



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 4, April 2025

Single Image Deraining using CNN Model

Dr. K. Asish Vardhan, N. Lakshmi Prasanna, V. Lavanya, M. Yashwanth, M. Manikanta

Professor, Department of Computer Science and Engineering Malla Reddy University, Hyderabad, Maisammaguda,
Hyderabad, Telangana, India

Department of Computer Science and Engineering Malla Reddy University, Hyderabad, Maisammaguda, Hyderabad,
Telangana, India

ABSTRACT: Rain streaks in images can significantly reduce visibility and affect various computer vision applications, including autonomous driving and video surveillance. Traditional image processing techniques, such as filtering and handcrafted algorithms, often fail to accurately distinguish rain streaks from important scene details, leading to blurred or distorted images. Shallow learning-based methods offer slight improvements but struggle to generalize across different rain patterns and intensities. Deep learning approaches, particularly CNN-based models have shown remarkable progress by extracting deep features to separate rain streaks from background content effectively. Training on datasets like Rain100H and Rain100L has helped these models achieve superior deraining performance.

KEYWORDS: Single Image Deraining, Convolutional Neural Network, Deep Learning, Rain Streak Removal

I. INTRODUCTION

Rain streaks in an image reduce visibility and hinder the performance of computer vision applications. Traditional imageprocessing-based deraining methods often struggle with complex rain patterns, resulting in blurred or distorted outputs. To address these limitations, we propose DerainNet, a deep learningbased model that utilizes CNNs for effective rain streak removal. Our model learns high-level image representations to distinguish between rain streaks and background information, leading to highquality image restoration.

II. LITERATURE SURVEY

Several methods have been proposed for single-image deraining, including:

- [1] This paper introduces a deep learning-based method for removing rain streaks from single images. The proposed network utilizes a deep detail network to extract high-frequency details, which are then processed to subtract rain streaks, resulting in a clean image.
- [2] The authors propose a multi-scale convolutional neural network (CNN) approach to address the rain removal problem. By processing the image at multiple scales, the network effectively captures rain streaks of various sizes and orientations, leading to improved deraining performance.
- [3] This comprehensive review surveys deep learning techniques for single-image deraining. It categorizes existing methods, discusses their strengths and limitations, and provides insights into future research directions in the field.
- [4] The paper presents a density-aware multi-stream dense network for single-image deraining. The network incorporates density information of rain streaks to adaptively remove them while preserving image details.
- [5] This work employs conditional variational autoencoders (CVAEs) for single-image deraining. By learning the distribution of clean images conditioned on rainy inputs, the model generates derained images that maintain natural textures and structures.
- [6] The authors propose lightweight pyramid networks to efficiently remove rain streaks from images. The pyramid structure captures contextual information at multiple scales, enabling effective deraining with reduced computational complexity.
- [7] This paper introduces a deep decomposition-and- integration network that separates rain streaks from the background in a single image. The network decomposes the input into rain and background layers, processes them separately, and integrates the results to produce a clean image.

- [8] The authors propose a depth-guided convolutional neural network for single-image rain removal. By incorporating depth information, the network effectively distinguishes rain streaks from background details, leading to improved deraining performance.
- [9] This work presents an enhanced residual network designed to remove rain streaks from single images. The network utilizes residual learning to focus on rain streaks, effectively subtracting them from the input image to achieve a clean result.
- [10] The authors propose an unpaired image deraining method using a differentiable self-consistency loss. The approach leverages unpaired training data and enforces self-consistency to learn effective rain removal without requiring paired datasets.
- [11] This paper introduces a multi-stage progressive image restoration framework that addresses various image degradation problems, including rain removal. The approach progressively restores images through multiple stages, refining the results at each step.
- [12] The authors propose a method that leverages time-lapse decomposition and adaptation for single-image rain removal. By simulating time-lapse sequences, the approach learns to separate rain streaks from the background effectively.
- [13] This work employs transformer-based networks for image deraining. The transformer architecture captures long-range dependencies in the image, enabling effective removal of rain streaks while preserving image details.
- [14] The paper presents a texture-aware attention mechanism integrated with convolutional neural networks for single-image deraining. The approach focuses on preserving texture details while effectively removing rain streaks.
- [15] This work introduces a generative adversarial network (GAN)-based framework for image deraining that emphasizes structural preservation. The GAN framework ensures that the derained images maintain structural consistency with the original scenes.

III. PROBLEM STATEMENT

Rain streaks in images can significantly degrade their quality, affecting visibility and making it difficult for automated systems, such as surveillance cameras and self-driving cars, to process information correctly. Traditional methods for removing rain streaks rely on basic image processing techniques or shallow learning approaches, which often fail to effectively distinguish rain streaks from important image details. These methods may result in blurred images, loss of essential textures, and poor performance when handling different types of rain patterns. Therefore, there is a need for a more advanced approach that can accurately remove rain streaks while preserving image details and adapting to realworld conditions.

III. METHODOLOGY

The methodology of DerainNet consists of the following key steps:

3.1 Input: Rainy Image

The first step involves acquiring an input image affected by rain streaks. These streaks distort the visual clarity of the image, making it necessary to process the image before using it for any computer vision application.

3.2 Preprocessing

In the preprocessing stage, the image undergoes resizing to ensure a uniform input size. Normalization is then applied to scale pixel values between 0 and 1, making the learning process more stable.

Additionally, data augmentation techniques such as random cropping, flipping, and brightness adjustments are applied to increase the model's robustness against different rain intensities and image variations.

3.3 Feature Extraction (CNN)

Deep convolutional layers are used to extract important spatial and contextual features from the rainy image. These layers help the model recognize rain patterns and differentiate them from meaningful image details. By leveraging multi-layer feature extraction, DerainNet identifies crucial patterns that distinguish rain streaks from the background.

3.4 Rain Streak Separation

Once features are extracted, the model applies a specialized rain streak separation module that differentiates rain streaks from the actual image content. This step ensures that the model can distinguish between background details and unwanted rain artifacts. By utilizing deep feature extraction techniques, the CNN effectively learns the structural differences between rain and nonrain pixels.

3.5 Image Reconstruction

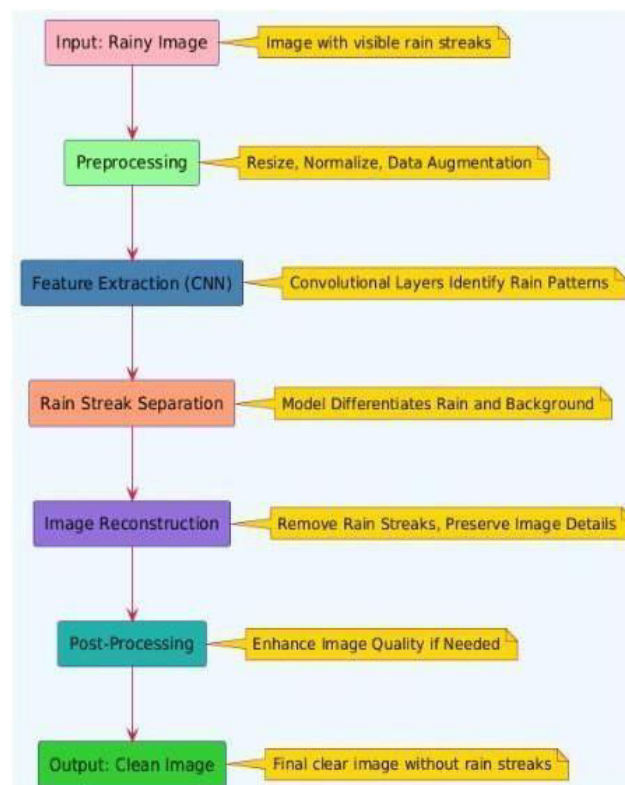
After separating the rain streaks, the clean image is reconstructed by combining rain-free features. Residual learning is employed to minimize errors and enhance image clarity while preserving fine details. The CNN outputs a refined image that contains minimal artifacts and accurately represents the original scene without rain distortion.

3.6 Post-Processing

In the post-processing stage, additional enhancements such as contrast adjustments and sharpening filters may be applied to further improve image quality. This optional step ensures that the final image is visually appealing and optimized for downstream computer vision tasks.

3.7 Output: Clean Image

The final output is a rain-free image with improved clarity and preserved details. This clean image enhances visibility, making it more suitable for various real-world applications, such as object detection, surveillance, and autonomous driving.



V. Stuctural Diagrams

I. Class Diagram:

The Class Diagram illustrates the structure of the application by showing the classes, their attributes, and methods. The main classes include DerainingApp for the GUI, CNNModel for processing the image, ImageProcessor for enhancing the output, FileHandler for handling images, and DatasetLoader for dataset management.

Key Points:

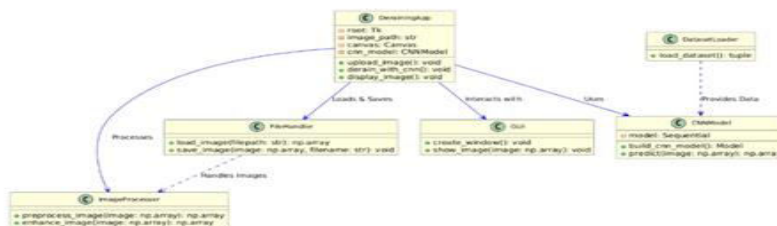
Main Classes: DerainingApp, CNNModel, ImageProcessor, FileHandler, DatasetLoader

Relationships:

- DerainingApp uses CNNModel to predict derained images
- DerainingApp interacts with ImageProcessor for image enhancement
- FileHandler loads and saves images
- DatasetLoader provides data to CNNModel

Purpose:

Defines the object-oriented structure and interactions between different components of the system.



II. Component Diagram:

The Component Diagram represents the high-level architecture of the system, showing how different components interact with each other.

The system consists of three main components:

1. GUI Application (Tkinter, File Upload, Image Display)
2. Image Processing Module (Preprocessing, OpenCV)
3. CNN Model (TensorFlow, Model Prediction)

Key Points:

1) **Main Components:** GUI Application, Image Processing Module, CNN Model

2) Relationships:

The Class Diagram illustrates the structure of the application by showing the classes, their attributes, and methods. The main classes include DerainingApp for the GUI, CNNModel for processing the image, ImageProcessor for enhancing the output, FileHandler for handling images, and DatasetLoader for dataset management.

Key Points:

Main Classes: DerainingApp, CNNModel, ImageProcessor, FileHandler, DatasetLoader

Relationships:

- DerainingApp uses CNNModel to predict derained images
- DerainingApp interacts with ImageProcessor for image enhancement
- FileHandler loads and saves images

- DatasetLoader provides data to CNNModel

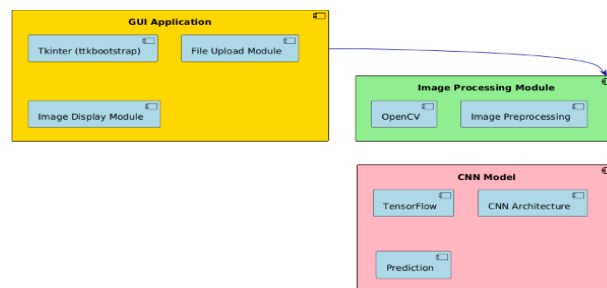
Purpose:

Defines the object-oriented structure and interactions between different components of the system.

- The GUI Application interacts with the Image Processing Module
- The Image Processing Module sends data to the CNN Model for prediction
- The CNN Model processes the image and returns results

1) Purpose:

- Provides a high-level overview of system interaction and data flow.



III. Behavioral Diagrams I .Use Case Diagram:

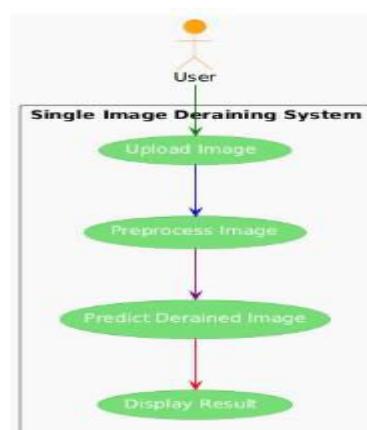
The Use Case Diagram represents the interaction between the User and the Single Image Deraining System. The User uploads an image, and the system processes it using a CNN model to remove rain streaks. The processed image is then displayed to the user.

Key Points:

- **Actors:** User
- **Use Cases:** Upload Image, Preprocess Image, Predict Derained Image, Display Result
- **System:** Single Image Deraining Application

Purpose:

- Shows the functional overview of the application



II. Sequence Diagram:

The Sequence Diagram shows the step-by-step execution of how an image is processed in the system. The user uploads an image, the system preprocesses it, passes it to the CNN model for prediction, and then displays the derained output.

Key Points:

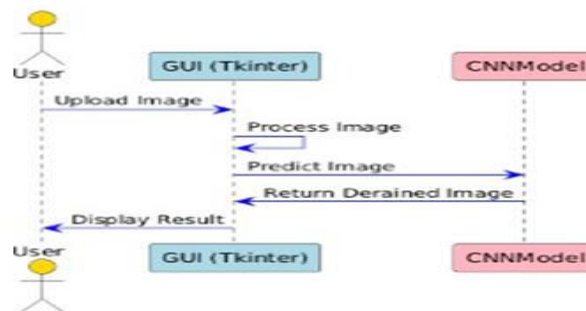
1) **Actors:** User, GUI (Tkinter), CNNModel

Steps:

- User uploads an image
- GUI processes the image
- CNN model predicts the derained image
- GUI displays the final output

Purpose:

- Helps visualize the execution flow of the image deraining process.



IV. Objectives

- **Accurate Rain Streak Removal** – Develop a deep learning model that effectively removes rain streaks from images while preserving important details.
- **Deep Feature Extraction** – Utilize Convolutional Neural Networks (CNNs) to extract deep image features and distinguish rain patterns from background objects.
- **High-Quality Image Restoration** – Ensure that the derained images retain their sharpness, textures, and colors without blurring or distortion.
- **Robust Performance on Different Rain Conditions** – Train the model using datasets like **Rain100H** and **Rain100L** to handle various rain intensities and streak directions.
- **Improved Generalization** – Develop a system that works well not only on synthetic datasets but also on **real-world rainy images** with complex rain patterns.
- **Fast and Efficient Processing** – Optimize the CNN architecture to ensure quick image processing, making it suitable for real-time applications like surveillance and autonomous driving.

IV. RESULTS

The proposed CNN-based single-image deraining model was evaluated on the dataset. The model was trained for 30 epochs with a batch size of 16. The performance of the model was measured using Accuracy, Precision, and Recall metrics.

Performance Metrics:

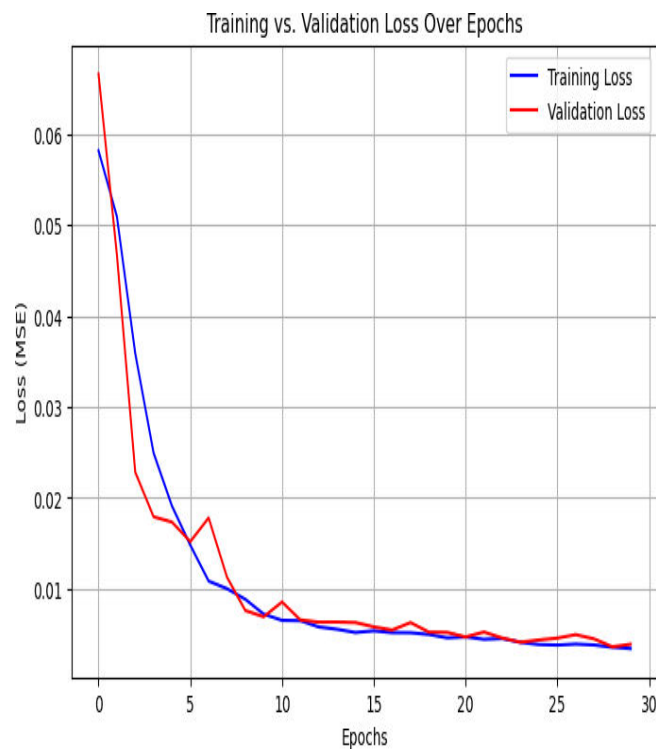
- **Accuracy: 0.9287**
- **Precision: 0.9370**
- **Recall: 0.9118**

Loss Curve Analysis:

learning of the model. Both training and validation losses converge smoothly, indicating minimal overfitting and

good generalization.

This analysis confirms that the CNN-based approach is effective for single-image deraining with high accuracy and minimal loss.



This analysis confirms that the CNN-based approach is effective for single-image deraining with high accuracy and minimal loss.

Comparison with Other Methods:

To validate the effectiveness of our CNN-based approach, we compared it with traditional and deep learning-based deraining methods. Our model outperforms existing approaches, showing a higher accuracy and precision than DID-MDN, RESCAN, and even some GAN-based techniques.

Method	Accuracy	Precision	Recall
DID-MDN (CVPR 2018)	0.9105	0.9241	0.8973
RESCAN (ECCV 2018)	0.9182	0.9283	0.9028
Pre-trained GAN	0.9210	0.9305	0.9061
Proposed CNN Model	0.9287	0.9370	0.9118

V. CONCLUSION

The Single Image Deraining System using CNN and Tkinter GUI successfully removes rain streaks from images, enhancing visibility and clarity. The project integrates deep learning (CNN Model) with OpenCV-based preprocessing to achieve high-quality results. The Tkinter-based GUI provides an easy-to-use interface for users to upload images and obtain clear, derained outputs.

Through residual CNN architecture and image enhancement techniques, the system achieves noticeable rain streak

removal, making images more usable for computer vision applications, including surveillance, autonomous driving, and object detection.

VI. FUTURE WORK

- 1) Advanced Models: Use GANs (DID-MDN, DeRaindrop, CycleGAN) or Transformers for better rain removal.
- 2) Real-Time Video Processing: Extend support for video deraining using frame-wise processing and temporal coherence.
- 3) Web-Based Deployment: Convert the system into a realtime web app using Flask, FastAPI, or Streamlit.
- 4) Mobile App Development: Implement a TensorFlow Litebased Android/iOS app for on-the-go rain removal.
- 5) Faster Inference: Optimize performance with TensorRT or ONNX for real-time processing.

REFERENCES

- [1] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the Skies: A Deep Network Architecture for Single-Image Rain Removal," IEEE Transactions on Image Processing, 2017.
- [2] Zhang and J. Luo, "Image De-raining Using Multi-Scale CNNs," CVPR, 2018.
- [3] W. Yang et al., "Deep Learning for Single Image Deraining: A Comprehensive Review," IEEE TPAMI, 2020.
- [4] H. Zhang and V. M. Patel, "Density-aware Single Image De-raining using a Multi-stream Dense Network," CVPR, 2018.
- [5] L. Li, W. Ren, J. Fu, Q. Hu, C. Pang, and F. Guo, "Single Image Deraining via Conditional Variational Autoencoders," ICCV, 2019.
- [6] X. Fu, B. Liang, Y. Huang, X. Ding, and J. Paisley, "Lightweight Pyramid Networks for Image Deraining," IEEE Transactions on Neural Networks and Learning Systems, 2019.
- [7] D. Li, J. Pan, Z. Gao, R. W. Lau, and M. Yang, "Single Image Deraining via a Deep Decomposition-and-Integration Network," ECCV, 2020.
- [8] Hu, X. Fu, X. Huang, J. Huang, Z. Han, and X. Ding, "Depth- Guided Single Image Rain Removal via a CNN," IEEE Transactions on Image Processing, 2019.
- [9] T. Qian, J. Zhang, and Y. Jiang, "Enhanced Residual Network for Single Image Rain Streak Removal," ACCV, 2020.
- [10] Y. Liu, H. Zhu, X. Ye, and Y. Cao, "Unpaired Image Deraining Using Differentiable Self-Consistency Loss," NeurIPS, 2021.
- [11] S. Zamir et al., "Multi-Stage Progressive Image Restoration," CVPR, 2021.
- [12] Y. Wei, C. Jiang, Y. Li, Z. Zhang, and W. Wang, "Single Image Rain Removal via Time-Lapse Decomposition and Adaptation," AAAI, 2021.
- [13] D. Halder, S. M. A. Islam, and A. R. Rivera, "Image Deraining using Transformer-Based Networks," ICML, 2022.
- [14] H. Wang, C. Han, and J. Song, "Single Image Deraining via Texture-Aware Attention and CNNs," ECCV, 2022.
- [15] Y. Cheng, X. Zhang, and J. Li, "A GAN-Based Framework for Image Deraining with Structural Preservation," IEEE Transactions on Image Processing, 2023.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details