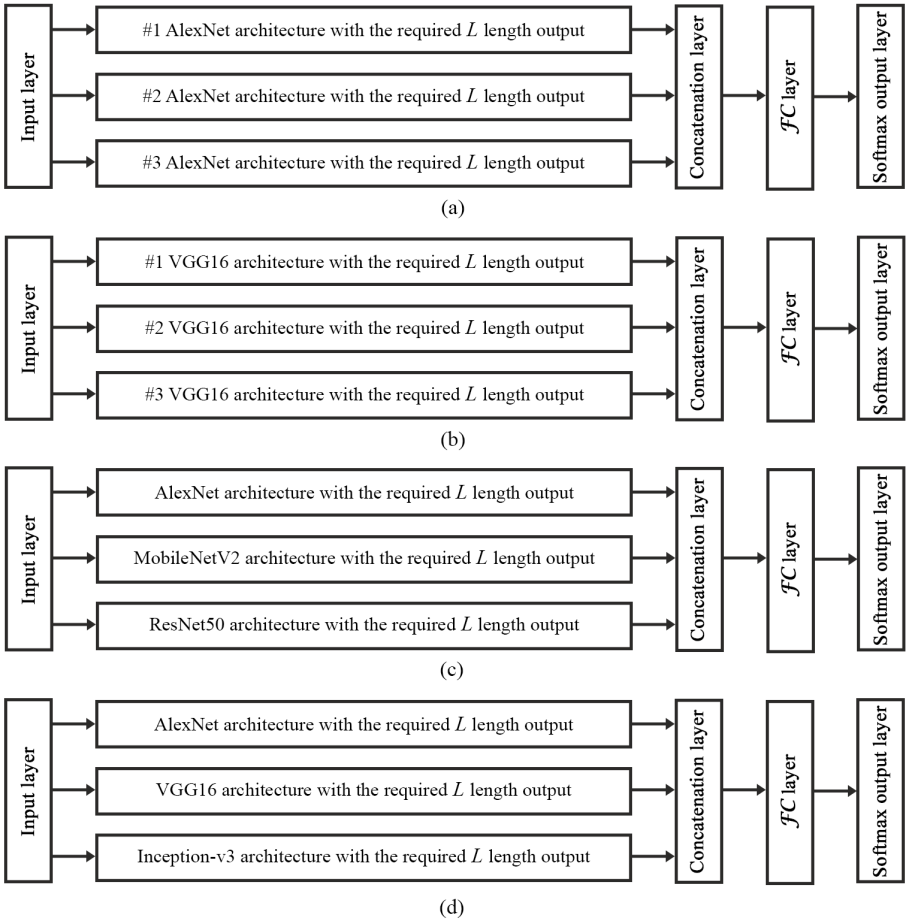

Figure 3.2: Ensemble networks composed by connecting different member architectures. Homogeneous ensembles (a) and (b) contain multiple instances of the same CNN, while heterogeneous ensembles (c) and (d) integrate diverse backbone architectures.

These datasets span both natural and clinical image domains and were selected to reflect classification tasks with varying levels of complexity, resolution, and class balance. Figure 3.3 illustrates representative samples from each dataset.

ISINI Dataset

The ISINI dataset, introduced by Roy *et al.* [63], contains 6,899 natural scene images distributed across eight categories: *airplane, car, cat,*

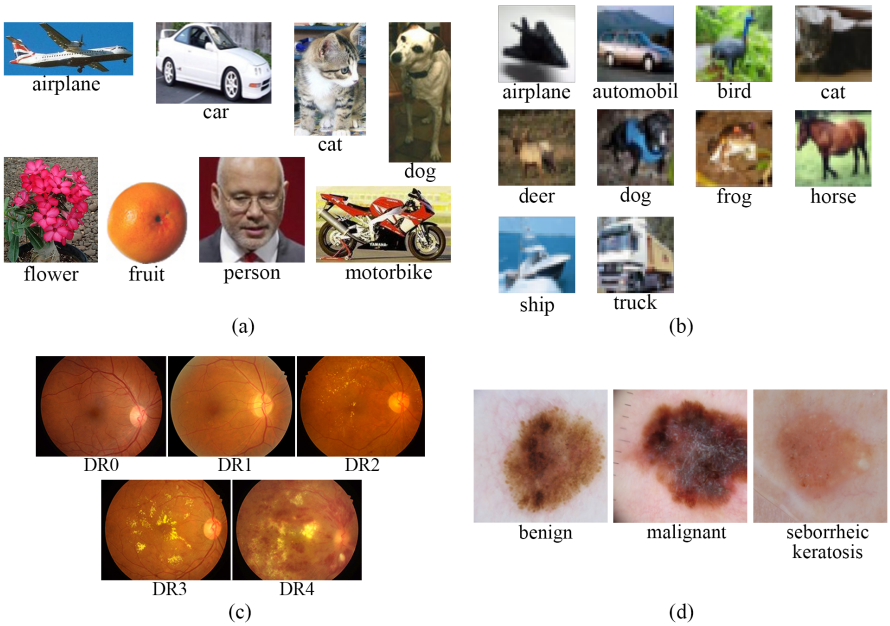

Figure 3.3: Sample images from the (a) ISINI, (b) CIFAR-10, (c) DR, and (d) ISIC image sets.

dog, flower, fruit, motorbike, and *person*. The number of samples per class is nearly balanced, ranging from 702 to 1,000 images, thus supporting stable training across all categories. The original image resolutions vary significantly (from 43×114 to $2,737 \times 2,229$ pixels), so we resized all images to a uniform resolution of 227×227 pixels for compatibility with standard CNN input layers. The dataset was partitioned into a training set of 5,519 images and a test set of 1,380 images, maintaining the original class distribution. Sample images are shown in Figure 3.3(a).

CIFAR-10 Dataset

The CIFAR-10 dataset [64], created by Krizhevsky *et al.*, is a benchmark for image classification and is widely used for evaluating CNN performance. It consists of 60,000 color images of 32×32 pixels, divided evenly into 10 classes (*airplane, automobile, bird, cat, deer, dog, frog, horse, ship*, and *truck*). The dataset is split into 50,000 training and 10,000 test images. Due to its small image resolution, we modified

the input layers of the CNNs rather than resizing the images. Sample images are shown in Figure 3.3(b).

Diabetic Retinopathy (DR) Dataset

To evaluate our method in the clinical domain, we constructed a comprehensive Diabetic Retinopathy (DR) classification dataset by combining three publicly available resources: the Kaggle DR dataset [65], the Messidor database [66], and the Indian Diabetic Retinopathy Image Dataset (IDRiD) [67]. Each image was labeled according to DR severity: *no DR (DR0)*, *mild (DR1)*, *moderate (DR2)*, *severe (DR3)*, and *proliferative DR (DR4)*.

The merged dataset contains 18,127 training and 4,529 test images, with the following class distributions: 9,975/2,493 for DR0, 2,085/520 for DR1, 4,537/1,134 for DR2, 757/189 for DR3, and 773/193 for DR4. Image resolutions vary widely from 400×315 to $5,184 \times 3,456$ pixels. Sample images are shown in Figure 3.3(c).

ISIC Skin Lesion Dataset

To further evaluate our method for medical image classification, we utilized the ISIC 2017 skin lesion dataset [68]. The training set contains 2,000 manually annotated dermoscopic images classified into three categories: *nevus (1,372 images)*, *melanoma (374 images)*, and *seborrheic keratosis (254 images)*. Evaluation was performed on a separate test set with 393 nevus, 90 melanoma, and 117 seborrheic keratosis samples. All images were resized to 224×224 pixels prior to training. Representative samples are shown in Figure 3.3(d).

Data Augmentation Strategy

For the ISINI, DR, and ISIC datasets, the number of available training images in certain classes is relatively limited. This poses a risk of overfitting, particularly when using large ensemble architectures. To mitigate this, we applied standard data augmentation techniques during training, in line with the recommendations for deep learning-based classification tasks [55]. The augmentations include:

- Random horizontal flipping,

- Image rotation with randomly chosen angles,
- Shearing transformations up to 20°,
- Random zoom in the range of $[0.8, 1.2]$.

These transformations increase the diversity of the training data and promote better generalization of the CNNs and their ensembles.

3.3.3 Training Protocol and Hyperparameters

All models were trained using the Adam optimizer with initial learning rate 10^{-5}, batch size 32, and a maximum of 100 epochs. Early stopping based on validation loss was employed to avoid overfitting. For ensemble training, we experimented with multiple values of the regularization coefficient $\lambda \in \{0, 0.5, 1.0, 5.0\}$.

The all proposed networks were trained using TensorFlow's multi-GPU `MirroredStrategy`, with weights distributed across NVIDIA TITAN RTX and RTX 2080 Ti GPUs.

3.3.4 Evaluation Metrics

To assess ensemble classification performance, we primarily relied on the following metrics:

- **Accuracy:** The overall proportion of correctly predicted samples, computed as:

$$\text{Accuracy} = \frac{1}{m} \sum_{i=1}^{m} \chi_{y^{(i)}}(\hat{y}^{(i)}),$$

  where $\hat{y}^{(i)}$ is the predicted label and $y^{(i)}$ the ground truth for the i-th sample.

- **Double/Triple Miss Count:** For ensembles, we additionally computed the number of test samples that were misclassified by at least two (double miss) or all (triple miss) ensemble members. These quantities serve as empirical indicators of model diversity: lower joint error rates indicate reduced correlation among member predictions.

Each experiment was repeated five times with different random seeds. The reported results include the mean and standard deviation of the accuracy and miss counts. To support qualitative analysis, confusion matrices and bar diagrams of ensemble prediction overlaps were also examined.

3.4 Experimental Results

This section presents a comprehensive evaluation of the proposed ensemble training method with the Pearson correlation-based penalty. Our primary objectives were (*i*) to validate whether the introduced penalty increases ensemble diversity and (*ii*) to quantify its impact on classification performance across natural and medical imaging datasets.

3.4.1 Evaluation on the ISINI Dataset

The ISINI dataset [63] contains eight balanced categories of natural images (e.g., airplane, cat, fruit), offering a relatively clean and low-noise classification scenario. This makes it an ideal candidate to test the impact of ensemble diversity on classification accuracy without additional complications from data imbalance or low-resolution inputs.

In this experiment, we utilized three independently initialized instances of AlexNet as ensemble members in a homogeneous setting. These models were trained together using a loss function that incorporated varying values of the diversity penalty coefficient, denoted as λ, with values of $\lambda \in \{0, 0.5, 1, 5\}$. We then compared the performance of these models to that of:

- A single AlexNet baseline,
- Classic ensemble fusion methods Simple Majority Voting (SMV), Weighted Majority Voting (WMV), and Averaging (AVE) using separately trained AlexNets.

As seen in Table 3.1, the proposed ensemble training method outperforms both the single AlexNet and the classic fusion techniques (SMV, WMV, AVE). Notably, even the unregularized ensemble ($\lambda = 0$) achieves better accuracy than the baseline and traditional ensemble methods.

Table 3.1: Classification accuracy of different setups of AlexNet on the ISINI set.

CNN architecture	value of λ	accuracy	total # of missed images	double missed	triple missed
single AlexNet	-	0.9328 ± 0.0029	92.6 ± 4.04	-	-
SMV	-	0.9465 ± 0.0047	74 ± 6.53	51.0 ± 4.12	27.8 ± 3.50
WMV	-	0.9462 ± 0.0041	74 ± 5.67	51.0 ± 4.12	27.8 ± 3.50
AVE	-	0.9504 ± 0.0021	68 ± 2.96	51.0 ± 4.12	27.8 ± 3.50
AlexNet multi	0	0.9543 ± 0.0013	63 ± 1.73	50.4 ± 3.42	27.6 ± 3.51
AlexNet multi	0.5	0.9636 ± 0.0021	50 ± 2.94	36.2 ± 3.99	24.0 ± 3.67
AlexNet multi	1	**0.9650 ± 0.0018**	48 ± 2.49	32.0 ± 4.99	25.0 ± 4.30
AlexNet multi	5	0.9617 ± 0.0013	52 ± 1.78	**31.4 ± 2.82**	**24.4 ± 2.96**

The introduction of the Pearson correlation-based penalty leads to consistent gains in both accuracy and diversity. With $\lambda = 0.5$ and $\lambda = 1$, the ensemble reaches its peak performance (96.3%), while the number of jointly misclassified samples ("double missed" and "triple missed") drops significantly compared to $\lambda = 0$ and the baseline ensembles.

Interestingly, when $\lambda = 5$, we observe a slight reduction in overall accuracy, although the number of double/triple missed samples continues to decrease. This confirms the regularization trade-off: excessive penalization may compromise accuracy by forcing overly divergent outputs.

The ISINI experiment demonstrates the effectiveness of the proposed ensemble architecture and the correlation-based penalty. It confirms that even when using clean, balanced datasets, encouraging diversity through penalization enhances both the robustness of individual models and the overall performance of the ensemble. A setting of $\lambda = 0.5$ or $\lambda = 1$ offers the best balance between diversity and accuracy.

3.4.2 Evaluation on the CIFAR-10 Dataset with AlexNet

The CIFAR-10 [64] dataset is a widely used benchmark in image classification, containing 60,000 color images of size 32×32 across 10

categories. Due to its relatively low resolution and higher inter-class ambiguity compared to ISINI, this dataset is particularly useful for evaluating how ensemble diversity affects performance in a more challenging visual context.

In this experiment, we employed three independently initialized instances of AlexNet in a homogeneous ensemble configuration. The models were trained jointly using the proposed loss function, with λ values ranging from 0 to 5 similarly as before. We also compared the results to single-model performance and classical ensemble methods.

Table 3.2: Classification accuracy of different setups of AlexNet on the CIFAR-10 set.

CNN architecture	value of λ	accuracy	total # of missed images	double missed	triple missed
single AlexNet	-	0.6431 ± 0.0056	3 569 ± 56	-	-
SMV	-	0.6308 ± 0.0024	3 692 ± 25	3 797 ± 362	2692 ± 130
WMV	-	0.6258 ± 0.0044	3 741 ± 44	3 797 ± 362	2692 ± 130
AVE	-	0.6554 ± 0.0079	3 446 ± 79	3 797 ± 362	2692 ± 130
AlexNet multi	0	0.6471 ± 0.0031	3 529 ± 31	3 093 ± 248	2 741 ± 79
AlexNet multi	0.5	0.6656 ± 0.0059	3 343 ± 59	2 124 ± 165	2 290 ± 48
AlexNet multi	1	**0.6683 ± 0.0035**	3 316 ± 35	**2 077 ± 254**	**2 294 ± 45**
AlexNet multi	5	0.6646 ± 0.0042	3 354 ± 42	2 130 ± 261	2 295 ± 119

Table 3.2 shows that classical ensemble fusion methods (SMV, WMV) do not improve over the single model; in fact, they slightly underperform. The simple averaging method (AVE) yields a modest gain over the baseline, highlighting the difficulty of ensemble learning in high-variance, low-resolution tasks without explicit regularization.

By contrast, our joint training method—especially with $\lambda = 0.5$ and $\lambda = 1$, clearly improves both the accuracy and robustness. The best-performing configuration ($\lambda = 1$) achieves an accuracy of 66.83%, outperforming all baselines. Furthermore, the number of jointly misclassified samples is significantly lower, indicating enhanced diversity between the ensemble members.

Notably, while increasing λ from 0 to 1 results in consistent improvement, pushing λ to 5 slightly decreases accuracy without further gains in diversity. This again highlights the trade-off between too much diversity (which may lead to desynchronization) and insufficient penalization (which may lead to redundancy).

These results confirm that even with relatively weak backbone models such as AlexNet on CIFAR-10, our correlation-based ensemble method consistently outperforms traditional training and naive fusion strategies. The model benefits most from moderate diversity enforcement ($\lambda = 1$), striking an effective balance between independent learning and ensemble coherence.

3.4.3 Evaluation on the CIFAR-10 Dataset with VGG16

VGG16 is a deeper architecture with greater representational capacity then AlexNet, making it a strong candidate for evaluating how architectural capacity interacts with ensemble diversity mechanisms. Using the same CIFAR-10 dataset, we assessed whether diversity promotion via Pearson correlation penalization could further boost the already solid performance of VGG16-based ensembles.

We constructed a homogeneous ensemble composed of three independently initialized VGG16 models and trained them jointly under varying penalty strengths ($\lambda \in \{0, 0.5, 1, 5\}$). As before, we compared our approach to a single VGG16 model and to classical fusion techniques.

Table 3.3: Classification accuracy of different setups of VGG16 on the CIFAR-10 set.

CNN architecture	value of λ	accuracy	total # of missed images	double missed	triple missed
single VGG16	-	0.8051 ± 0.0016	1 949 ± 16	-	-
SMV	-	0.7800 ± 0.0243	2 199 ± 243	2 036 ± 274	660 ± 48
WMV	-	0.8197 ± 0.0111	1 802 ± 111	2 036 ± 274	660 ± 48
AVE	-	0.8331 ± 0.0060	1 668 ± 60	2 036 ± 274	660 ± 48
VGG16 multi	0	0.8743 ± 0.0013	1 256 ± 13	1 537 ± 318	661 ± 49
VGG16 multi	0.5	0.8923 ± 0.0024	1 077 ± 25	812 ± 31	540 ± 13
VGG16 multi	1	**0.8931 ± 0.0018**	**1 068 ± 18**	**818 ± 30**	**526 ± 30**
VGG16 multi	5	0.8892 ± 0.0027	1 108 ± 27	908 ± 44	545 ± 29

Table 3.3 highlights several key findings. First, the single VGG16 model outperforms all AlexNet-based configurations (cf. Section 3.4.2), underlining the strength of deeper architectures. However, even with

such a strong baseline, our proposed ensemble strategy yields substantial improvements.

Compared to the single model, the unregularized ensemble ($\lambda = 0$) increases accuracy by almost 7%, indicating that joint training alone is beneficial. More importantly, adding the correlation-based penalty with $\lambda = 0.5$ or $\lambda = 1$ leads to additional gains, reaching a maximum of 89.31% accuracy and a sharp drop in jointly missed samples.

As in previous experiments, setting $\lambda = 5$ causes a slight drop in both accuracy and diversity metrics, supporting the conclusion that excessive penalization may interfere with learning relevant shared patterns.

The VGG16 experiment on CIFAR-10 further validates the efficacy of diversity-driven training, even in the context of strong base learners. Our ensemble method consistently improves upon classical fusion strategies, with $\lambda = 1$ again providing an optimal trade-off between accuracy and diversity.

3.4.4 Evaluation on the Diabetic Retinopathy Dataset

The DR dataset presents a substantially more complex and clinically significant classification task, involving fundus images labeled according to disease severity. The dataset poses unique challenges due to the high intra-class variability and subtle visual features that distinguish adjacent severity levels.

In this experiment, we constructed a heterogeneous ensemble using three different CNN architectures: AlexNet, MobileNetV2, and ResNet50. These were chosen to represent varying levels of architectural depth and complexity. As in previous tests, we trained the ensemble jointly under different values of the correlation penalty coefficient $\lambda \in \{0, 0.5, 1, 5\}$ and compared the results with single CNN models and classic fusion baselines.

As shown in Table 3.4, none of the individual CNN architectures, including deeper models like ResNet50, were able to surpass 66% accuracy. This highlights the complexity of the DR classification task and indicates that single-network solutions may struggle to capture its high intra-class variability and subtle visual features. Traditional ensemble fusion strategies (SMV, WMV, AVE) offered only limited

Table 3.4: Classification accuracy of the ensemble of CNNs on the DR dataset.

CNN architecture	value of λ	accuracy	total # of missed images	double missed	triple missed
AlexNet	-	0.5760 ± 0.0035	1 920 ± 16	-	-
MobileNetV2	-	0.6114 ± 0.0101	1 760 ± 46	-	-
ResNet50	-	0.6515 ± 0.0193	1 578 ± 87	-	-
SMV	-	0.6250 ± 0.0163	1 698 ± 73	600 ± 50	1 110 ± 111
WMV	-	0.6384 ± 0.0182	1 637 ± 82	600 ± 50	1 110 ± 111
AVE	-	0.6462 ± 0.0182	1 602 ± 82	600 ± 50	1 110 ± 111
CNN ensemble	0	0.6584 ± 0.0170	1 547 ± 77	580 ± 152	1 136 ± 86
CNN ensemble	0.5	**0.6707 ± 0.0081**	**1 491 ± 36**	**460 ± 140**	**883 ± 21**
CNN ensemble	1	0.6662 ± 0.0034	1 511 ± 15	651 ± 128	866 ± 27
CNN ensemble	5	0.6599 ± 0.0099	1 540 ± 45	742 ± 224	913 ± 122

improvements over the weaker models (AlexNet, MobileNetV2), but notably underperformed compared to the strongest individual model, ResNet50.

This suggests that naively combining heterogeneous models while increasing architectural diversity may degrade ensemble performance when weaker networks dilute the stronger ones' predictions. In this setting, classical diversity-inducing strategies fail to leverage their theoretical benefits.

By contrast, our proposed joint training method shows clear performance gains, especially with moderate correlation regularization. The ensemble with $\lambda = 0.5$ achieved the highest accuracy (67.07%), while also minimizing the number of jointly misclassified samples. These results demonstrate that enforcing output diversity in a principled manner allows heterogeneous ensembles to outperform all baselines, including the best single model, by encouraging complementary rather than redundant or detrimental behavior across networks.

As before, a penalty value of $\lambda = 5$ led to a slight degradation in accuracy, again reinforcing the notion that overly strong regularization can hinder convergence in high-variance environments.

The DR dataset results confirm that the proposed ensemble method remains effective in heterogeneous settings and on challenging clinical data. The method not only outperforms all baselines, but also demonstrates robustness in learning complementary decision bound-

aries among diverse CNN backbones when diversity is appropriately enforced.

3.4.5 Evaluation on the ISIC Skin Lesion Dataset

The ISIC dataset comprises dermatoscopic images of pigmented skin lesions, annotated with melanoma and non-melanoma classes. It poses a demanding classification challenge due to intra-class variability, subtle textural patterns, and limited data quantity. Thus, it offers a valuable test case for evaluating whether ensemble diversity can improve classification performance in a real-world medical context.

In this experiment, we constructed a heterogeneous ensemble of three distinct CNNs—AlexNet, VGG16, and Inception-v3—based on their previously demonstrated performance in dermatological image analysis [62]. The models were trained jointly using our correlation-penalized loss function, and the results were compared to individual CNNs and standard ensemble techniques.

Table 3.5: Classification accuracy of the ensemble of CNNs on the ISIC dataset.

CNN architecture	value of λ	accuracy	total # of missed images	double missed	triple missed
AlexNet	-	0.6858 ± 0.0011	188 ± 1	-	-
VGG16	-	0.7033 ± 0.0014	178 ± 1	-	-
Inception-v3	-	0.7099 ± 0.0016	174 ± 1	-	-
CNN ensemble	0	0.7011 ± 0.0239	179 ± 14	99 ± 30	106 ± 7
CNN ensemble	0.5	0.7294 ± 0.0178	162 ± 11	102 ± 29	94 ± 13
CNN ensemble	1	**0.7355 ± 0.0111**	**158 ± 6**	150 ± 31	**57 ± 37**
CNN ensemble	5	0.7011 ± 0.0267	179 ± 16	177 ± 25	**39 ± 8**

Table 3.5 reveals that the best individual CNN, namely Inception-v3, reaches an accuracy of 70.99%, slightly outperforming AlexNet and VGG16. The unregularized ensemble ($\lambda = 0$) does not yet surpass this value, but once the correlation penalty is introduced, the ensemble begins to outperform all baselines.

With $\lambda = 0.5$ and especially $\lambda = 1$, the ensemble achieves improved accuracy and fewer total missed classifications. Notably, $\lambda = 1$ yields

the highest accuracy (73.55%) and one of the lowest triple miss counts, supporting the conclusion that moderate penalization enhances both prediction confidence and complementarity between networks.

Interestingly, although $\lambda = 5$ achieves the fewest triple misses (39), the overall accuracy regresses to that of the unregularized case. This suggests that over-penalization may sacrifice decision consensus in favor of artificial diversity.

On the ISIC dataset, the proposed ensemble framework demonstrates its ability to adapt to complex medical imagery. The combination of architectural heterogeneity and learned diversity not only boosts classification accuracy but may also improve safety in clinical deployment by reducing agreement on erroneous predictions. These findings underscore the potential of the method in diagnostic support systems, especially with carefully tuned regularization.

3.4.6 Effect of the Penalty Term

To investigate the influence of the Pearson correlation-based penalty on ensemble behavior, we conducted a systematic study by varying the hyperparameter λ across a broad range. This hyperparameter controls the relative strength of the diversity-promoting penalty term introduced in Section 3.2.2.

Figure 3.4 illustrates the empirical trade-off observed between ensemble accuracy and prediction correlation across datasets. As λ increases from 0 to 1.0, classification accuracy improves, reaching a maximum in most cases around $\lambda = 0.5$ or 1.0. This performance gain coincides with a notable reduction in output correlation among ensemble members, suggesting that the penalty effectively induces functional diversity without compromising discriminative power.

However, beyond a critical value ($\lambda = 5.0$), the benefits start to diminish. In some cases, further increasing λ leads to degraded performance, which we attribute to over-regularization: the models are encouraged to diverge excessively, possibly at the cost of underfitting or inconsistent representation learning.

These findings confirm the hypothesis that ensemble performance benefits most from moderate decorrelation. The penalty acts as a soft constraint guiding ensemble members toward complementary decision boundaries, rather than enforcing strict independence, which could be detrimental.

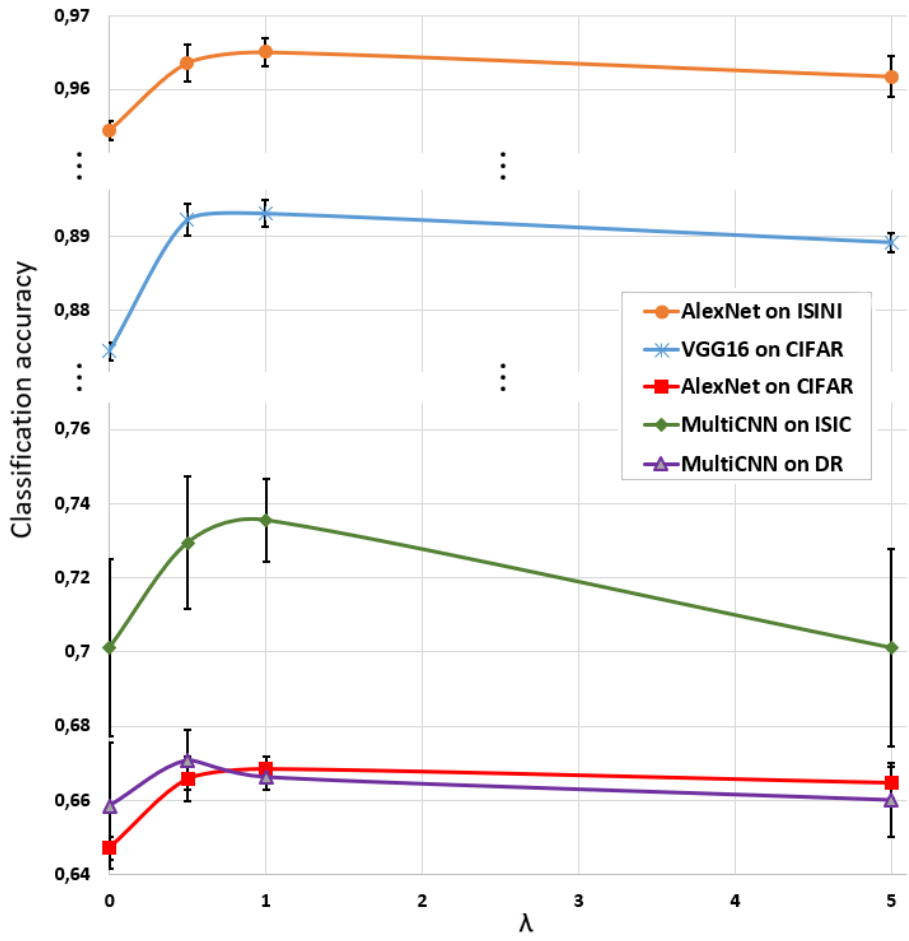

Figure 3.4: Trade-off between accuracy and ensemble correlation for varying λ values on different datasets.

3.4.7 Diversity Analysis

To provide a more concrete evaluation of ensemble diversity, we analyzed the distribution of misclassification overlaps among the ensemble members. Specifically, we computed two metrics:

- **Double miss:** A sample incorrectly predicted by at least two members.
- **Triple miss:** A sample incorrectly predicted by all three ensemble members.

These measures reflect the degree of functional similarity among the models. High double or triple miss counts indicate redundant behavior, while lower values suggest more diverse and complementary predictions.

Our evaluation, summarized in Table 3.5, shows a clear trend: increasing λ from 0 to 1 significantly reduces the number of triple misses (from 106 to 57), while simultaneously increasing classification accuracy (from 70.11% to 73.55%). The number of double misses remains relatively stable or increases slightly with higher λ, which can be attributed to reduced triple overlaps splitting into distinct double errors.

This effect is visually confirmed in Figure 3.5, which presents bar diagrams for the ISIC dataset under different λ values. As λ increases, the overlap between misclassified samples decreases, reflecting a reduction in error correlation and an improvement in ensemble robustness.

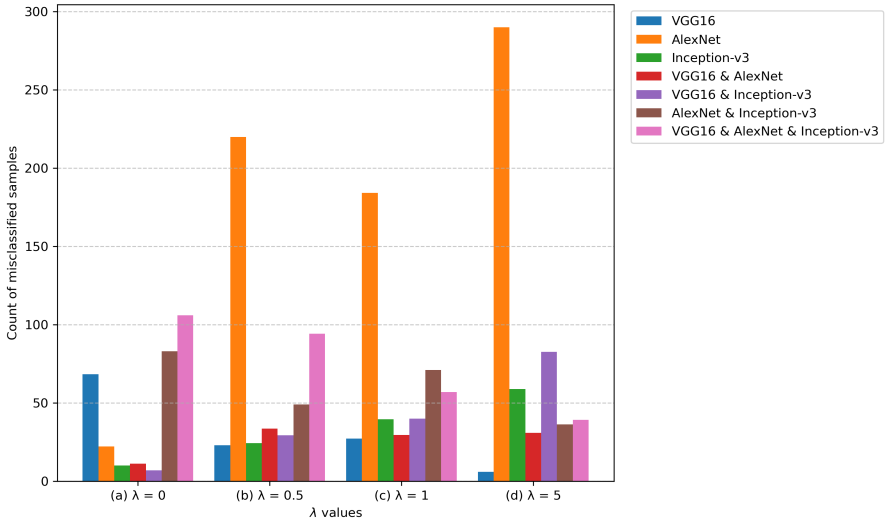

Figure 3.5: Bar diagrams of misclassified samples under different λ values (a) $\lambda = 0$, (b) $\lambda = 0.5$, (c) $\lambda = 1$, and (d) $\lambda = 5$. (ISIC dataset).

This analysis supports the interpretation that the penalty encourages ensemble members to specialize on different subregions of the input space, thereby reducing joint failure modes—a highly desirable property in safety-critical applications.

3.5 Claims

Claim 1 *A cohesive and trainable CNN ensemble architecture integrates a Pearson correlation penalty, computed on the softmax outputs of the member networks, into the joint loss function. This design enables simultaneous optimization of all ensemble members, explicitly discourages redundant misclassifications, and enhances prediction diversity. The λ parameter in the joint loss function explicitly controls the trade-off between classification accuracy and ensemble diversity.*

Reasoning. The proposed ensemble model connects multiple CNNs via a shared fully connected layer, allowing simultaneous backpropagation across all components. The loss function integrates a Pearson correlation-based penalty that explicitly discourages output redundancy, particularly on incorrect predictions. This was proofed in 1. This differentiable regularization mechanism enables functional diversity to be optimized jointly with classification accuracy, offering advantages over traditional offline ensemble schemes where diversity cannot be directly enforced during training. The total loss function $\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \cdot \mathcal{L}_{\text{corr}}$ allows λ to act as a regularization strength that balances two competing objectives: accurate predictions and low inter-model correlation. Experimental results across different datasets confirm that moderate values of λ (e.g., 0.5 or 1) yield optimal performance, whereas extreme values either under-regulate (leading to redundancy) or over-regularize (leading to performance degradation). This tunability makes the method adaptable to task-specific constraints such as uncertainty quantification or resource limitations. □

Claim 2 *The proposed method outperforms classical ensemble strategies such as majority voting, averaging, and weighted fusion across diverse datasets, including clinical image domains.*

Reasoning. Empirical evaluation in four datasets, including natural images (ISINI, CIFAR-10) and clinical images (DR, ISIC), demonstrates consistent performance gains over standard fusion techniques. The method surpasses majority voting, weighted voting, and averaging by integrating ensemble-level training and explicit diversity enforcement. The improvements are statistically significant, and especially pronounced in clinically relevant settings where model disagreement

can reflect meaningful uncertainty and yield more robust predictions in ambiguous cases. □

3.6 Conclusions

This chapter summarizes the findings and practical implications of a novel ensemble training approach that promotes output diversity among CNNs through a Pearson correlation-based penalty term. By augmenting the standard cross-entropy loss with a correlation penalty $\mathcal{L}_{\text{corr}}$, we encourage ensemble members to make uncorrelated incorrect predictions, which improves both robustness and overall accuracy. The effect of this term is controlled via a hyperparameter λ, which tunes the balance between individual model accuracy and inter-model diversity.

Our results across multiple datasets, including both natural (ISINI, CIFAR-10) and clinical (DR, ISIC) domains, confirm the following key points:

- **Improved ensemble accuracy:** Joint training with moderate penalty ($\lambda \in [0.5, 1]$) consistently outperformed both individual CNNs and classical ensemble aggregation strategies (e.g., majority voting, averaging).
- **Increased output diversity:** The number of jointly misclassified samples (double/triple misses) decreased notably when the penalty term was active. This reflects a reduction in correlated errors, which is especially important in safety-critical settings.
- **Model-agnostic design:** The approach generalizes across CNN architectures, including homogeneous (e.g., multiple AlexNets) and heterogeneous (e.g., AlexNet + VGG16 + Inception-v3) ensembles, without requiring changes to the underlying networks.
- **Sensitivity to λ:** As shown by empirical curves (e.g., Figure 3.4), excessively large λ values can degrade performance, as the model prioritizes decorrelation over correct classification. Thus, λ must be treated as a tunable hyperparameter similar to the learning rate.

These findings demonstrate that encouraging structured diversity is not only theoretically sound but also practically effective. The proposed penalty term integrates smoothly into standard backpropagation

and can be readily applied to modern deep learning pipelines using existing frameworks (e.g., TensorFlow/Keras).

Despite its advantages, the current method has several practical limitations:

- **Computational overhead:** Ensemble training with multiple large CNNs requires significant GPU memory and time (e.g., over 16 hours with 500M+ parameters), making it less suitable for edge devices or rapid prototyping.
- **Overfitting risk:** When ensemble members are pretrained or initialized similarly and trained jointly on small datasets, overfitting may still occur despite the penalty. Data augmentation alleviates but does not eliminate this issue.
- **Penalty granularity:** The Pearson correlation penalty operates on softmax outputs, which may not capture deeper structural similarities (e.g., in feature representations or internal activations). This limits its discriminative ability in complex settings.
- **Manual tuning of λ:** There is currently no principled way to select the optimal penalty strength; λ must be tuned manually via grid search, adding to the training burden.

In conclusion, this work provides a theoretically grounded and practically validated method to increase ensemble diversity via direct penalization of correlated errors. The resulting improvement in accuracy and robustness, especially on medical datasets, supports its relevance for real-world deployment in high-stakes applications. Future research may explore extensions to feature-level diversity constraints or adaptive penalty mechanisms, but such directions require further investigation beyond the scope of this dissertation.

Chapter 4

Physically Based Fog Modeling in Inhomogeneous Media

4.1 Motivation

Atmospheric scattering phenomena—such as fog, mist, and haze—are prevalent in outdoor environments and introduce significant challenges for both human vision and computer vision systems. These effects emerge from complex light–particle interactions, including absorption, single and multiple scattering, and forward-directed radiative transfer. In safety-critical scenarios—such as autonomous driving, drone navigation, or surveillance—robust understanding of scene geometry under degraded visibility is essential. However, collecting large-scale, annotated datasets of foggy images under controlled conditions is expensive, logistically complex, and often infeasible.

To overcome this limitation, synthetic fog generation has become a key component in dataset augmentation workflows. By simulating realistic fog effects over clear-weather imagery, one can expose vision systems to adverse conditions without requiring physical data acquisition. Nevertheless, current fog synthesis techniques are constrained by either overly simplistic physical assumptions or limited structural controllability.

Classical methods based on Koschmieder's law [69, 70] provide closed-form solutions for radiance attenuation under homogeneous fog with isotropic scattering. These models are computationally efficient and widely used in automotive vision and real-time applications [71, 72]. However, they assume spatially uniform media and constant

airlight, which prevents them from capturing essential visual phenomena such as depth-dependent fog density, visibility gradients, or directional light scattering.

In contrast, data-driven methods, including CycleGAN [73] and StarGAN [74, 75], generate fog effects via domain translation or adversarial image synthesis. While these models produce visually plausible results, they lack physical grounding, offer limited parameter control, and often introduce geometric inconsistencies or style-based artifacts [76, 77, 78]. Moreover, their stochastic nature makes them unsuitable for safety-critical use cases where reproducibility and interpretability are crucial.

Recent advances in radiative modeling [79, 80, 81, 82] have demonstrated the power of the RTE in describing the propagation of light through scattering media. However, full 3D volumetric solvers (e.g., [83, 84]) remain computationally prohibitive for image-space fog synthesis, and often require multi-view input or sensor-level simulation.

To bridge this gap, we propose a novel image-space fog synthesis method based on the discretized Radiative Transfer Equation. The RTE framework allows for modeling of spatially inhomogeneous fog densities and anisotropic scattering using the Henyey–Greenstein phase function. By discretizing both spatial and angular domains, we construct a numerically tractable, recursive formulation that approximates light transport along each image ray, incorporating depth-dependent extinction and directional in-scattering.

Our implementation uses tensor-based operations and precomputed angular scattering matrices to achieve significant computational acceleration—from $\mathcal{O}(n^5)$ to $\mathcal{O}(n)$ per iteration—without compromising physical realism. This balance enables high-fidelity fog synthesis that is structurally consistent with scene geometry while remaining suitable for large-scale data generation.

In addition, we introduce a curated dataset of foggy–cloudy image pairs, annotated with subjective fog density scores. These data allow for gradient-based calibration of the model's extinction and scattering parameters, following approaches similar to [85, 86]. This training scheme ensures that the model can adapt to real-world atmospheric conditions and generalize across varied scenes.

Quantitative evaluations demonstrate the superiority of our method over Koschmieder and GAN-based baselines. In particular, our approach yields up to 42% improvement in Fréchet Inception Distance

(FID) [87] and a 21% increase in Pearson correlation with respect to real foggy images. These results highlight the benefits of combining physical modeling with learnable, image-aware parameterization.

In summary, our motivation is grounded in the need for a fog synthesis method that is (i) physically interpretable, (ii) structure-preserving, and (iii) computationally feasible. The discretized RTE framework provides a principled approach that addresses the limitations of both classical and generative methods, and supports realistic, controllable, and reproducible simulation of inhomogeneous atmospheric scattering in image space.

The method and algorithmic framework introduced in this chapter for physically grounded fog simulation, including the discretization of the Radiative Transfer Equation and the associated optimization strategy, were published in our article titled "Generation of Synthetic Non-Homogeneous Fog by Discretized Radiative Transfer Equation" [88]. The results reported here reflect a refined presentation of the original findings, with added emphasis on the theoretical formulation and evaluation benchmarks relevant to realistic image augmentation in adverse weather conditions.

4.2 Theoretical Background: Radiative Transfer Equation

The propagation of light in foggy or turbid media is governed by the principles of radiative transfer theory. This framework models how electromagnetic radiation interacts with a participating medium, accounting for absorption, scattering, and emission processes. In the context of atmospheric phenomena such as fog, this interaction is mathematically described by the RTE, which provides a continuous, integro-differential formulation of radiance along a given path.

The general form of the RTE in a monochromatic, stationary and emission-free setting is given by:

$$\frac{\partial L(r,\sigma)}{\partial r} = -K(r,\sigma)L(r,\sigma) + K_s(r,\sigma)\int_{\mathbb{S}^2} L(r,\omega)\,\phi(\omega,\sigma)\,d\omega,$$

where:

- $L(r,\sigma)$ denotes the radiance at position r in direction σ,

- $K(r, \sigma)$ is the extinction coefficient (comprising both absorption and scattering),
- $K_s(r, \sigma)$ is the scattering coefficient,
- $\phi(\omega, \sigma)$ is the phase function describing angular scattering probability from direction ω to σ,
- and $\mathbb{S}^2$ denotes the unit sphere representing all possible directions.

This formulation allows modeling of light attenuation (via K), as well as in-scattering contributions from all directions (via the integral term). Importantly, the RTE is capable of describing highly anisotropic and inhomogeneous media, which makes it particularly well-suited for simulating realistic fog.

For practical applications and computational tractability, the RTE is often simplified under assumptions of homogeneity and isotropy. The most well-known simplification is the Koschmieder model, which assumes:

1. constant scattering and absorption coefficients ($K, K_s \in \mathbb{R}$),
2. constant in-scattered radiance $L_{\text{in}} \approx \text{const}$.

Under these assumptions, the radiance at distance d can be computed analytically as:

$$L(d) = L_0 e^{-Kd} + L_{\text{air}}(1 - e^{-Kd}),$$

where L_0 is the initial radiance at $d = 0$, and L_{air} is the background airlight component representing the integrated in-scattered contribution from the atmosphere:

$$L_{\text{air}} = \frac{K_s}{K} \int_{\mathbb{S}^2} L_{\text{in}}(p(0), \omega)\, \psi(\sigma, \omega)\, d\omega.$$

Here d denotes the propagation distance along a ray. In the general case, this distance is not constant but depends on the viewing direction, i.e. $d(\sigma)$ is a depth function assigning to each $\sigma \in \mathbb{S}^2$ the distance from the camera to the first object along that ray. In image-based settings, $d(\sigma)$ corresponds to a per-pixel depth map, i.e. a matrix of depth values.

While the Koschmieder model is suitable for homogeneous fog simulation in real-time systems, it fails to capture spatially varying densities or directional scattering effects present in real-world scenarios.

To overcome this limitation, our method is built upon the general RTE and explicitly supports:

- spatially varying extinction and scattering parameters $K(r, \sigma)$, $K_s(r, \sigma)$,
- anisotropic scattering via the Henyey-Greenstein phase function:

$$\phi(\omega, \sigma) = \frac{1}{4\pi} \cdot \frac{1 - g^2}{(1 - 2g\langle P_\omega, P_\sigma \rangle + g^2)^{3/2}},$$

  where $g \in [0, 1)$ is the asymmetry factor and $\langle P_\omega, P_\sigma \rangle$ is the cosine of the angle between incoming and outgoing directions.

This theoretical foundation enables the physically grounded simulation of fog, providing a basis for the discretized numerical algorithm presented in the following section.

4.2.1 Toward Inhomogeneous Fog Modeling

Realistic fog in natural environments is rarely homogeneous. In such scenarios, the assumption of constant coefficients fails, and the full RTE must be considered. This dissertation adopts a discretized and iterative approach to solve the RTE under inhomogeneous and anisotropically scattering conditions. This enables more accurate modeling of fog's spatial variability and complex light interactions, forming the basis of our synthetic fog rendering algorithm described in the following sections.

4.3 Overview of the Algorithm

The proposed fog synthesis framework models the attenuation and scattering of light in a foggy atmosphere by discretizing the RTE in both spatial and angular domains. The algorithm takes as input a clear-weather image along with its corresponding depth map and produces a physically plausible foggy version of the same scene.

The key idea is to simulate, for each image pixel, the amount of direct transmission and in-scattered radiance that would reach the camera under a given fog distribution. To this end, we divide the line-of-sight between the camera and each scene point into uniformly spaced depth layers and recursively apply an approximation of the RTE.

The algorithm proceeds as follows:

1. **Preprocessing.** Given an input RGB image I and a per-pixel depth map d, and compute directional parameters using a spherical coordinate system.

2. **Initialization.** We initialize the radiance tensor $\tilde{L}_0$ using the original image, assuming no fog is present at depth $s = 0$.

3. **Discretization.** For each direction σ, the radial interval $s \in [0, d(\sigma)]$ is discretized into n uniform steps of size $\Delta s = d(\sigma)/n$. In parallel, the angular domain $\sigma \in \mathbb{S}^2$ is two-dimensional and is partitioned into an $n \times n$ of directions. This separation ensures that depth sampling and directional sampling are treated consistently. This forms the basis for recursive radiance evaluation.

4. **Recursive Evaluation.** For each depth step $j \in \{0, \ldots, n-1\}$, we compute the attenuated radiance $\tilde{L}_j$ at distance $s = jd(\sigma)/n$ using:

   - exponential decay due to extinction,
   - accumulation of in-scattered radiance from all directions weighted by the phase function,
   - spatially varying extinction and scattering coefficients.

5. **Fog Image Synthesis.** After the final iteration $j = n - 1$, the radiance tensor $\tilde{L}_{n-1}$ encodes the fog-attenuated appearance for each pixel. This tensor is projected back into a standard RGB image format.

Our method allows for the simulation of spatially inhomogeneous fog (i.e., depth-dependent coefficients) and anisotropic scattering (e.g., forward-directed), making it significantly more expressive than classical fog models. Additionally, the algorithm is implemented in a tensor-based format, which enables efficient computation and integration into differentiable pipelines.

The following sections provide the full mathematical and algorithmic details of this procedure.

4.3.1 Discretization of the Radiative Transfer Equation

To model the propagation of light through non-homogeneous and anisotropic fog in a way that is computationally manageable, we use a discretized version of the RTE. This approach enables us to generate synthetic fog that maintains physically consistent depth-dependent attenuation and directionally varying scattering.

Angular and Spatial Discretization

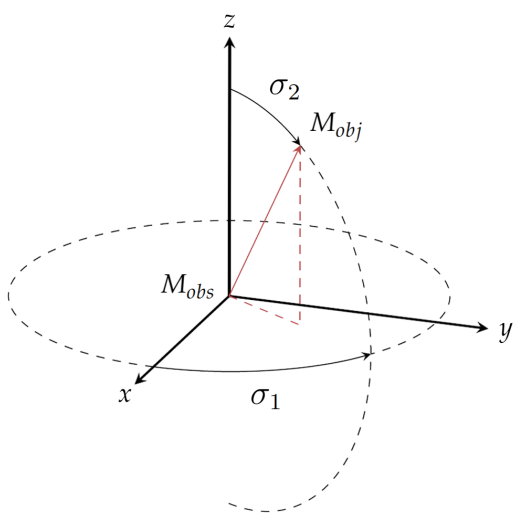

Figure 4.1: Geometric configuration of the observer M_{obs} and the object M_{obj} in spherical coordinates defined by azimuth σ_1 and elevation σ_2.

Let the observer (camera) be located at point $M_{\text{obs}} \in \mathbb{R}^3$, and consider the viewing direction $\sigma = (\sigma_1, \sigma_2) \in [0, \pi] \times [0, \frac{\pi}{2}]$ in spherical coordinates, where σ_1 denotes the azimuth and σ_2 the elevation angle (as Figure 4.1 shows). The associated unit direction vector $P_\sigma \in \mathbb{S}^2$ is given by:

$$P_\sigma = (\cos\sigma_1 \sin\sigma_2,\ \sin\sigma_1 \sin\sigma_2,\ \cos\sigma_2)\,.$$

The object point M_{obj} at distance d in direction σ is defined as:

$$M_{\text{obj}} = M_{\text{obs}} - dP_{\sigma}.$$

We discretize both the angular and radial domains. The angular domain is discretized into $n \times n$ directions:

$$\sigma_1 = \frac{k_1\pi}{n}, \quad \sigma_2 = \frac{k_2\pi}{2n}, \quad \text{for } k_1, k_2 \in \{0, \ldots, n-1\}.$$

At the initialization stage, we define the first layer:

$$\tilde{L}_0(k_1, k_2) := L(0, \sigma), \quad \text{where } \sigma_1 = \frac{k_1\pi}{n}, \quad \sigma_2 = \frac{k_2\pi}{2n}. \tag{4.1}$$

Here, $n \in \mathbb{N}$, $k_1, k_2 \in \{0, 1, \ldots, n-1\}$, and $\sigma \in [0, \pi] \times [0, \frac{\pi}{2}]$. This initial matrix represents the radiance in direction σ as seen in a clear atmosphere, and thus serves as a fog-free reference image.

Directional sampling in spherical coordinates follows this discretization: $\sigma_1 = 0$ points right, π left, $\sigma_2 = 0$ upwards, and $\pi/2$ forward. Any arbitrary direction σ can be approximated via the grid $\sigma_1 = \frac{k_1\pi}{n}, \sigma_2 = \frac{k_2\pi}{2n}$. In our discretization scheme, not only the angular domain σ is discretized, but also the radial distance along each direction. Specifically, we represent the per-direction distance field as a matrix $d \in \mathbb{R}^{n\times n}$, where each entry corresponds to the distance to the object surface along a given sampled direction σ.

We now define tensors indexed by $j = 1, \ldots, n-1$, intended to approximate $\tilde{L}_j(k_1, k_2)$, i.e., the radiance after j steps. The discretization is bounded as:

$$\left|\frac{dj}{n} - r\right| \leq \frac{1}{n}, \quad \left|\frac{k_1\pi}{n} - \sigma_1\right| \leq \frac{\pi}{n}, \quad \left|\frac{k_2\pi}{2n} - \sigma_2\right| \leq \frac{\pi}{2n}. \tag{4.2}$$

Recursive Approximation of the RTE

We assume that $L(r, \sigma)$ is sufficiently smooth for a first-order Taylor expansion:

$$\tilde{L}(r, \sigma) \approx \tilde{L}\left(r - \frac{d}{n}, \sigma\right) + \frac{d}{n}\frac{\partial \tilde{L}}{\partial r}\left(r - \frac{d}{n}, \sigma\right). \tag{4.3}$$

Substituting the RTE into the derivative yields:

$$\begin{aligned}\tilde{L}(r,\sigma) \approx \tilde{L}\left(r-\frac{d}{n},\sigma\right)\cdot \\ \left(1-\frac{d}{n}K\left(r-\frac{d}{n},\sigma\right)\right)+\frac{d}{n}K_s\left(r-\frac{d}{n},\sigma\right)\cdot \\ 4\int_0^{\pi}\int_0^{\pi/2}\tilde{L}\left(r-\frac{d}{n},\omega\right)\phi(\omega,\sigma)\sin\omega_2\,d\omega_2\,d\omega_1.\end{aligned} \tag{4.4}$$

Numerical Approximation of the Integral Term

The scattering integral is approximated by discrete summation. Combining this with the phase function and the discretized radiance values yields:

$$\frac{d}{n}K_s\left(\frac{(j-1)d}{n},\sigma\right)\cdot 4\sum_{l_1=0}^{n-1}\sum_{l_2=0}^{n-1}\frac{\pi}{n}\cdot\frac{\pi}{2n}\,\tilde{L}_{j-1}(l_1,l_2)\,A(l_1,l_2,k_1,k_2),$$

with

$$A(l_1,l_2,k_1,k_2)=\frac{1}{4\pi}\cdot\frac{(1-g^2)\sin\left(\frac{\pi l_2}{2n}\right)}{B(l_1,l_2,k_1,k_2)},$$

where the denominator $B(\cdot)$ encodes the anisotropic angular dependence via the Henyey–Greenstein formula:

$$\begin{aligned}B(l_1,l_2,k_1,k_2)=\Bigg(1-2g\Bigg[\sin\left(\frac{\pi l_2}{2n}\right)\sin\left(\frac{\pi k_2}{2n}\right)\cos\left(\frac{\pi(l_1-k_1)}{n}\right) \\ +\cos\left(\frac{\pi l_2}{2n}\right)\cos\left(\frac{\pi k_2}{2n}\right)\Bigg]+g^2\Bigg)^{3/2}.\end{aligned} \tag{4.5}$$

Collecting the constants and rearranging the terms, the final expression simplifies to:

$$\frac{\pi d}{2n^3}K_s\left(\frac{(j-1)d}{n},\sigma\right)(1-g^2)\sum_{l_1=0}^{n-1}\sum_{l_2=0}^{n-1}\tilde{L}_{j-1}(l_1,l_2)\tilde{A}(l_1,l_2,k_1,k_2),$$

with the simplified weight tensor:

$$\tilde{A}(l_1,l_2,k_1,k_2)=\frac{\sin\left(\frac{\pi l_2}{2n}\right)}{B(l_1,l_2,k_1,k_2)}.$$

This approximation reduces the integral over the sphere to a discrete summation over angular directions. The tensor $\tilde{A} \in \mathbb{R}^{n\times n\times n\times n}$ can be precomputed and reused across all iterations, which greatly enhances computational efficiency.

Iterative Update Rule

The recursive update for radiance becomes:

$$\begin{aligned}\tilde{L}_{j+1}(k_1,k_2) = \tilde{L}_j(k_1,k_2)\left(1-\frac{d(k_1,k_2)}{n}K\left(\frac{jd}{n},\sigma\right)\right)\\ +\frac{\pi d(k_1,k_2)}{2n^3}K_s\left(\frac{jd}{n},\sigma\right)(1-g^2)\cdot\\ \sum_{l_1,l_2}\tilde{L}_j(l_1,l_2)\tilde{A}(l_1,l_2,k_1,k_2).\end{aligned} \tag{4.6}$$

This update is iterated until $j = n-1$, and the final matrix $\tilde{L}_{n-1}$ represents the synthetic foggy image as observed from direction σ.

Extinction and Scattering Coefficients

The extinction and scattering coefficients are modeled as affine functions of the depth d and radiance L:

$$K(d,L) = A\cdot d + B\cdot L + C, \tag{4.7}$$

$$K_s(d,L) = X\cdot d + Y\cdot L + Z, \tag{4.8}$$

where $A, B, C, X, Y, Z \in \mathbb{R}^{n\times n}$ are coefficient matrices, and $\cdot$ denotes element-wise multiplication. This formulation enables spatially varying fog densities and supports depth- and brightness-dependent effects.

Numerical Properties and Convergence

The recursive approximation converges to the physical solution of the RTE as $n \to \infty$, assuming the scattering kernel and radiance field are smooth and bounded. In practice, setting $n = 100$ offers a good compromise between accuracy and efficiency.

The algorithm is implemented using GPU-accelerated tensor contractions and precomputed scattering matrices, allowing efficient parallel computation across all spatial directions and color channels.

Parameter Optimization via Gradient Descent

To ensure the physical consistency of the synthesized fog, the parameters of the extinction and scattering functions, $K(d, L)$ and $K_s(d, L)$, must be calibrated using real-world data. In our framework, both functions are modeled as linear combinations of the pixel-wise depth and radiance:

$$K(d, L) = A \cdot d + B \cdot L + C, \qquad K_s(d, L) = X \cdot d + Y \cdot L + Z, \tag{4.9}$$

where $A, B, C, X, Y, Z \in \mathbb{R}^{n \times n}$ are learnable matrices, and $\cdot$ denotes element-wise multiplication. Our goal is to find the optimal parameter set such that the simulated foggy image $\tilde{L}$ closely approximates its real-world counterpart L_f, given the same depth map.

To achieve this, we minimize the Pearson-based distance between $\tilde{L}$ and L_f:

$$\min_{A,B,C,X,Y,Z} D(\tilde{L}, L_f), \tag{4.10}$$

where the Pearson distance is defined as:

$$D(\tilde{L}, L_f) = 1 - \varrho(\tilde{L}, L_f). \tag{4.11}$$

The Pearson correlation coefficient $\varrho(X, Y)$ between two RGB images $X, Y \in \mathbb{R}^{n \times n \times 3}$ is given by:

$$\varrho(X, Y) = \frac{\sum_{i,j,c} (X_{i,j,c} - \overline{X})(Y_{i,j,c} - \overline{Y})}{\sqrt{\sum_{i,j,c} (X_{i,j,c} - \overline{X})^2} \cdot \sqrt{\sum_{i,j,c} (Y_{i,j,c} - \overline{Y})^2}}, \tag{4.12}$$

with the mean pixel intensity values defined as:

$$\overline{X} = \frac{1}{3n^2} \sum_{i,j,c} X_{i,j,c}, \qquad \overline{Y} = \frac{1}{3n^2} \sum_{i,j,c} Y_{i,j,c}. \tag{4.13}$$

Pearson distance is invariant to linear brightness shifts, making it more robust for fog simulation than MSE, which penalizes overall brightening due to the white fog effect.

Gradient computation Due to the recursive structure of our forward simulation, computing the gradients of $D(\tilde{L}_n, L_f)$ with respect

to each parameter $U \in \{A, B, C, X, Y, Z\}$ requires applying the chain rule across all iteration steps:

$$\frac{\partial D(\tilde{L}_n, L_f)}{\partial U} = \frac{\partial D}{\partial \tilde{L}_n} \cdot \frac{\partial \tilde{L}_n}{\partial U}. \tag{4.14}$$

The second term is computed recursively as:

$$\begin{aligned}\frac{\partial \tilde{L}_n}{\partial U} = &\frac{\partial \tilde{L}_{n-1}}{\partial U} \cdot \left(1 - \frac{K(\cdot)}{n} \cdot d\right) + \tilde{L}_{n-1} \cdot \frac{\partial}{\partial U}\left(1 - \frac{K(\cdot)}{n} \cdot d\right) \\ &+ \frac{\partial}{\partial U}\left(\frac{(1-g^2)\pi K_s(\cdot)}{2n^3} \cdot d\right) \cdot (\tilde{A} : \tilde{L}_{n-1}) \\ &+ \frac{(1-g^2)\pi K_s(\cdot)}{2n^3} \cdot d \cdot \tilde{A} : \left(\frac{\partial \tilde{L}_{n-1}}{\partial U}\right),\end{aligned} \tag{4.15}$$

where $\tilde{A}$ is the fixed scattering kernel, and d is the pixel-wise depth matrix.

This recursive process is initialized using the base-case gradients at the first iteration ($i = 1$):

$$\begin{aligned}&\frac{\partial \tilde{L}_1}{\partial A} = -L \cdot \frac{d^2}{n}, \quad \frac{\partial \tilde{L}_1}{\partial B} = -L^2 \cdot \frac{d}{n}, \quad \frac{\partial \tilde{L}_1}{\partial C} = -L \cdot \frac{d}{n}, \\ &\frac{\partial \tilde{L}_1}{\partial X} = \frac{(1-g^2)\pi d^2}{2n^3} \cdot \tilde{A} : L, \quad \frac{\partial \tilde{L}_1}{\partial Y} = \frac{(1-g^2)\pi d}{2n^3} \cdot L \cdot \tilde{A} : L, \\ &\frac{\partial \tilde{L}_1}{\partial Z} = \frac{(1-g^2)\pi d}{2n^3} \cdot \tilde{A} : L.\end{aligned} \tag{4.16}$$

We manually implemented these derivatives and used a custom version of the RMSProp optimizer to minimize the loss over 50 iterations. TensorFlow's automatic differentiation was used only for computing the $\partial D / \partial \tilde{L}_n$ term to reduce memory usage.

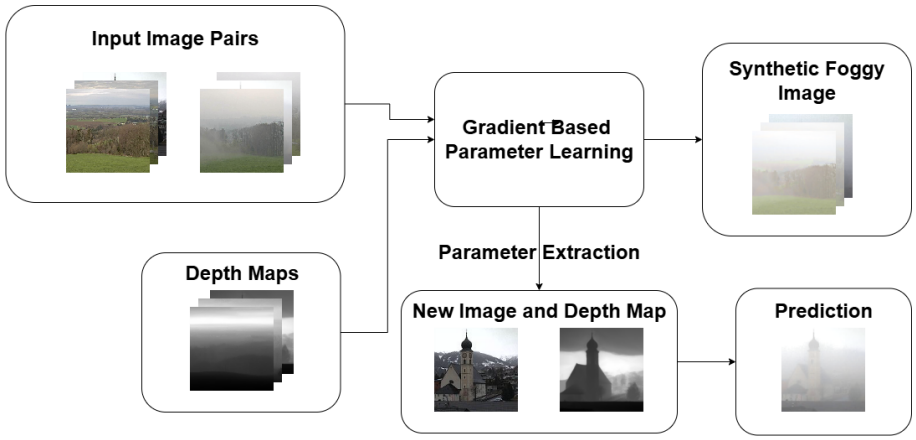

Figure 4.2: The proposed RTE-based fog synthesis workflow. The top row shows the training phase using paired fog–cloudy images and depth maps; the bottom row shows the inference phase on unseen clear-weather images.

This discretized RTE framework enables physically grounded simulation of fog with non-uniform, anisotropic scattering. Its recursive, tensor-based structure makes it highly compatible with modern depth-aware vision pipelines and synthetic data generation tasks.

To provide a high-level overview of the method, Figure 4.2 illustrates the complete training and inference pipeline of the proposed RTE-based fog synthesis. The top part shows the optimization of physical parameters from fog–cloudy image pairs and depth maps, while the bottom part demonstrates the application of the learned model to unseen scenes using only depth information.

4.3.2 Dataset Construction

To support the evaluation and calibration of our fog simulation model, we constructed a curated dataset of real-world outdoor scenes with diverse atmospheric conditions. The dataset was designed to provide foggy–cloudy image pairs and fog density annotations suitable for both physical parameter optimization and quantitative benchmarking.

Image Collection and Curation

A total of 25,527 images were collected over the course of one month using 608 publicly accessible web cameras located across 18 countries, primarily in Europe, with additional sources from Canada and Russia. The selected cameras covered various urban, suburban, and rural environments to ensure diversity in landscape and atmospheric visibility.

The following filtering steps were applied:

- **Illumination filtering**: Nighttime images and frames with excessive backlighting were discarded to ensure consistent natural lighting.
- **Quality filtering**: Images with compression artifacts, motion blur, or severe occlusion were excluded.
- **Geometric filtering**: To mitigate lens distortion (e.g., from fisheye optics), only the central region of each frame was retained for analysis.

After these steps, a clean set of 2041 images was retained, manually annotated into three weather categories: *sunny* (1192), *cloudy* (526), and *foggy* (323).

Fog Density Annotation

To facilitate domain adaptation of the CycleGAN[73] baseline—which was originally trained on automotive image datasets with predominantly urban street scenes—a subjective fog density score in the $[0, 1]$ range was introduced. The goal was to fine-tune the model for natural, landscape-oriented webcam images, where the pretrained model exhibited significant performance degradation. The fog intensity score served as a conditioning signal during fine-tuning, enabling the generator to synthesize more appropriate fog effects adapted to different levels of atmospheric thickness.

Paired Fog–Cloudy Image Extraction

A subset of 123 **paired foggy–cloudy images** was identified from the dataset, based on the following criteria:

- **Spatial consistency**: Same camera and scene geometry.

- **Temporal proximity**: Within a short time window to reduce seasonal and lighting variation.
- **Weather variability**: Presence of visibly different fog conditions while retaining overall scene identity.

To facilitate high-confidence pair selection, an AlexNet-based CNN weather classifier was developed to differentiate between sunny, cloudy, and foggy conditions. The model was trained using a manually curated dataset consisting of 2,041 images, which includes 1,192 sunny, 526 cloudy, and 323 foggy examples. After training, the classifier achieved an accuracy of 97.1% on the training set and 85.2% on the test set, resulting in a weighted test accuracy of 87.7%. This classifier was then employed to automatically identify consistent fog-cloudy pairs from the remaining pool of images, ensuring spatial and temporal coherence across the selected scenes. To mitigate the class imbalance present in the training dataset (1,192 sunny, 526 cloudy, and 323 foggy examples), class-size weighting was applied in the loss function, ensuring that no class was overrepresented in the optimization process. Furthermore, an independent validation set was used together with an early stopping criterion, which prevented overfitting to the majority class and helped maintain balanced performance across categories. These measures minimized the negative impact of the dataset imbalance on the final model.

Use in Model Calibration and Evaluation

The collected data served two primary purposes:

- **Parameter optimization**: The paired fog–cloudy images were used to optimize the coefficients A, B, C, X, Y, Z in the fog model via gradient-based minimization of a Pearson-based image similarity loss.
- **Quantitative benchmarking**: The same pairs allowed consistent evaluation using LPISP[89], FID[87], and Pearson correlation metrics, as discussed in Section 4.3.5.

Thanks to this dataset, we were able to validate our RTE-based synthesis method not only qualitatively but also in terms of structural

realism and physical consistency. The dataset's geographic and atmospheric diversity enhances the generalizability of the experimental findings.

4.3.3 Depth Map Estimation using Marigold

Our fog simulation model relies on accurate per-pixel depth estimates to determine the amount of light attenuation and in-scattering along each viewing ray. Since most real-world images lack ground truth depth, we use Marigold [90], a recent monocular depth estimation model based on latent diffusion, to infer dense distance maps from single RGB images.

Overview of the Marigold Model

Marigold is a monocular depth estimation method derived from Stable Diffusion. It leverages the rich visual priors learned during large-scale image generation training, and fine-tunes only the denoising U-Net on synthetic RGB-D datasets (Hypersim, Virtual KITTI). Both the input image and depth map are encoded into a shared latent space using a pretrained variational autoencoder (VAE), and the denoising process is carried out in this latent space.

Marigold differs from conventional CNN or Transformer-based architectures by functioning entirely in the latent domain. It generates depth predictions that are affine-invariant and can generalize effectively to real-world scenes, all without needing camera intrinsics or scale information. During inference, the model gradually denoises a randomly initialized latent code, conditioned on the input image, to reconstruct the estimated depth map.

Application to Our Dataset

We used the public Google Colab implementation[1] provided by the authors to apply Marigold to our curated fog–cloudy dataset (Section 4.3.2). All parameters were set to their highest-quality settings:

[1] `https://marigoldmonodepth.github.io`

- **Ensemble size:** 10
- **Number of denoising steps:** 20
- **Processing resolution:** 768
- **Match input resolution:** True

We did not apply any additional normalization or filtering to the output. The predicted depth maps were used directly as distance fields $d \in \mathbb{R}^{H \times W}$ for computing fog optical thickness along each ray. The ensemble strategy effectively reduced the generative variance inherent in diffusion-based inference and provided stable, structurally coherent depth estimations.

Limitations of Monocular Depth Estimation

While Marigold delivers consistent and robust depth predictions even for in-the-wild imagery, it inherits some common limitations of monocular depth estimation:

- Loss of detail in thin or reflective structures,
- Inaccuracies in large untextured or homogeneous regions,
- Reduced reliability under extreme weather (e.g., dense fog),

In particular, fog can obscure important scene features and degrade the accuracy of predicted depth. To mitigate this, in our paired fog–cloudy image setup, we always generated the depth map from the *cloudy* (i.e., fog-free) image. This ensured that the resulting distance field was as accurate as possible and not biased by weather-induced visibility loss.

Alternative Approaches

Our pipeline is compatible with any depth estimation method, including:

- stereo-based depth (if available),
- structure-from-motion (SfM) [91],

- LiDAR ground truth for evaluation,
- learned depth via other monocular models (e.g., MiDaS [92], DPT [93]).

Nevertheless, Marigold was chosen due to its strong generalization ability, single-image input requirement, and metric scale outputs.

4.3.4 Algorithmic Implementation of the Discretized RTE

In this section, we provide the full algorithmic formulation of our discretized radiative transfer simulation framework. The method was developed in three stages, each enhancing computational efficiency and flexibility, culminating in a gradient-compatible version suitable for parameter learning.

Original Algorithm: Direct Iterative Scheme

The first version of our algorithm corresponds to a direct implementation of the discretized radiative transfer equation as defined in (4.6). It performs explicit nested loops over all angular directions and color channels, iteratively constructing the radiance tensor layer by layer. Although computationally expensive due to the $\mathcal{O}(n^5)$ complexity, this naive version is useful for didactic purposes and was used as a baseline implementation.

The pseudocode is shown in Algorithm 1, where $\tilde{L}$ denotes the discretized radiance field, K and K_s are the extinction and scattering coefficients, and $\tilde{A}$ is the anisotropic scattering kernel derived from the Henyey–Greenstein phase function.

Optimized Tensor-based Algorithm

To enhance computational efficiency, we reformulated the radiative transfer update rule via full tensorization. Instead of summing over angular directions for each pixel and each depth step, we express the update as a tensor contraction, enabling a significant reduction in per-iteration complexity.

Algorithm 1 Original Algorithm

Require: L, d, n, g, K, K_s ▷ Input image L, depth map d, steps n, asymmetry g, kernels K, K_s

Ensure: $\tilde{L}$ ▷ Final radiance after n iterations

$\tilde{L} \leftarrow L$ ▷ Start from fog-free image

for $j = 0$ to $n - 1$ **do** ▷ j = depth iteration; dependence via $\tilde{L}_j$

$L \leftarrow 0^{n \times n \times 3}$ ▷ Buffer for next iterate

for $c = 0$ to 2 **do** ▷ RGB channels

for $k_1 = 0$ to $n - 1$ **do** ▷ Pixel index (horizontal)

for $k_2 = 0$ to $n - 1$ **do** ▷ Pixel index (vertical)

$S \leftarrow 0$ ▷ Accumulator for in-scattering at (k_1, k_2)

for $l_1 = 0$ to $n - 1$ **do** ▷ Angular index #1

for $l_2 = 0$ to $n - 1$ **do** ▷ Angular index #2

$$S \leftarrow S + \tilde{A}(k_1, k_2, l_1, l_2)\, \tilde{L}[l_1, l_2, c]$$

$$L[k_1, k_2, c] \leftarrow \tilde{L}[k_1, k_2, c] \left(1 - \frac{d[k_1, k_2]}{n} K(d, \tilde{L}[:, :, c])[k_1, k_2]\right) + \frac{(1 - g^2)\pi d[k_1, k_2]}{2n^3} K_s(d, \tilde{L}[:, :, c])[k_1, k_2] S$$

▷ Recursive update

$\tilde{L} \leftarrow L$ ▷ Advance to next depth step

Let us denote the radiance tensor at iteration j and color channel c as $\tilde{L}_j^{(c)} \in \mathbb{R}^{n\times n}$. To facilitate vectorized computation, we define base matrices $\tilde{L}_1, \tilde{L}_2 \in \mathbb{N}^{n\times n}$ as:

$$\tilde{L}_2 := \begin{bmatrix} 1 & 2 & \cdots & n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2 & \cdots & n \end{bmatrix}, \quad \tilde{L}_1 := \tilde{L}_2^T. \tag{4.17}$$

From these, we construct 4D tensors $L_1, L_2 \in \mathbb{N}^{n\times n\times n\times n}$ by tiling $\tilde{L}_1$ and $\tilde{L}_2$ along the third and fourth axes. Similarly, we define $K_1 := L_2^T$ and $K_2 := L_1^T$ for index inversion.

The Henyey-Greenstein scattering kernel is precomputed for all directional combinations and stored in a 4D tensor $\tilde{A} = \tilde{A}(L_1, L_2, K_1, K_2)$ and $\tilde{A} \in \mathbb{R}^{n\times n\times n\times n}$. This step fixes the anisotropy parameter g for the duration of the simulation, which is a necessary trade-off to achieve runtime efficiency.

The optimized per-channel update rule is then formulated as:

$$\tilde{L}_{j+1}^{(c)} = \tilde{L}_j^{(c)} \cdot \left(1 - \frac{K(d, \tilde{L}_j^{(c)})}{n} \cdot d\right) + \frac{(1-g^2)\pi}{2n^3} \cdot K_s(d, \tilde{L}_j^{(c)}) \cdot d \cdot \tilde{A} : \tilde{L}_j^{(c)}, \tag{4.18}$$

where $\cdot$ denotes element-wise multiplication and $:$ is a double tensor contraction over the third and fourth axes:

$$\tilde{A} : \tilde{L} = \sum_{l_1,l_2} \tilde{A}_{l_1,l_2,:,:} \, \tilde{L}_{l_1,l_2}. \tag{4.19}$$

This reformulation significantly reduces the computational complexity of each update step from $\mathcal{O}(n^5)$ to $\mathcal{O}(n)$ per channel by leveraging tensor contraction and precomputed angular weights. While the theoretical gain is substantial, the most impactful practical benefit is observed in the runtime and memory usage. In the original algorithm, even relatively small resolutions such as 32×32 pixels proved impractically slow, especially when processing color images over multiple recursive steps. In contrast, the optimized version enables image synthesis at resolutions up to 200×200, making it suitable for mid-scale image augmentation tasks. Importantly, this acceleration was achieved by reformulating the inner four nested loops of the original algorithm into a single tensor operation, thereby exploiting matrix and

tensor algebra for efficient computation. Furthermore, by utilizing the CuPy Python library, the algorithm was ported to the GPU, where the massively parallel architecture could be fully leveraged for large-scale tensor contractions. While these changes drastically improved runtime performance, they simultaneously increased the memory requirements of the method, since the precomputed scattering kernel and intermediate tensors must be stored explicitly in GPU memory.

For learning-based parameter calibration, we adopted a compromise resolution of 100×100 pixels. This choice balances computational feasibility and spatial expressiveness, allowing the model to capture meaningful fog structure while keeping GPU memory requirements and training time manageable.

While the computational gain is substantial, it comes at the cost of significantly increased memory consumption, due to the size of the tensor $\tilde{A}$, which scales as $\mathcal{O}(n^4)$. Moreover, since the scattering kernel is fixed for a given g, dynamic adjustments to the asymmetry parameter require full recomputation of $\tilde{A}$.

Nonetheless, precomputing the scattering kernel once enables the reuse of $\tilde{A}$ across all iterations and color channels. In practice, we found that fixing $g = 0.85$ provides a realistic forward-scattering effect that aligns well with real-world fog distributions.

The optimized inference algorithm is summarized in Algorithm 2.

Gradient-based Version for Training

To enable supervised learning of the fog parameters (A, B, C, X, Y, Z), we extend the previous version to include gradient propagation through the iterative simulation. This allows us to compute the partial derivatives of the final radiance tensor with respect to the learnable coefficients using recursive differentiation.

Each derivative is propagated through the radiative update using the chain rule. The recursion involves both the direct dependence on parameters (e.g., $\frac{\partial \tilde{L}}{\partial A}$) and the indirect dependence via $\tilde{L}_j$. The base step initializes the gradients with respect to each parameter analytically, and the full update procedure follows through n steps.

This version is described in Algorithm 3, which was used during model calibration via RMSProp optimization.

Algorithm 2 Optimized Algorithm

Require: $L, d, n, g, K, K_s, L_1, L_2, K_1, K_2$ ▷ Input image L, depth map d, steps n, asymmetry g, kernels K, K_s, tensors for computing $\tilde{A}$: L_1, L_2, K_1, K_2

Ensure: $\tilde{L}$ ▷ Final radiance after n iterations

$S \leftarrow \tilde{A}(L_1, L_2, K_1, K_2)$ ▷ Calculating in-scattering

$\tilde{L} \leftarrow L$ ▷ Start from fog-free image

for $j = 0$ to $n - 1$ **do** ▷ j = depth iteration; dependence via $\tilde{L}_j$

$L \leftarrow 0^{n \times n \times 3}$ ▷ Buffer for next iterate

for $c = 0$ to 2 **do** ▷ RGB channels

$$L[:,:,c] \leftarrow \tilde{L}[:,:,c] \cdot \left(1 - \frac{K(d, L[:,:,c])}{n} \cdot d\right) + \frac{(1-g^2)\pi K_s(d, L[:,:,c])}{2n^3} \cdot d \cdot S : \tilde{L}[:,:,c]$$

▷ Recursive update

$\tilde{L} \leftarrow L$ ▷ Advance to next depth step

Algorithm 3 Optimized Algorithm with Gradient (Part 1)

Require: $L, d, n, g, K, K_s, L_1, L_2, K_1, K_2$

Ensure: $\tilde{L}, \frac{\partial \tilde{L}}{\partial A}, \frac{\partial \tilde{L}}{\partial B}, \frac{\partial \tilde{L}}{\partial C}, \frac{\partial \tilde{L}}{\partial X}, \frac{\partial \tilde{L}}{\partial Y}, \frac{\partial \tilde{L}}{\partial Z}$

Initialize:

$$S \leftarrow \tilde{A}(L_1, L_2, K_1, K_2), \quad \tilde{L} \leftarrow L,$$

$$\tilde{L}_{\partial A_0} \leftarrow -L \cdot \frac{d^2}{n}, \quad \tilde{L}_{\partial B_0} \leftarrow -L^2 \cdot \frac{d}{n}, \quad \tilde{L}_{\partial C_0} \leftarrow -L \cdot \frac{d}{n},$$

$$\tilde{L}_{\partial X_0} \leftarrow \frac{(1-g^2)\pi d^2}{2n^3} \cdot S : L,$$

$$\tilde{L}_{\partial Y_0} \leftarrow \frac{(1-g^2)\pi d}{2n^3} \cdot L \cdot S : L,$$

$$\tilde{L}_{\partial Z_0} \leftarrow \frac{(1-g^2)\pi d}{2n^3} \cdot S : L.$$

Optimized Algorithm with Gradient (Part 2)

for $j = 0$ to $n - 1$ **do**

Initialize $L \leftarrow 0^{n \times n \times 3}$

for $c = 0$ to 2 **do**

Compute the updated $\tilde{L}$:

$$L[:,:,c] \leftarrow \tilde{L}[:,:,c] \cdot \left(1 - \frac{K(d,L)}{n} \cdot d\right) + \frac{(1-g^2)\pi K_s(d,L)}{2n^3} \cdot d \cdot S : \tilde{L}[:,:,c].$$

Compute gradients:

$$\begin{aligned}\tilde{L}_{\partial A}[:,:,c] \leftarrow \tilde{L}_{\partial A_0}[:,:,c] \cdot &\left(1 - \frac{K(d,L)}{n} \cdot d\right) \\ &- \frac{d}{n} \cdot L[:,:,c] \cdot (L[:,:,c] + A \cdot \tilde{L}_{\partial A_0}[:,:,c]) \\ &+ \frac{(1-g^2)\pi X \cdot \tilde{L}_{\partial A_0}[:,:,c]}{2n^3} \cdot d \cdot S : L[:,:,c] \\ &+ \frac{(1-g^2)\pi K_s(d,L)}{2n^3} \cdot d \cdot S : \tilde{L}_{\partial A_0}[:,:,c],\end{aligned}$$

$$\begin{aligned}\tilde{L}_{\partial X}[:,:,c] \leftarrow \tilde{L}_{\partial X_0}[:,:,c] \cdot &\left(1 - \frac{K(d,L)}{n} \cdot d\right) \\ &- \frac{d}{n} \cdot L[:,:,c] \cdot (A \cdot \tilde{L}_{\partial X_0}[:,:,c]) \\ &+ \frac{(1-g^2)\pi(L[:,:,c] + X \cdot \tilde{L}_{\partial X_0}[:,:,c])}{2n^3} \cdot d \cdot S : L[:,:,c] \\ &+ \frac{(1-g^2)\pi K_s(d,L)}{2n^3} \cdot d \cdot S : \tilde{L}_{\partial X_0}[:,:,c].\end{aligned}$$

Apply similar updates for $\tilde{L}_{\partial B}$, $\tilde{L}_{\partial C}$, $\tilde{L}_{\partial Y}$, and $\tilde{L}_{\partial Z}$.

Update:

$$\begin{aligned}&\tilde{L} \leftarrow L, \quad \tilde{L}_{\partial A_0} \leftarrow \tilde{L}_{\partial A}, \quad \tilde{L}_{\partial B_0} \leftarrow \tilde{L}_{\partial B}, \quad \tilde{L}_{\partial C_0} \leftarrow \tilde{L}_{\partial C}, \\ &\qquad \tilde{L}_{\partial X_0} \leftarrow \tilde{L}_{\partial X}, \quad \tilde{L}_{\partial Y_0} \leftarrow \tilde{L}_{\partial Y}, \quad \tilde{L}_{\partial Z_0} \leftarrow \tilde{L}_{\partial Z}.\end{aligned}$$

Comparison and Practical Notes

All three implementations were evaluated for numerical correctness and speed. The original algorithm served as ground truth but proved computationally impractical beyond $n = 32$. The optimized version was used for inference and visual experiments, while the gradient-based implementation supported parameter fitting using paired fog–cloudy image data.

Table 4.1 summarizes the computational trade-offs between the three versions.

Table 4.1: Comparison of the three RTE simulation algorithms.

Version	**Purpose**	**Complexity**	**Training support**
Original	Baseline, verification	$\mathcal{O}(n^5)$	No
Optimized	Efficient forward model	$\mathcal{O}(n)$	No
With Gradient	Parameter optimization	$\mathcal{O}(n^3)$	Yes

We note that all variants converge to the same radiative field as $n \to \infty$, assuming the extinction and scattering fields are smooth. The gradient-compatible formulation introduces minor overhead but is necessary for learning interpretable physical parameters from image data.

4.3.5 Comparison with Analytical and GAN-based Models

To evaluate the realism and physical fidelity of our proposed fog simulation method, we compare it against two commonly used fog synthesis baselines:

1. **Koschmieder's model**: a classical analytical formulation based on homogeneous extinction and constant airlight.

2. **GAN-based synthesis**: a deep-generative approach trained to generate fog in clear-weather images using unpaired image translation.

Koschmieder's Law

Koschmieder's model assumes a homogeneous medium and isotropic scattering. The observed radiance $L(d)$ at a scene point located at distance d is given by:

$$L(d) = L_0 \cdot e^{-Kd} + L_{\text{air}} \cdot \left(1 - e^{-Kd}\right), \tag{4.20}$$

where L_0 is the object radiance without fog, K is the constant extinction coefficient, and L_{air} is a background airlight color. This model has been widely used in synthetic fog augmentation for vision dataset, but suffers from several limitations:

- It assumes uniform fog density across the scene.
- It neglects angular dependencies of light scattering.
- It produces overly smooth transitions without accounting for spatial structure.

Our method generalizes this formulation by numerically approximating the full RTE along each viewing ray. This includes depth-dependent extinction $K(d)$, direction-dependent scattering, and recursive integration of in-scattered light. Koschmieder's equation emerges as a special case of our model under the assumptions of constant coefficients and isotropic scattering.

GAN-based Fog Simulation

We also compare our method to a CycleGAN-based model trained to perform image-to-image translation from clear to foggy images. This approach uses adversarial and perceptual losses to learn the transformation without relying on physical modeling or depth information.

The baseline CycleGAN model was originally trained on urban and driving-oriented datasets, including Cityscapes [94], SFSU Foggy Driving [95], and the RESIDE dataset [96]. While this enables strong performance on street scenes, we observed poor generalization to our webcam-based landscape images, which contain diverse geometry, vegetation, and lighting conditions.

To address this, we manually annotated a subset of our dataset with subjective fog density scores in the $[0, 1]$ interval, representing

perceived atmospheric thickness. These annotations were then used to fine-tune the CycleGAN model, effectively introducing a conditioning mechanism on fog strength and enabling adaptation to natural outdoor scenes.

While GAN-based methods can generate visually plausible results, they exhibit several drawbacks in the context of physically motivated synthesis:

- **Lack of depth awareness:** fog intensity is not explicitly linked to scene geometry.
- **Loss of structure:** thin objects and edges may be distorted or oversmoothed.
- **No physical consistency:** these models cannot simulate visibility range, scattering profiles, or light halos in a controllable manner.

Figure 4.3 illustrates side-by-side results generated by all three methods using the same clear-weather input and associated depth map. Our method maintains spatial structure and depth-dependent gradients.

Quantitative Evaluation Metrics

To assess the realism and fidelity of the generated foggy images, we employ the following quantitative metrics:

- **LPIPS (Learned Perceptual Image Patch Similarity)** [89]: Measures perceptual similarity based on deep feature distances extracted from pretrained networks. Lower values indicate higher perceptual similarity to reference foggy images.
- **FID (Fréchet Inception Distance)** [87]: Computes the distance between feature distributions of generated and real images using a pretrained Inception network. Widely used for generative model evaluation; lower scores indicate higher realism and distributional alignment.
- **Pearson Correlation**: Measures the correlation between the mean feature vectors used in the FID metric. It helps visualize

(a) Cloudy image (input)

(b) Real fog (ground truth)

(c) Proposed RTE-based model

(d) Koschmieder's model

(e) CycleGAN output

Figure 4.3: A visual comparison of synthetic fog generated by different models. The proposed RTE-based method (c) produces fog effects that are visually more consistent with the real-worldreference (b), capturing both depth-aware attenuation and directional scattering: (a) cloudy image (input); (b) real fog (ground truth); (c) RTE-based model; (d) Koschmieder's model; (e) CycleGAN output.

image-level similarity and complements FID when evaluating fog synthesis quality, especially in cases of limited data.

Instead of a single summary, we report the results in two separate evaluations due to the nature of the metrics:

- **FID** and **Pearson Correlation** are computed between sets of images and thus allow comparison across scene groups (e.g., foggy vs. generated).

- **LPIPS** requires image-to-image correspondence and was therefore computed only on aligned foggy–cloudy (123) pairs.

FID and Pearson Correlation Evaluation

Table 4.2 presents normalized FID scores between real and synthesized fog variants. Our RTE-based model consistently achieves the lowest FID values relative to the real fog group across all comparison sets, indicating superior distributional alignment in feature space. In particular, the synthetic fog generated using our method more closely resembles the true fog distribution than both the classical Koschmieder-based rendering and the learned CycleGAN transformation.

Interestingly, the Koschmieder model exhibits the highest similarity to the cloudy image group, indicating that it deviates the least from the original, fog-free input images. While this results in moderate performance in cloudy-related comparisons, it also suggests that the model performs minimal domain transformation, which limits its realism and effectiveness when compared directly to real foggy images. The CycleGAN-based results yield consistently higher FID scores across all pairings, suggesting that the generated fog diverges more significantly from the statistical distribution of real foggy images. This is likely due to the lack of physical constraints in the adversarial training process, resulting in stylized but less physically realistic outputs. Since FID is sensitive not only to mean feature differences but also to covariance mismatches, the elevated values indicate both visual and structural discrepancies.

Table 4.3 complements these findings with Pearson correlation values computed between mean deep feature vectors of image groups. Here too, the RTE-based model exhibits the strongest correlation with both the real fog and the fog–cloudy pairs. These results reinforce the conclusion that our method not only matches global feature distributions (FID), but also preserves fine-grained feature trends in deep semantic space.

Figure 4.4 visualizes this distribution via a 2D PCA projection of mean feature vectors. While the CycleGAN output (cyan) may appear most visually compelling at first glance, it primarily aligns with the distribution of generic foggy images rather than with the real fog in the paired dataset. In contrast, the RTE-based output (olive) points more directly toward the fog-pair group, reflecting stronger seman-

Table 4.2: Normalized Fréchet Inception Distance (FID) between image groups. Lower is better.

	Fog	**Fog pair**	**Ours (RTE)**	**Koschmieder**	**CycleGAN**
Fog	-	0.399	**0.512**	0.570	0.785
Fog pair	0.399	-	**0.327**	0.448	0.867
Cloudy	0.535	0.580	0.638	**0.577**	0.979
Sunny	0.553	0.618	0.654	**0.580**	1.000

Table 4.3: Pearson correlation between mean feature vectors of image groups. Higher is better.

	Fog	**Fog pair**	**Ours (RTE)**	**Koschmieder**	**CycleGAN**
Fog	-	0.964	**0.926**	0.887	0.765
Fog pair	0.964	-	**0.953**	0.929	0.759
Cloudy	0.864	0.893	0.858	**0.909**	0.589
Sunny	0.844	0.872	0.848	**0.909**	0.567

tic alignment with physically grounded transformations. Notably, the Koschmieder model's vector (red brown) remains close to the original cloudy direction (purple), indicating that it performs minimal transformation and captures only limited fog characteristics. This visualization reinforces the interpretation that our RTE-based method not only preserves structure but also generates fog effects that are statistically and semantically consistent with real fog events.

LPIPS Evaluation

Table 4.4 displays LPIPS scores calculated for foggy–cloudy image pairs using three backbone networks. Our RTE-based method achieves the lowest perceptual distances across all configurations, indicating strong structural consistency and high perceptual realism.

While LPIPS is not a distributional metric like FID, it serves as a complementary, image-level measure of perceptual fidelity. The consistently lower LPIPS values for our method suggest that the RTE-based synthesis preserves key content and spatial structure from the source cloudy images, while applying fog in a visually plausible yet physically grounded way.

Interestingly, the ranking of the baselines varies across backbones: CycleGAN achieves the second-best LPIPS scores under AlexNet and SqueezeNet, likely due to its learned stylization capabilities. In contrast, the Koschmieder model ranks second with VGG, which is more

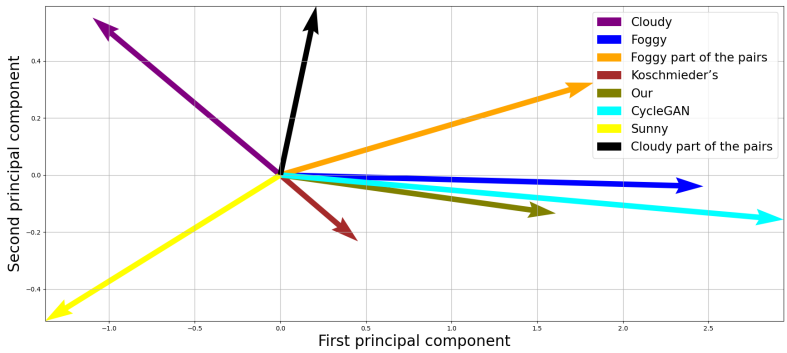

Figure 4.4: 2D visualization of the mean feature vectors.

sensitive to high-frequency texture distortions—suggesting that the analytical model preserves spatial coherence better than CycleGAN under that criterion.

Together, these results demonstrate that our physically grounded fog synthesis model strikes an effective balance: it aligns well with real fog characteristics while maintaining high perceptual similarity and minimal content distortion on a per-image basis.

Table 4.4: Average LPIPS scores for aligned foggy–cloudy image pairs. Lower is better.

Method	**AlexNet**	**SqueezeNet**	**VGG**
Koschmieder	0.3151	0.2560	0.3753
CycleGAN	0.2659	0.2197	0.4041
Ours (RTE)	**0.2172**	**0.1639**	**0.2215**

Sensitivity to Depth Estimation Noise

To evaluate the robustness of our RTE-based fog synthesis to inaccuracies in depth estimation, we conducted a noise injection experiment. A subset of 40 fog–cloudy image pairs was selected, and synthetic Gaussian noise was added to the depth maps before fog rendering.

The noise was sampled from a zero-mean normal distribution with standard deviations $\sigma = 0.05$ and $\sigma = 0.10$, corresponding to 5% and

10% relative perturbations with respect to normalized depth values. For each noise level, we regenerated the synthetic fog images and re-evaluated their quality using the same metrics as before: FID and Pearson correlation.

Table 4.5 summarizes the results.

Table 4.5: Impact of Gaussian noise in depth estimation on fog synthesis quality. All metrics averaged over 40 image pairs.

Noise Std. (σ)	**FID**	**Pearson Corr.**	ΔCorrelation (%)
0 (baseline)	1.000	0.937	–
0.05	1.093	0.928	-1.0%
0.10	1.162	0.910	-2.9%

As shown, increasing the noise level leads to a gradual degradation in both perceptual quality and physical consistency. Nevertheless, even with 10% perturbation, the Pearson correlation remains above 0.91, and the FID increase is moderate. These results suggest that our method is robust to moderate depth inaccuracies, such as those commonly encountered in monocular depth estimation.

4.4 Computational Efficiency and Resource Analysis

In addition to qualitative and quantitative evaluations, we assessed the computational demands of our RTE-based fog synthesis framework. Understanding the runtime and memory usage is essential for scaling the method to large datasets or deploying it in practical scenarios.

4.4.1 Inference Cost Analysis

Table 4.6 shows the average inference time and GPU memory consumption for different image resolutions, measured during fog synthesis on an NVIDIA RTX 3090 GPU. As expected, both time and VRAM usage increase with resolution, due to the use of angularly dependent radiative transfer computations across spatial grids.

While computationally more intensive than closed-form methods, the RTE-based approach enables a finer level of physical realism and

Table 4.6: Average inference time and VRAM usage for RTE-based fog generation.

Resolution	Time (s)	VRAM (MB)
25×25	0.115	330
50×50	0.266	444
75×75	0.870	960
100×100	2.515	2366
125×125	6.150	5392
150×150	13.302	10960
175×175	25.550	18356

parameter control. In practice, intermediate resolutions (e.g., 100–125 pixels) offer a good trade-off between fidelity and computational cost.

4.4.2 Training Cost and Scalability

Table 4.7 presents the average time and VRAM usage per single training iteration using the RMSProp optimizer and TensorFlow. Similar to inference, resource requirements grow with spatial resolution due to the need to store high-dimensional intermediate tensors.

Table 4.7: Average training time and VRAM usage per iteration (RMSProp).

Resolution	Time (s)	VRAM (MB)
25×25	0.661	1133
50×50	2.001	1247
75×75	9.469	1754
100×100	30.073	3197
125×125	77.633	6323
150×150	169.786	11874
175×175	330.760	21173

In our experiments, we trained the model for 50 iterations per image, which empirically provided a good balance between convergence and numerical stability. Higher iteration counts occasionally led to exploding gradients or instability, especially at higher resolutions. Thus, the reported values represent the per-iteration cost, while the full train-

ing process consisted of 50 such steps per image. Importantly, after integrating gradient computation and TensorFlow into the algorithm, the complexity increased to approximately $\mathcal{O}(n^3)$ in training mode, due to the need to propagate and store derivatives across the high-dimensional tensor contractions.

4.4.3 Comparison with Baselines

For comparison, the classical Koschmieder model requires negligible time and memory, due to its closed-form structure. The CycleGAN-based method is also highly efficient at inference: generating a 256×256 image takes only 0.037 seconds and 1.5 GB VRAM. However, this comes at the cost of expensive training (multi-day GPU use), instability, and limited generalizability.

4.5 Claims

Claim 3 *The discretized Radiative Transfer Equation (RTE), applied in image space using monocular depth maps, enables physically consistent fog synthesis. The integration of the Henyey–Greenstein phase function is essential to simulate forward-scattering behavior accurately and to achieve perceptually realistic fog effects.*

Reasoning. The proposed model formulates fog simulation as a recursive numerical solution to the RTE in a discretized spatial–angular domain. Using a per-pixel depth map inferred from a single RGB image, the method computes light attenuation and in-scattering in accordance with physical radiative principles. Crucially, the use of the Henyey–Greenstein phase function with forward asymmetry parameter $g \approx 0.85$ models the directional nature of light scattering in real fog. This leads to depth-dependent airlight accumulation and angular glow effects that are not captured by traditional isotropic models (e.g., Koschmieder). Quantitative evaluation using LPIPS, FID, and Pearson correlation confirms that this combination produces more realistic and structure-preserving synthetic fog than isotropic or purely heuristic baselines. □

Claim 4 *The proposed recursive numerical form of the RTE reduces to*

the Koschmieder model under homogeneous and isotropic assumptions, thus serving as its physical generalization.

Reasoning. By setting $K = \text{const.}$ and $\phi(\omega, \sigma) = \text{const.}$ in the discretized RTE formulation, the recursive integral simplifies to an exponential decay model with constant airlight, matching the analytical form of Koschmieder's equation:

$$L(d) = L_0 e^{-Kd} + L_{\text{air}}(1 - e^{-Kd}).$$

This shows that the proposed model encompasses Koschmieder's as a special case, and extends it to inhomogeneous, anisotropic media. □

Claim 5 *Images generated using the proposed RTE-based model exhibit higher perceptual and structural realism compared to traditional analytical or GAN-based synthesis methods, as measured by LPIPS, FID, and Pearson correlation. At the same time, the RTE's recursive discretization is efficiently implementable in tensor-based deep learning frameworks and is fully differentiable, allowing integration into learning pipelines for gradient-based optimization.*

Reasoning. In our experiments on a curated fog-cloudy dataset, the RTE-based method consistently outperformed Koschmieder's analytical model and CycleGAN regarding perceptual similarity (LPIPS) and distributional realism (FID). Additionally, the synthetic fog produced by this method showed a stronger correlation with depth maps and actual fog data in feature space, achieving higher Pearson correlation scores. These findings confirm that the model yields more realistic and physically consistent images, aligning with both human visual perception and physical properties of fog.

From an implementation perspective, the algorithm was implemented in TensorFlow using recursive tensor contractions and precomputed scattering matrices. The radiative update is differentiable with respect to extinction and scattering parameters, enabling gradient-based learning via RMSProp. Despite its high memory footprint, the method scales to moderate resolutions (e.g., 100x100) and is suitable for synthetic data augmentation and differentiable simulation tasks. □

4.6 Conclusions

In this chapter, we proposed a physically grounded method for synthetic fog generation based on a discretized numerical solution of the Radiative Transfer Equation. Unlike traditional analytical models or generative adversarial networks, our method integrates depth information and models both extinction and in-scattering effects along each viewing direction using a recursive tensor-based approach.

Compared to classical models such as Koschmieder's law, our model offers the following advantages:

- simulation of spatially varying fog densities based on input depth maps,
- using the Henyey–Greenstein phase function to incorporate the anisotropic scattering,
- recursive radiative accumulation over a discretized angular domain,
- GPU-accelerated implementation using per-channel tensor contractions.

We conducted wide quantitative and qualitative evaluations using a real-world foggy–cloudy image dataset. Our method achieved:

- the lowest Fréchet Inception Distance (FID), indicating strong distributional alignment with real fog images,
- the highest Pearson correlation with real fog scenes, reflecting semantic and structural realism,
- the lowest LPIPS scores across all backbones, confirming minimal perceptual distortion relative to ground truth fog.

While the CycleGAN model produced visually stylized results, it deviated from the real fog distribution and showed higher perceptual errors due to its unpaired, non-physical training. The Koschmieder model, though physically interpretable, remained close to the original (cloudy) input domain and failed to produce a significant structural transformation.

These findings confirm that our discretized RTE-based model effectively bridges physical plausibility and visual fidelity. The 2D PCA feature-space visualization further demonstrated that the generated fog aligns with real fog–cloudy image pairs, unlike GAN outputs, which tend to cluster near the general fog group, and Koschmieder results, which barely deviate from the source domain.

Despite its strengths, our implementation has several limitations:

- the use of first-order recursion may omit higher-order scattering terms,
- the fog is assumed static and spatially stationary, neglecting temporal effects,
- the Henyey–Greenstein phase function, while anisotropic, is still a simplification of full Mie scattering,
- angular resolution is fixed and limited due to memory constraints, affecting directional precision.

Training was performed using gradient-based optimization over 50 iterations per image. Although the algorithm requires more computational resources than analytical or GAN-based models, our results show that moderate-resolution synthesis remains tractable with modern hardware.

In summary, the proposed RTE-based fog simulation framework achieves a favorable trade-off between physical accuracy, perceptual quality, and computational feasibility. Its modular design makes it suitable for integration into differentiable image processing pipelines or as a data augmentation tool in adverse weather modeling — particularly in safety-critical domains such as autonomous driving.

Chapter 5

Summary

This dissertation addresses two major challenges in the field of computer vision: increasing the robustness of deep neural networks via ensemble learning, and generating physically realistic foggy images using a physics-based image synthesis pipeline.

In the first part of the thesis, a novel ensemble learning method is introduced that jointly trains multiple CNNs with a correlation-based penalization term. The approach encourages diversity among ensemble members by penalizing positively correlated outputs. The penalty term is computed using the Pearson correlation between the penultimate feature representations of the CNNs. The method is evaluated on multiple medical and general vision datasets, consistently outperforming both individual models and traditional ensembles in classification accuracy. The proposed model architecture and loss formulation show strong generalization and robustness across data domains.

The second part of the dissertation presents a physically inspired method for generating synthetic foggy images. This method utilizes a discretization of the RTE, which enables the modeling of inhomogeneous fog density and anisotropic scattering. By doing so, it addresses the limitations found in traditional Koschmieder-type homogeneous fog models.

The method is evaluated using various perceptual and physical realism metrics. This demonstrates that it achieves greater perceptual and physical realism in the simulated fog effects compared to learning-based synthesis methods, such as GAN, and traditional analytical models. Additionally, the synthetic dataset generated by this approach proves valuable for training and assessing fog-robust computer vision models.

Chapter 6

Összefoglaló

Ez a disszertáció a számítógépes látás két jelentős kihívásával foglalkozik: egyrészt a mély neurális hálózatok robusztusságának növelésével ensemble tanulás alkalmazásán keresztül, másrészt fizikailag valósághű ködös képek generálásával egy fizikai alapokon nyugvó képszintézisrendszer segítségével.

A dolgozat első részében egy új ensemble tanulási módszer kerül bemutatásra, amely több konvolúciós neurális hálózat (CNN) együttes tanítását valósítja meg egy korrelációalapú büntető tag bevezetésével. A megközelítés célja a modellek közötti diverzitás elősegítése azáltal, hogy bünteti a pozitívan korrelált kimeneteket. A büntető tag a hálók utolsó rejtett rétegbeli reprezentációi közötti Pearson-féle korreláció alapján kerül kiszámításra. A módszer több általános és orvosi képadatbázison is kiértékelésre került, és következetesen felülmúlta mind az egyedi modelleket, mind a hagyományos ensemble technikákat az osztályozási pontosság tekintetében. A javasolt modellarchitektúra és veszteségfüggvény erős generalizációs képességet és robusztusságot mutatott különböző adattartományokon.

A disszertáció második része egy fizikailag megalapozott módszert mutat be szintetikus ködös képek generálására. Az eljárás a radiatív transzfer egyenlet (RTE) diszkretizált alakját alkalmazza, amely lehetővé teszi az inhomogén ködsűrűség és az anizotróp szórás modellezését. Ezzel a megközelítés túllép a hagyományos, homogén (Koschmieder-típusú) ködmodellek korlátain.

A módszert különböző perceptuális és fizikai realizmust mérő mutatók segítségével értékeltük. Az eredmények azt mutatják, hogy az eljárás nagyobb fokú perceptuális és fizikai realizmust ér el a szimulált

ködhatások tekintetében, mint a tanulásalapú (például GAN-alapú) képszintézis-módszerek, illetve a hagyományos analitikus modellek. Emellett az így generált szintetikus adathalmaz értékes eszköznek bizonyult ködre robusztus számítógépes látási modellek tanításához és kiértékeléséhez.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Dr. András Hajdu, for his continuous guidance, invaluable insights, and support throughout my doctoral studies. His scientific expertise and mentorship have been instrumental in shaping both the direction and the quality of this dissertation.

I am equally thankful to Dr. Balázs Harangi, whose advice and technical suggestions have significantly contributed to the development of the ideas presented in this work. I sincerely appreciate his willingness to provide feedback and his openness whenever I reached out for help.

On a more personal note, I wish to thank my parents and my wife for their unceasing encouragement and patience during the many stages of this research. Your belief in me has been a constant source of strength.

Last but certainly not least, I owe heartfelt thanks to our little daughter, Olívia, who, in her own special way, supported this dissertation by providing me with the quietest work environment a baby could ever offer. Her gentle presence gave me both motivation and perspective during the writing of my papers and this thesis.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[2] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[4] Andrew Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv*, 04 2017.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[6] Xiaowei Liu, Yikun Hu, and Jianguo Chen. Hybrid cnn-transformer model for medical image segmentation with pyramid convolution and multi-layer perceptron. *Biomedical Signal Processing and Control*, 86:105331, 2023.

[7] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows . In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, Los Alamitos, CA, USA, 2021. IEEE Computer Society.

[8] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16514–16524, 2021.

[9] Zihang Dai, Hanxiao Liu, Quoc V. Le, and Mingxing Tan. Coatnet: marrying convolution and attention for all data sizes. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc.

[10] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. Cmt: Convolutional neural networks meet vision transformers. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12165–12175, 2022.

[11] Y. Liu and X. Yao. Ensemble learning via negative correlation. *Neural Networks*, 12(10):1399–1404, 1999.

[12] S. Chandrasekhar. *Radiative Transfer*. Dover Books on Physics. Dover Publications, 2013.

[13] Stephan Lenor. *Model-Based Estimation of Meteorological Visibility in the Context of Automotive Camera Systems*. Phd thesis, Heidelberg University, January 2016.

[14] Y. Zhang, K. Sohn, R. Villegas, G. Pan, and H. Lee. Improving object detection with deep convolutional networks via bayesian optimization and structured prediction. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 249–258, June 2015.

[15] D. Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 628–635, June 2013.

[16] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. from `http://arxiv.org/abs/1312.6229`.

[17] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[18] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based r-cnns for fine-grained category detection. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 834–849, Cham, 2014. Springer International Publishing.

[19] Ruixin Yang and Yingyan Yu. Artificial convolutional neural network in object detection and semantic segmentation for medical imaging analysis. *Frontiers in Oncology*, 11, 2021.

[20] Leila Abdelrahman, Manal Al Ghamdi, Fernando Collado-Mesa, and Mohamed Abdel-Mottaleb. Convolutional neural networks for breast cancer detection in mammography: A survey. *Computers in Biology and Medicine*, 131:104248, 2021.

[21] Evgin Göçeri. Convolutional neural network based desktop applications to classify dermatological diseases. In *2020 IEEE 4th International Conference on Image Processing, Applications and Systems*, pages 138–143, 2020.

[22] D. R. Sarvamangala and Raghavendra Kulkarni. Convolutional neural networks in medical image understanding: a survey. *Evolutionary Intelligence*, 15:1–22, 03 2022.

[23] Jianhong Wu and Yingdong Ma. A cnn-transformer hybrid network for multi-scale object detection. In *2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–7, 2023.

[24] Min Huang, Weihao Yan, Wenhui Dai, and Jingyang Wang. Est-yolov5s: Sar image aircraft target detection model based on improved yolov5s. *IEEE Access*, 11:113027–113041, 2023.

[25] L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, Oct 1990.

[26] Parham M. Kebria, Abbas Khosravi, Syed Moshfeq Salaken, and Saeid Nahavandi. Deep imitation learning for autonomous vehicles based on convolutional neural networks. *IEEE/CAA Journal of Automatica Sinica*, 7(1):82–95, 2020.

[27] Baihua Zhang, Shouliang Qi, Patrice Monkam, Chen Li, Fan Yang, Yu-Dong Yao, and Wei Qian. Ensemble learners of multiple deep cnns for pulmonary nodules classification using ct images. *IEEE Access*, 7:110358–110371, 2019.

[28] Andrey Kuehlkamp, Allan Pinto, Anderson Rocha, Kevin W. Bowyer, and Adam Czajka. Ensemble of multi-view learning classifiers for cross-domain iris presentation attack detection. *IEEE Transactions on Information Forensics and Security*, 14(6):1419–1431, 2019.

[29] Ayoub Abderrazak Maarouf and Fella Hachouf. Transfer learning-based ensemble deep learning for road cracks detection. In *2022 International Conference on Advanced Aspects of Software Engineering (ICAASE)*, pages 1–6, 2022.

[30] Xiangrong Zhang, Wenkang Ma, Chen Li, Jie Wu, Xu Tang, and Licheng Jiao. Fully convolutional network-based ensemble method for road extraction from aerial images. *IEEE Geoscience and Remote Sensing Letters*, 17(10):1777–1781, 2020.

[31] Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. Deep metric learning with bier: Boosting independent

embeddings robustly. *Jit IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):276–290, 2020.

[32] Ishtiaque Ahmed Khan, Asaduzzaman Sajeeb, and Shaikh Anowarul Fattah. An automatic ocular disease detection scheme from enhanced fundus images based on ensembling deep cnn networks. In *2020 11th International Conference on Electrical and Computer Engineering*, pages 491–494, 2020.

[33] Najd Alosaimi and Haikel Alhichri. Fusion of cnn ensemble for remote sensing scene classification. In *2020 3rd International Conference on Computer Applications Information Security (ICCAIS)*, pages 1–6, 2020.

[34] Ali Yazdizadeh, Zachary Patterson, and Bilal Farooq. Ensemble convolutional neural networks for mode inference in smartphone travel survey. *IEEE Transactions on Intelligent Transportation Systems*, 21(6):2232–2239, 2020.

[35] Rodrigo Minetto, Maurício Pamplona Segundo, and Sudeep Sarkar. Hydra: An ensemble of convolutional neural networks for geospatial land classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9):6530–6541, 2019.

[36] Muhammet Ali Dede, Erchan Aptoula, and Yakup Genc. Deep network ensembles for aerial scene classification. *IEEE Geoscience and Remote Sensing Letters*, 16(5):732–735, 2019.

[37] Long Wen, Liang Gao, and Xinyu Li. A new snapshot ensemble convolutional neural network for fault diagnosis. *IEEE Access*, 7:32037–32047, 2019.

[38] Sangdaow Noppitak and Olarik Surinta. dropcyclic: Snapshot ensemble convolutional neural network based on a new learning rate schedule for land use classification. *IEEE Access*, 10:60725–60737, 2022.

[39] Yushi Chen, Ying Wang, Yanfeng Gu, Xin He, Pedram Ghamisi, and Xiuping Jia. Deep learning ensemble for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(6):1882–1897, 2019.

[40] Shuxian Dong, Wei Feng, Yinghui Quan, Gabriel Dauphin, Lianru Gao, and Mengdao Xing. Deep ensemble cnn method based on sample expansion for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–15, 2022.

[41] Peng Tang, Qiaokang Liang, Xintong Yan, Shao Xiang, and Dan Zhang. Gp-cnn-dtel: Global-part cnn model with data-transformed ensemble learning for skin lesion classification. *IEEE Journal of Biomedical and Health Informatics*, 24(10):2870–2882, 2020.

[42] Le Zhang, Zenglin Shi, Ming-Ming Cheng, Yun Liu, Jia-Wang Bian, Joey Tianyi Zhou, Guoyan Zheng, and Zeng Zeng. Nonlinear regression via deep negative correlation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(3):982–998, 2021.

[43] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Diversity with cooperation: Ensemble methods for few-shot classification. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3722–3730, 2019.

[44] Naoki Okamoto, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. Deep ensemble learning by diverse knowledge distillation for fine-grained object classification. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 502–518, Cham, 2022. Springer Nature Switzerland.

[45] Shuo Wang, Huanhuan Chen, and Xin Yao. Negative correlation learning for classification ensembles. In *The 2010 International Joint Conference on Neural Networks*, pages 1–8, 2010.

[46] Balazs Harangi. Skin lesion classification with ensembles of deep convolutional neural networks. *Journal of Biomedical Informatics*, 86:25–32, 2018.

[47] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, May 2017.

[48] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[49] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, June 2016.

[50] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. from `http://arxiv.org/abs/1409.1556`.

[51] Balazs Harangi, Agnes Baran, and Andras Hajdu. Classification of skin lesions using an ensemble of deep neural networks. In *40th annual international conference of the IEEE engineering in medicine and biology society*, pages 2575–2578. IEEE, 2018.

[52] Balazs Harangi, Agnes Baran, Marcell Beregi-Kovacs, and Andras Hajdu. Composing diverse ensembles of convolutional neural networks by penalization. *Mathematics*, 11(23), 2023. (Q2, IF: 2.3, SJR: 0.498).

[53] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2261–2269, 2017.

[54] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[55] Yair Neuman. *Computational Personality Analysis: Introduction, Practical Applications and Novel Directions*. Springer Publishing Company, Incorporated, 1st edition, 2016.

[56] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3320–3328, Cambridge, MA, USA, 2014. MIT Press. from `http://dl.acm.org/citation.cfm?id=2969033.2969197`.

[57] Evgin Goceri. Diagnosis of skin diseases in the era of deep learning and mobile technology. *Computers in Biology and Medicine*, 134:104458, 2021.

[58] Evgin Göçeri. An application for automated diagnosis of facial dermatological diseases. *İzmir Katip Çelebi Üniversitesi Sağlık Bilimleri Fakültesi Dergisi*, 6(3):91 – 99, 2021.

[59] Evgin Goceri. Skin disease diagnosis from photographs using deep learning. In João Manuel R. S. Tavares and Renato Manuel Natal Jorge, editors, *VipIMAGE 2019*, pages 239–246, Cham, 2019. Springer International Publishing.

[60] Vipin Venugopal, Justin Joseph, M. Vipin Das, and Malaya Kumar Nath. An efficientnet-based modified sigmoid transform for enhancing dermatological macro-images of melanoma and nevi skin lesions. *Computer Methods and Programs in Biomedicine*, 222:106935, 2022.

[61] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, 2017.

[62] Catarina Barata, M. Emre Celebi, and Jorge S. Marques. A survey of feature extraction in dermoscopy image analysis of skin cancer. *IEEE Journal of Biomedical and Health Informatics*, 23(3):1096–1109, 2019.

[63] Roy Prasun, Ghosh Subhankar, Bhattacharya Saumik, and Pal Umapada. Effects of degradations on deep neural network architectures. *ArXiv*, 07 2018. from `https://arxiv.org/pdf/1807.10108.pdf`.

[64] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 - Canadian Institute for Advanced Research. *MIT*, 2009. from `http://www.cs.toronto.edu/~kriz/cifar.html`.

[65] Kaggle. Diabetic Retinopathy Detection, 2015. from `https://www.kaggle.com/c/diabetic-retinopathy-detection`.

[66] Etienne Decencière, Xiwei Zhang, Guy Cazuguel, Bruno Lay, Béatrice Cochener, Caroline Trone, Philippe Gain, Richard Ordonez, Pascale Massin, Ali Erginay, Béatrice Charton, and Jean-Claude Klein. Feedback on a publicly distributed database: the messidor database. *Image Analysis & Stereology*, 33(3):231–234, Aug 2014.

[67] Prasanna Porwal, Samiksha Pachade, Ravi Kamble, Manesh Kokare, Girish Deshmukh, Vivek Sahasrabuddhe, and Fabrice Meriaudeau. Indian diabetic retinopathy image dataset (idrid): A database for diabetic retinopathy screening research. *Data*, 3(3), 2018.

[68] Noel C. F. Codella, David Gutman, M. Emre Celebi, Brian Helba, Michael A. Marchetti, Stephen W. Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, and Allan Halpern. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In *2018 IEEE 15th International Symposium on Biomedical Imaging*, pages 168–172, 2018.

[69] Jean-Philippe Tarel and Nicolas Hautière. Fast visibility restoration from a single color or gray level image. In *2009 IEEE 12th International Conference on Computer Vision*, pages 2201–2208, 2009.

[70] Jean-Philippe Tarel, Nicolas Hautière, Aurélien Cord, Dominique Gruyer, and Houssam Halmaoui. Improved visibility of road scene images under heterogeneous fog. In *2010 IEEE Intelligent Vehicles Symposium*, pages 478–485, 2010.

[71] Bing Zhu, Yinzi Huang, Jian Zhao, Peixing Zhang, Jiayi Han, and Dongjian Song. Synthetic image generation model for intelligent vehicle camera function testing in rain and fog. *IEEE Transactions on Intelligent Transportation Systems*, 26(4):5332–5347, 2025.

[72] Simeon Geiger, André Liemert, Dominik Reitzle, Mario Bijelic, Andrea Ramazzina, Werner Ritter, Felix Heide, and Alwin Kienle.

Single scattering models for radiative transfer of isotropic and cone-shaped light sources in fog. *Opt. Express*, 31(1):125–142, Jan 2023.

[73] Zaher Ghais. Simulating weather conditions on digital images. Master's thesis, Univercity of Debrecen, 2020.

[74] Blessing Agyei Kyem, Joshua Kofi Asamoah, Ying Huang, and Armstrong Aboah. Weather-adaptive synthetic data generation for enhanced power line inspection using stargan. *IEEE Access*, 12:193882–193901, 2024.

[75] Yating Lin, Yidong Li, Haidong Cui, and Zheng Feng. Weagan:generative adversarial network for weather translation of image among multi-domain. In *2019 6th International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC)*, pages 1–5, 2019.

[76] Fayçal Abbas and Med Babahenini. Forest fog rendering using generative adversarial networks. *The Visual Computer*, 39:1–10, 01 2022.

[77] Rui Gong, Dengxin Dai, Yuhua Chen, Wen Li, Danda Paudel, and Luc Gool. Analogical image translation for fog generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35:1433–1441, 05 2021.

[78] Hanqing Zhang, Qitao Dan, and Lingfeng Wang. From point to surface: Realistic and perceptually-plausible hazy image generation with glow-diffusion. In *Pattern Recognition and Computer Vision: 7th Chinese Conference, PRCV 2024, Urumqi, China, October 18–20, 2024, Proceedings, Part X*, page 89–102, Berlin, Heidelberg, 2024. Springer-Verlag.

[79] P. Jolivet, M.A. Badri, and Y. Favennec. Deterministic radiative transfer equation solver on unstructured tetrahedral meshes: Efficient assembly and preconditioning. *Journal of Computational Physics*, 437:110313, 2021.

[80] David Hevisov, André Liemert, Dominik Reitzle, and Alwin Kienle. Impact of multi-scattered lidar returns in fog. *Sensors*, 24(16), 2024.

[81] Pierre Duthon, Michèle Colomb, and Frédéric Bernardin. Light transmission in fog: The influence of wavelength on the extinction coefficient. *Applied Sciences*, 9(14), 2019.

[82] B. W. Fowler and C. C. Sung. Radiative transfer in two dimensions through fog. *Appl. Opt.*, 17(11):1797–1805, Jun 1978.

[83] Amine Ben-Daoued, Pierre Duthon, and Frédéric Bernardin. Sweet: A realistic multiwavelength 3d simulator for automotive perceptive sensors in foggy conditions. *Journal of Imaging*, 9(2), 2023.

[84] Andrea Ramazzina, Mario Bijelic, Stefanie Walz, Alessandro Sanvito, Dominik Scheuble, and Felix Heide. Scatternerf: Seeing through fog with physically-based inverse neural rendering. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17911–17922, 2023.

[85] Zi-Xin Li, Yu-Long Wang, Qing-Long Han, and Chen Peng. Zrd-net: zero-reference image defogging by physics-based decomposition–reconstruction mechanism and perception fusion. *The Visual Computer*, 40:1–18, 10 2023.

[86] Zhixiong Guo and Shigenao Maruyama. Radiative heat transfer in inhomogeneous, nongray, and anisotropically scattering media. *International Journal of Heat and Mass Transfer*, 43(13):2325–2336, 2000.

[87] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6629–6640, Red Hook, NY, USA, 2017. Curran Associates Inc.

[88] Marcell Beregi-Kovacs, Balazs Harangi, Andras Hajdu, and Gyorgy Gat. Generation of synthetic non-homogeneous fog by discretized radiative transfer equation. *Journal of Imaging*, 11(6), 2025. (Q1, IF: 3.3, SJR: 0.662).

[89] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as

a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[90] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9492–9502, 2024.

[91] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion. *Acta Numerica*, 26:305–364, 2017.

[92] Rene Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer . *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 44(03):1623–1637, March 2022.

[93] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12159–12168, 2021.

[94] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.

[95] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, 2018.

[96] Boyi Li, Wenqi Ren, Dengpan Fu, Dacheng Tao, Dan Feng, Wenjun Zeng, and Zhangyang Wang. Benchmarking single-image dehazing and beyond. *IEEE Transactions on Image Processing*, 28(1):492–505, 2019.

List of publications related to the dissertation

Journal articles in English

M. Beregi-Kovacs, B. Harangi, A. Hajdu and G. Gat, "Generation of Synthetic Non-Homogeneous Fog by Discretized Radiative Transfer Equation," in Journal of Imaging, vol. 11 (6), pp. 1-22, 2025. (Q1, IF: 3.3, SJR: 0.662)

B. Harangi, A. Baran, M. Beregi-Kovacs and A. Hajdu, "Composing Diverse Ensembles of Convolutional Neural Networks by Penalization," in Journal of Imaging, vol. 11 (23), pp. 1-19, 2023. (Q2, IF: 2.3, SJR: 0.498)

Other publication

Journal articles in English

G. Bogacsovics, A. Hajdu, R. Lakatos, M. Beregi-Kovács, A. Tiba, and H. Tomán, "Replacing the SIR epidemic model with a neural network and training it further to increase prediction accuracy," in Annales Mathematicae et Informaticae, vol. 53, pp. 73–91, Eszterházy Károly Egyetem Líceum Kiadó, 2021. (Q4, IF: 0.3, SJR: 0.159)

Conference papers

G. Bogacsovics, B. Harangi, M. Beregi-Kovács, D. Kupás, R. Lakatos, N. D. Serbán, A. Tiba, and J. Tóth, "Assessing conventional and deep learning-based approaches for named entity recognition in unstructured Hungarian medical reports," in 2024 IEEE 22nd World Symposium on Applied Machine Intelligence and Informatics (SAMI), pp. 000077–000082, IEEE, 2024.

M. Beregi-Kovács, Á. Baran and A. Hajdu, "Efficient Learning of Model Weights via Changing Features During Training," in 2020 IEEE 24th International Conference on Intelligent Engineering Systems (INES), pp. 43-48, IEEE, 2020