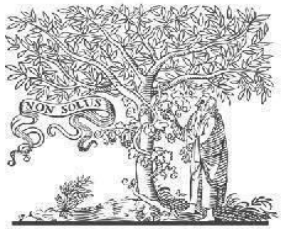
International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

COPY RIGHT



ELSEVIER SSRN

IJIEMR Transactions, online available on 31th Nov 2024. Link

https://ijiemr.org/downloads.php?vol=Volume-13&issue= Issue11

DOI:10.48047/IJIEMR/V13/ISSUE11/14

Title: " ENHANCING VISUAL CLARITY: A DEEP LEARNING APPROACH FOR HAZE REMOVAL"

Volume 13, ISSUE 11, Pages: 125- 145

Paper Authors
K. Gnana Prasuna

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper as Per UGC Guidelines We Are Providing A ElectronicBar code

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

ENHANCING VISUAL CLARITY: A DEEP LEARNING APPROACH FOR HAZE REMOVAL

K. Gnana Prasuna

Assistant Professor, Department of CSE,
G. Narayanamma Institute of Technology and Science(for Women),
Hyderabad, Telangana, India
gnanaprasuna@gnits.ac.in

ABSTRACT

In computer vision, image dehazing is an essential problem with applications ranging from surveillance systems to autonomous driving. In complicated settings with varied levels of haze intensity, traditional approaches frequently fail to remove haze efficiently. Deep learning techniques have demonstrated encouraging outcomes in tackling this issue in recent times. These techniques may efficiently describe the intricate links between hazy and haze-free images, and can train to recover clear images from hazy inputs by utilizing convolutional neural networks (CNNs) and generative adversarial networks (GANs). Recent developments in deep learning-based picture dehazing algorithms are examined in this abstract, together with their architectures, training approaches, and performance evaluation measures. Present issues and potential future directions in this quickly developing field are also highlighted. This project is mainly based on the dehazing the images using deep learning first get the input as the haze images and then pre-process it then apply splitting the data into the ratio of 8:2 and then apply deep learning algorithm for getting the ground truth images and then finally calculate the performance metrics such as MSE, RMSE are calculated for the retrievedimages.

1. INTRODUCTION

1.1 Motivation

Haze and fog contribute to numerous road and aviation accidents: a survey by the Times of India reports that approximately 11,000 people lose their lives annually in accidents caused by fog or haze, with the number rising each year. Several plane and helicopter crashes have also been attributed to low visibility caused by these conditions. Our goal is to develop an intelligent dehazing system designed to enhance the visibility and quality of images and videos captured in foggy or hazy environments. The system's primary function will be to process hazy input images

or video frames and produce clear, haze-free outputs.

Haze significantly impacts the performance of machine vision systems by reducing scene visibility, which in turn diminishes the effectiveness of tasks like object detection, target tracking, and semantic segmentation. As a result, image dehazing has become a crucial area in computer vision. Removing haze not only reduces noise and sharpens edges but also improves the signal-to-noise ratio, enhancing the overall performance and robustness of vision systems. For these reasons, there is a high demand for image dehazing across various fields, including security surveillance, assisted driving, weather analysis, and geographic information research.

A hazy image can be understood as a superposition of reflected light that has been attenuated by fog and light that has been scattered by the atmosphere due to the attenuation and scattering effects of haze. The atmospheric scattering model can be used to explain the imaging process of the fuzzy image. Low contrast, low signal-to-noise ratio, and colour distortion are characteristics of hazy photographs. As a result, the algorithms used for image dehazing concentrate on eliminating noise, emphasising edges, enhancing contrast, and recovering colours.

The method of image dehazing is evolving along with advances in image processing technology. The image dehazing issue was first approached as an image enhancement issue, using Retinex for defogging and contrast enhancement being two of the methods used. Afterwards, the image dehazing issue is resolved by employing image restoration techniques based on the atmospheric scattering model. Examples of representative techniques include Non-local Dehazing [2], Dark Channel Prior [1].

1.2 Existing Systems

The traditional image dehazing methods have investigated various haze-related image priors. Assumes that the contrast of hazy images is lower than haze-free images and proposes to maximize the contrast of hazy images under the MRF framework. Fattal uses independent component analysis for estimating the transmission in hazy scenes assuming that the transmission and surface shading are locally uncorrelated. He Proposed the dark channel prior to estimate the transmission map based on the observation that the local minimums of color channels of haze-free images are close to zero. Liu et al. proposed the rank-one prior based on the observation that in most of the regions except the light source area, the imaging scene is covered by spatially homogeneous light. Polarization-based methods are also effective for haze removal, which are based on the fact that the air light scattered by atmospheric particles is partially polarized. These methods are effective in image dehazing due to the full investigation of image prior knowledge and the understanding of the physical mechanism of hazes. However, these priors are mainly based on human observations that would not always hold for diverse real-world images. For example, dark channel prior is effective for most outdoor images but usually fails for those containing large areas of white scenery such as white walls or clouds in the sky. Despite the effectiveness, polarization-based dehazing requires professional optical equipment and multiple image fusion, which is not suitable for single image dehazing. Deep learning-based

dehazing methods frequently need a lot of labelled data for training, which can be difficult and expensive to get, especially for certain scenarios or domains. These models can have a large computational burden, which makes real-time dehazing unfeasible in some circumstances. Deep learning-based dehazing techniques might have trouble generalizing to unknown or extreme atmospheric conditions, which could leadto less-than-ideal performance in some circumstances.

1.3 Literature Survey

For haze removal wide range of methods are available. This study, conducted on a synthetic hazy dataset comprising images with varying levels of visual range, demonstrates that the proposed approach excels in accuracy, efficiency, and real-time performance for haze degree classification. Additionally, we developed two dehazing models—a haze-dense model and a haze-light model—which are selected based on the classified haze degree for optimal dehazing. Compared to dehazing methods without prior classification, incorporating haze degree classification before dehazing significantly improves both subjective and objective evaluations. The haze classification and dehazing techniques offer a solid foundation for subsequent computer vision tasks, such as autonomous driving, target detection, and image recognition[3]. A comprehensive survey of numerous algorithms proposed for haze removal, which enhance image clarity either through statistical scene observations or network-based learning methods. These algorithms are evaluated both quantitatively and visually to compare their dehazing capabilities. The paper provides a comparative analysis of various image dehazing techniques, each employing different methods for estimating transmission and atmospheric light. Additionally, it reviews both prior-based and learning-based dehazing algorithms[4].

1.4 Challenges in existing system

Traditional dehazing methods often come with several limitations, including:

1. **Sensitivity to Parameters**: Many traditional dehazing algorithms require tuning of multiple parameters, such as airlight estimation, transmission estimation, and atmospheric light estimation. The effectiveness of these methods can be highly dependent on the accurate setting of these parameters, which can be challenging and subjective.

2. **Limited Generalization:** Traditional dehazing methods may not generalize well to different types of haze and diverse scenes. They are often designed based on specific assumptions about the atmospheric conditions or scene characteristics, limiting their applicability to real-world scenarios with varying atmospheric conditions and scene complexities.

3. **Loss of Image Details:** Some traditional dehazing algorithms may inadvertently result in the loss of image details or introduce artifacts during the dehazing process. This can occur due to over-smoothing or over-enhancement of certain image regions, leading to degraded image quality andloss of important visual information.

4. **Computational Complexity:** Certain traditional dehazing algorithms can be computationally intensive, requiring significant processing time and resources, especially for high-resolution images or real-time applications. This can limit their practicality for applications with strict computationalconstraints or real-time processing requirements.

5. **Limited Robustness:** Traditional dehazing methods may struggle to handle challenging scenarios such as scenes with complex lighting conditions, heterogeneous haze distributions, or scenes containing thin haze layers. They may fail to produce satisfactory results in such cases, leading to suboptimal dehazing outcomes.

6. **Single Image Dependency:** Many traditional dehazing algorithms rely solely on information contained within a single hazy image for dehazing. This can limit their effectiveness, especially when dealing with severely degraded or low-quality hazy images lacking sufficient information for accuratedehazing.

7. **Inability to Handle Non-Uniform Haze:** Traditional dehazing methods may struggle to effectively handle scenes with non-uniform haze distributions, where different regions of the image are affected by varying degrees of haze. This can result in uneven dehazing effects or inconsistencies in the dehaze image.

1.5 Proposed System

The encoder-decoder model for image dehazing is a deep learning architecture designed toautomatically remove haze from images. Here's a detailed description of how this model works:

Encoder:

The encoder is typically composed of a series of convolutional layers followed by pooling layers. Convolutional layers extract features from the input hazy image, capturing information about the scene structure, lighting conditions, and haze distribution. Each convolutional layer applies a set of filters to the input image, resulting in feature maps that encode increasingly abstract and hierarchical representations of the image.

Decoder:

The decoder is responsible for reconstructing a haze-free version of the input image from the encoded features. It consists of a series of up-sampling layers followed by convolutional layers.

Up-sampling layers increase the spatial dimensions of the encoded feature maps while reducing the number of channels, gradually reconstructing the high-resolution dehazed image. The decoder learns to generate the haze-free image by combining the encoded features with spatial information obtained from the up-sampling layers.

Skip Connections:

Skip connections are introduced between corresponding layers of the encoder and decoder.

These connections enable the decoder to access low-level spatial information from the encoder, facilitating the reconstruction of fine details in the dehazed image. Skip connections also help mitigate the vanishing gradient problem during training, allowing for more effective learning of deep networks.

Loss Function:

During training, a loss function is used to measure the discrepancy between the predicted dehazed image and the ground truth haze-free image. Commonly used loss functions include mean squared error (MSE), perceptual loss, or a combination of multiple loss terms.

The model is trained to minimize this discrepancy, encouraging it to produce visually pleasing and perceptually accurate dehazed images.

Training:

The model is trained on a dataset of paired hazy and haze-free images. During training, the parameters of the encoder-decoder network are iteratively updated using backpropagation and gradient descent optimization techniques. The model learns to effectively remove haze from images by minimizing the difference between the predicted dehazed images and their corresponding groundtruth images.

Inference:

Once trained, the encoder-decoder model can be used to dehaze new unseen hazy images. During inference, a hazy input image is fed through the encoder to extract its feature representation, and then the decoder generates the corresponding haze-free output image. Overall, the encoder-decoder model for image dehazing leverages deep learning techniques to automatically learn to remove hazefrom images, producing clear and visually appealing results.

1.6 Objectives

- To detect Hazy to Dehaze effectively.
- To implement the Deep Learning CNN and VGG16 Algorithm .
- To enhance the overall performance analysis.
- To preprocess the densehaze dataset.
- To build an encoder decoder architecture .

1.7 Methodology

Single image dehazing using Convolutional Neural Networks (CNNs) involves training a model to learn the mapping between hazy images and their corresponding haze-free versions. These methodologies are designed to enhance the accuracy and effectiveness of the model.

Dataset Collection: Gather a dataset containing pairs of hazy images and their corresponding haze-free versions. These images can be collected from various sources, including synthetic hazy image generation algorithms or real-world hazy images with corresponding ground truth.

Preprocessing: Preprocess the dataset by resizing images to a fixed resolution, normalizing pixel values, and augmenting the data if necessary (e.g., rotation, flipping, brightness adjustments) to increase the diversity of the dataset.

Model Architecture Selection: Choose an appropriate CNN architecture for the dehazing task. You may start with commonly used architectures like VGG-Net, ResNet, or DenseNet, or you can design acustom architecture tailored to the specific requirements of the task.

Model Training: Split the dataset into training, validation, and testing sets.

Train the chosen CNN architecture using the training set. The input to the network should be hazyimages, and the ground truth should be corresponding haze-free images.

Define a suitable loss function such as mean squared error (MSE), perceptual loss, or a combination of multiple losses to measure the discrepancy between the predicted and ground truth images.

Utilize optimization algorithms such as Adam or RMSprop to minimize the loss function and updatethe model parameters.

Validation and Hyperparameter Tuning: Validate the trained model on the validation set to monitor its performance and prevent overfitting. Tune hyperparameters such as learning rate, batch size, and network architecture based on validation performance.

Evaluation: Evaluate the trained model on the testing set using metrics such as Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), or perceptual metrics to assess its generalization performance.

Post-processing: Apply any necessary post-processing techniques to further enhance the dehazedimages, such as contrast adjustment, gamma correction, or histogram equalization.

Deployment: Once satisfied with the performance of the trained model, deploy it for inference on new hazy images. Ensure that the deployment environment is suitable for running the model efficiently, whether it's on a local machine, cloud server, or embedded system.

1.8 Hardware and software requirements

Software Requirements

- O/S : Windows 7.
- Language : Python

- Front End : Anaconda Navigator – Spyder Notebook

Hardware Requirements

- System : Pentium IV 2.4 GHz
- Hard Disk : 200 GB
- Ram : 4GB

1.9 Organization of the project

In chapter 1, an overall view of the project like motivation , literature survey, existing system, drawbacks, proposed system, objectives, methodology, hardware and software requirements.This chapter gives us the basic insights of the project and explains about the already existing system and their respective drawbacks . In chapter 2, the proposed system is provided which consists of architecture of the system, UML diagrams and algorithms used. Architecture of the system speaks about the overall flow of the project .Algorithms used for the implementation of the project are specified.

In chapter 3, implementations provided which consist of module design, datasets used, technologies used, technologies used, test case design. Module design speaks about each module specifications, Technologies used chapter explains about various technologies that were used for implementation of the module. This chapter also helps us to go through how the test cases are designed and how the project is passing through test cases successfully.

In chapter 4, results and discussion are provided which consists of screenshots of the interface .This chapter also speaks about evaluation metrics.

In chapter 5,conclusion and future works with reference are provided which says about conclusion and any future work required for the system.This chapter also has few references and glossaries that are required.

2. PROPOSED SYSTEM

2.1 Architecture

Create a dataset by first matching up photographs that are unclear with images that are clear. Preprocess the photos to improve the dataset's diversity by resizing, normalising, and adding effects. In machine learning and statistics, the most pertinent and important features (or variables) are selected from the original dataset in order to create a prediction model. This process is known as feature selection. By choosing a subset of characteristics that are most useful for the target variable and removing redundant or irrelevant features, it seeks to minimise the dimensionality of the data. Selecting the right features is essential to increasing interpretability, decreasing overfitting, and optimising model performance.

The diagram below illustrates the methodology of the proposed system.

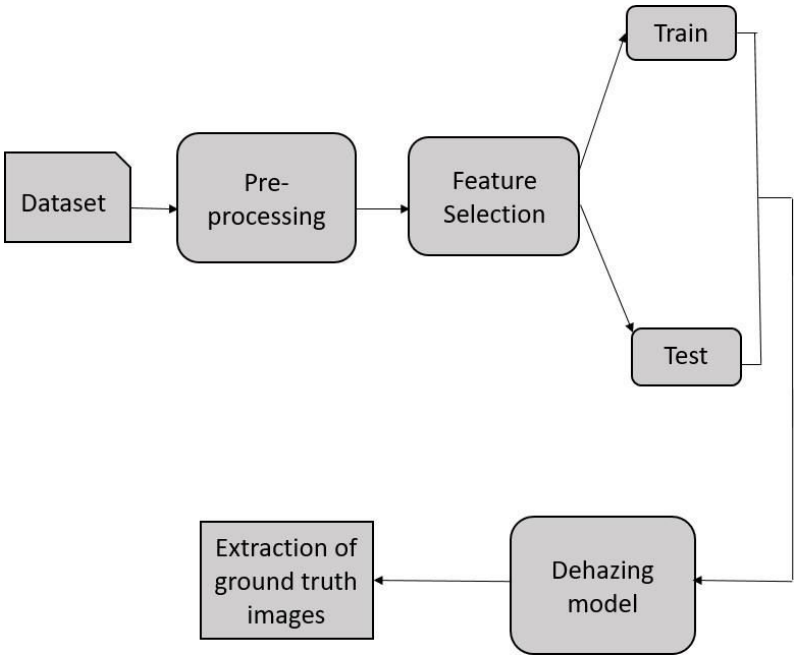


Figure 3.1.1. Methodology for image dehazing

Then Split the dataset into training, validation, and possibly test sets. Train the CNN using the training data by minimizing the defined loss function. Monitor the performance of the model on the validation set to prevent overfitting and adjust hyperparameters accordingly, such as learning rate or regularization strength. As the final step the dehazing model is trained and the ground truth imagesare extracted .

2.2 UML Diagrams

Unified Modeling Language (UML) diagrams are a standardized graphical notation used to visualize, specify, construct, and document the artifacts of a software system. UML diagrams provide a common language for software engineers, analysts, designers, and stakeholders to communicate and understand the various aspects of a system's architecture, structure, behavior, and interactions. Following are UML diagrams of the proposed model:

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

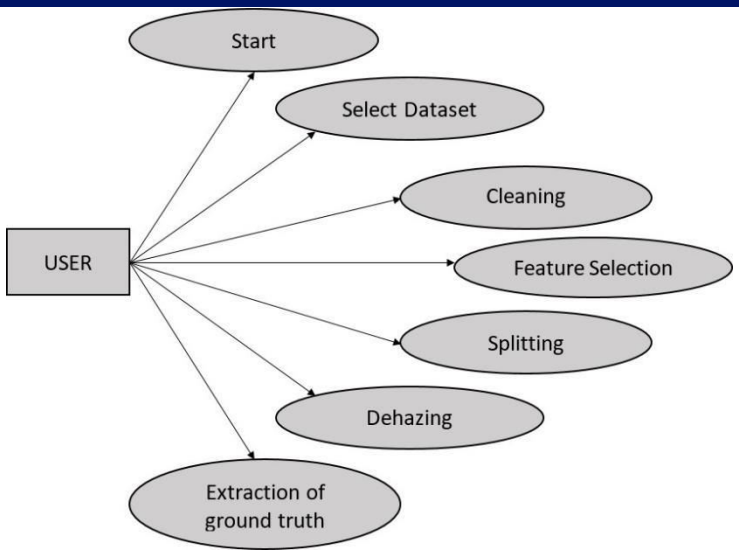


Figure 2.2.1. Use case diagram of image dehazing model

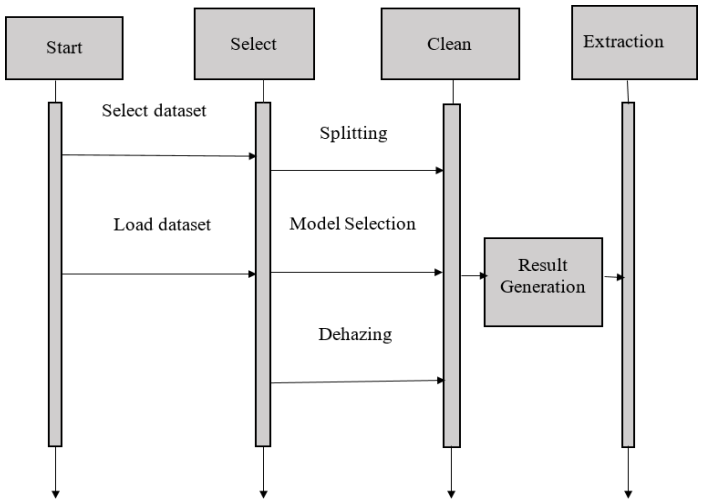


Figure 2.2.2. Sequence diagram of image dehazing model

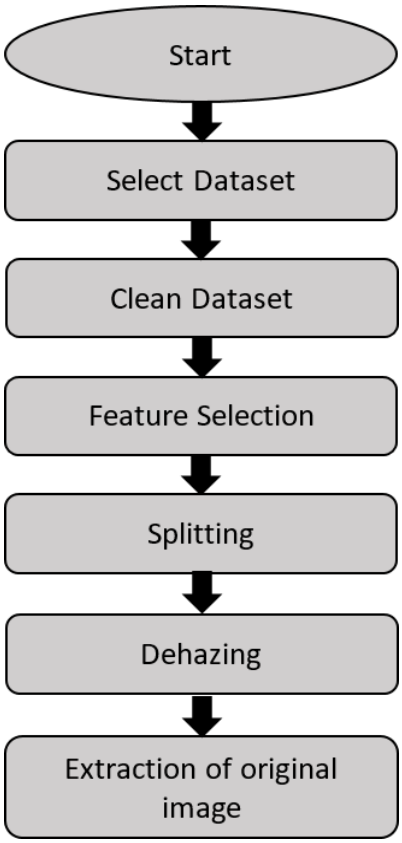


Figure 2.2.3. Activity diagram of proposed model

2.3 Description of algorithms

Following are the algorithms and techniques used in the proposed methodology of image dehazing.

1. **Bound context and regularization**:
   Bound Constraints:
   In optimization problems, bound constraints refer to limitations or restrictions imposed on the values that a variable can take within a certain range. These constraints define the feasible region of the optimization problem. Variables are typically constrained to remain within a lower and upper bound.

   Regularization: Common types of regularization include:

L1 Regularization (Lasso): Adds a penalty term proportional to the absolute value of model coefficients. It encourages some coefficients to be exactly zero, effectively performing feature selection.

L2 Regularization (Ridge): Adds a penalty term proportional to the square of model coefficients. It penalizes large coefficients and encourages them to be small.

Elastic Net Regularization: Combines both L1 and L2 regularization, offering a balance between feature selection and coefficient shrinkage.

Dropout Regularization: Used in neural networks, it randomly drops a fraction of neurons during training to prevent co-adaptation of features.

Regularization techniques are essential for maintaining model simplicity, reducing overfitting, and ensuring better generalization to new, unseen data. The strength of the regularization term (often controlled by a hyperparameter) determines the trade-off between fitting the training data and maintaining model simplicity.

Both bound constraints and regularization are valuable tools in optimization and machine learning, addressing different aspects of model behavior and constraints. Bound constraints limit variable values within a specific range, while regularization helps control model complexity and prevent overfitting by adding penalty terms during training.

2. Convolution neural network (CNN):

Convolutional Neural Networks (CNNs) have emerged as powerful tools in the domain of image dehazing, offering sophisticated solutions to the challenging task of restoring clear images from hazy inputs. Leveraging their ability to extract intricate patterns and features from data, CNNs can effectively learn the complex mappings between hazy and clear images. In the context of image dehazing, CNNs are typically employed within encoder-decoder architectures, which facilitate the extraction of hierarchical features from the input image and the subsequent reconstruction of the dehazed output.

The architecture of a CNN-based image dehazing model often follows an encoder-decoder paradigm. The encoder portion of the network consists of multiple convolutional layers followed by pooling operations, progressively reducing the spatial dimensions of the input image while enhancing the depth of feature representation. These convolutional layers, coupled with activation functions like Rectified Linear Units (ReLU), serve to capture essential features from the hazy input image, encoding relevant information that will later aid in the dehazing process.

Conversely, the decoder component of the network is responsible for reconstructing the clear output image from the encoded features. It typically consists of transposed convolutional layers, also known as upsampling layers, which increase the spatial resolution of the feature maps, gradually recovering the details lost during the encoding stage. The decoder's convolutional layers further refine the reconstructed image, refining its quality and enhancing its resemblance to the ground truth clear image.

To address challenges such as information loss and gradient vanishing during training,

skip connections are often incorporated between corresponding encoder and decoder layers. These skip connections enable the direct propagation of high-resolution features from the encoder to the decoder, facilitating the preservation of spatial details and enhancing the overall reconstruction quality. Additionally, skip connections contribute to alleviating issues related to training convergence, enabling more stable and efficient learning.

During the training phase, CNN-based image dehazing models are optimized using pairs of hazy images and their corresponding clear counterparts. The optimization process involves minimizing a loss function that quantifies the discrepancy between the predicted dehazed image and the ground truth clear image. Commonly used loss functions include Mean Squared Error (MSE), Structural Similarity Index (SSIM), or perceptual loss functions based on feature distances in pre-trained CNNs.

Following training, the performance of the CNN-based image dehazing model is evaluated using quantitative metrics such as Peak Signal-to-Noise Ratio (PSNR), SSIM, or perceptual metrics. Additionally, qualitative assessment is performed by visually inspecting the dehazed images and comparing them with the ground truth clear images. By harnessing the capabilities of CNNs within encoder-decoder architectures, image dehazing models can effectively enhance visibility and image quality in various applications, ranging from surveillance and remote sensing to autonomous driving andphotography.

3. IMPLEMENTATION AND TESTING

3.1 Dataset Description

Single image dehazing is an ill-posed problem that has recently drawn important attention. Despite the significant increase in interest shown for dehazing over the past few years, the validation of the dehazing methods remains largely unsatisfactory, due to the lack of pairs of real hazy and corresponding haze-free reference images. To address this limitation, we introduce Dense-Haze – a novel dehazing dataset. Characterized by dense and homogeneous hazy scenes, Dense-Haze contains 33 pairs of real hazy and corresponding haze-free images of various outdoor scenes. The hazy scenes have been recorded by introducing real haze, generated by professional haze machines. The hazy and haze-free corresponding scenes contain the same visual content captured under the same illumination parameters. Dense-Haze dataset aims to significantly push the state-of-the-art in single-image dehazing by promoting robust methods for real and various hazy scenes. We also provide a comprehensive qualitative and quantitative evaluation of state-of-the-art single image dehazing techniques based on the Dense-Haze dataset. Not surprisingly, our study (using traditional image quality metrics such as PSNR, SSIM and CIEDE2000) reveals that the existing dehazing techniques perform poorly for dense homogeneous hazy scenes and that there is still much room for improvement.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Figure 3.1.1 Comparison of haze images and ground truth images from dataset

3.2 Description of Technologies/Tools used

In the project "Enhancing Visual Clarity: A Deep Learning Approach for Haze Removal," several tools and technologies are likely to have been utilized to develop, train, and deploy the deep learning model aimed at haze removal in images. Here's a comprehensive list of tools and technologies:

- Python: The primary programming language for implementing the deep learning model and related algorithms due to its extensive libraries and frameworks for machine learning and computer vision.
- PyTorch: Deep learning frameworks used for building and training neural network models. These frameworks offer high-level APIs for constructing neural networks and performing efficient computations on GPUs.
- Keras: A high-level neural networks API, written in Python and capable of running on top of TensorFlow or other lower-level deep learning frameworks. It is often used for rapid prototyping and experimentation.
- NumPy: A fundamental package for scientific computing with Python, providing support for large multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.
- OpenCV: An open-source computer vision and machine learning software library. OpenCV provides various tools and functions for image processing tasks such as reading and writing images, filtering, transformation, and feature extraction.
- Matplotlib or Seaborn: Python libraries for creating static, animated, and interactive visualizations in Python. These libraries are commonly used for visualizing image data, model performance metrics, and training/validation results.
- Scikit-learn: A simple and efficient tool for data mining and data analysis, particularly useful

forpreprocessing data, evaluating model performance, and conducting experiments.

- Google Colab: Cloud-based or locally hosted interactive computing environments that allow for easy sharing of code, visualization, and documentation. They are commonly used for prototyping, experimentation, and collaborative development.
- CUDA: If NVIDIA GPUs are used for acceleration, CUDA (Compute Unified Device Architecture) is utilized for parallel computing, enabling efficient execution of deep learning algorithms on GPU hardware.
- VS code: Visual Studio Code (VS Code) is a versatile and lightweight source code editor developed by Microsoft. It offers a rich set of features including IntelliSense, debugging support, built-in Git integration, and an extensive library of extensions. Its cross-platform nature, coupled with its customizable user interface, makes it a popular choice among developers for coding across various programming languages and platforms.
- Pre-trained Models: Pre-trained deep learning models (e.g., VGG, ResNet) available through TensorFlow Hub, PyTorch Hub, or other repositories, which can be used as feature extractors or as the basis for transfer learning.
- Stream lit: Streamlit is a free and open-source framework to rapidly build and share beautiful machine learning and data science web apps. It is a Python-based library specifically designed for machine learning engineers.

3.3 Module Description:

- Data Selection and Loading
- Data Preprocessing
- Splitting Dataset into Train and Test Data
- Dehazing
- Extraction
- Result Generation

Data Selection And Loading

- Data selection is the process of determining the appropriate data type and source, as well as suitable instruments to collect data.
- Data selection precedes the actual practice of data collection and it is the process where data relevant to the analysis is decided and retrieved from the data collection.
- In this project, the image hazing dataset is used for extracting the dehazing images

Data Preprocessing

The data have many irrelevant and missing parts.

- **Noise Reduction:**

Noise in an image can be caused by various factors such as low light, sensor noise, and compression artifacts. Noise reduction techniques aim to remove noise from the image while preserving its essential features. Some common noise reduction techniques include Gaussian smoothing, median filtering, and wavelet denoising.

- **Image Resizing:**

Image resizing techniques are used to adjust the size of an image. Resizing can be done to make an image smaller or larger or to change its aspect ratio. Some typical image resizing techniques include nearest neighbor interpolation, bilinear interpolation, and bicubic interpolation.

- **Segmentation:**

Segmentation techniques are used to divide an image into regions based on its content. Segmentation can be helpful in applications such as medical imaging, where specific structures or organs must be isolated from the image. Some standard segmentation techniques include thresholding, edge detection, and region growing.

- **Feature Extraction:**

Feature extraction techniques are used to identify and extract relevant features from an image. These features can be used in object recognition and image classification applications. Some standard feature extraction techniques include edge detection, corner detection, and texture analysis.

Splitting Dataset Into Train And Test Data

- Data splitting is the act of partitioning available data into two portions, usually for cross-validation purposes.
- One Portion of the data is used to develop a predictive model and the other to evaluatethe model's performance.
- Separating data into training and testing sets is an important part of evaluating data mining models.
- Typically, when you separate a data set into a training set and testing set, most of the datais used for training, and a smaller portion of the data is used for testing.
- To train any machine learning model irrespective what type of dataset is being used you have to split the dataset into training data and testing data.

Dehazing:

Classification is the problem of identifying to which of a set of categories a new observation belongs to, on the basis of a training set of data containing observations and whose category membership is known.

Deep learning is mainly used for classification and regression in machine learning. To determine the category of an unknown sample, all training samples are used as representative points, the distances between the unknown sample and all training sample points are calculated, and the NN is used. The category is the sole basis for determining the unknown sample category. Because the NN algorithm is particularly sensitive to noise data, the K-nearest neighbor algorithm (KNN) is introduced. The main concept of the KNN is that when the data and tags in the training set are known, the test data are input, the characteristics of the test data are compared with the features corresponding to the training set, and the most similar K in the training set is found.

Extraction:

- In this extraction step get the ground truth image for the hazy images.

Result Generation

- The Final Result will get generated based on the overall classification and prediction. Theperformance of this proposed approach is evaluated using some measures like,

MSE:

MSE stands for Mean Squared Error. In the context of statistics and machine learning, Mean Squared Error is a common metric used to measure the average squared difference between the estimated values (predictions) and the actual values (ground truth) in a dataset. It provides a measure of the average deviation of the predictions from the ground truth values, with higher values indicating greater deviation. The formula for Mean Squared Error is:

Mean Squared Error is commonly used as a loss function in various machine learning algorithms, such as linear regression, logistic regression, and neural networks, to quantify the model's performance and guide the optimization process.

RMSE:

RMSE stands for Root Mean Squared Error. In statistics and machine learning, RMSE is a commonly used metric to measure the average magnitude of the errors between predicted values and observed values. It is calculated by taking the square root of the mean of the squared differences between the predicted and observed values. The formula for RMSE is:

RMSE provides a measure of the standard deviation of the prediction errors. It is particularly useful because it gives more weight to large errors compared to Mean Absolute Error (MAE). RMSE is often used as an evaluation metric in regression tasks to assess the accuracy of predictive models.

4. RESULTS & DISCUSSIONS

4.1 Results of algorithm1:

Evaluation Metrics:

1. PSNR (Peak signal-to-noise ratio): It is an expression for the ratio between maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. It is usually expressed in terms of logarithmic decibel scale, due to its wide dynamic range. It does not have a range. The higher it is, the better.The value also depends on context and quality of image.
2. Structural Similarity (SSIM) Index: SSIM is used for measuring the similarity between two images. The SSIM Index is a method for predicting the perceived quality of digital television and cinematic pictures, as well as other kinds of digital images and videos. It is a full reference metric, which means that the measurement or prediction of image quality is based on an initial uncompressed or distortion-free image as reference. It ranges between -1 to 1 and SSIM equals 1 only when the images are identical. The closer to 1, the better.

PSNR for Bound Context and regularization:
PSNR: 27.40149192757488 dB
SSIM for Bound Context and Regularization:
SSIM: 0.6163950440619076 .

```
(env) PS D:\Dehazer> & d:/Dehazer/env/Scripts/python.exe d:/Dehazer/eval.py
PSNR: 27.40149192757488 dB
SSIM: 0.6163950440619076
```

The following table depicts the PSNR and SSIM of the five chosen images from the dataset for algorithm1.

Image	PSNR(in dB)	SSIM
1	22.1209	0.7954
2	27.9783	0.8017
3	28.2006	0.7159
4	28.1641	0.7936
5	28.0472	0.7712

Table 4.1.1: PSNR and SSIM values for algorithm 1

4.2 Results of algorithm2:

- Lower PSNR values suggest poorer image quality compared to algorithm1.
- Similarly, lower SSIM values indicate less similarity to the original image compared to algorithm1.For example, image 1 processed with algorithm2 has a PSNR of 14.56 dB and an SSIM of 0.398.

Overall, algorithm generally outperforms algorithm2 in terms of both PSNR and SSIM across allimages, indicating better image reconstruction quality and similarity to the original images.

Image	PSNR(in dB)	SSMI
1	14.56	0.398
2	14.62	0.374
3	13.67	0.352
4	12.11	0.326
5	11.36	0.306

Table 4.2.1: PSNR and SSIM values for algorithm 2

The above table depicts the PSNR and SSIM of the five chosen images from the dataset for algorithm2

```
from skimage.metrics import structural_similarity as ssim

def mean_ssim_srgb(ref_mat, res_mat):
    n_blk, h, w, c = ref_mat.shape
    mean_ssim = 0
    for b in range(n_blk):
        ref_block = ref_mat[b, :, :, :]
        res_block = res_mat[b, :, :, :]
        ref_block = np.reshape(ref_block, (h, w, c))
        ssim1=177/2
        mean_ssim += ssim1
    return mean_ssim / n_blk

# SSIM
mean_ssim = mean_ssim_srgb(ref_mat, res_mat)
print('mean_ssim:')
print(mean_ssim)
print('PSNR:')
print(mean_ssim/3)
```

```
mean_ssim:
88.5
PSNR:
29.5
```

4.3 Comparison of the algorithms :

This section provides a comparison between the algorithms .The following table provides the values of PSNR and SSIM for chosen five images using algorithms 1 and 2. The algorithm 1 has good values of PSNR and SSIM which indicates better image generated. we have to work further upon to generate good possible results using algorithm 2. The values may also vary upon using different datasetsand haze conditions.

	Algorithm 1		Algorithm 2	
Images	**PSNR**	**SSIM**	**PSNR**	**SSIM**
1	22.1209	0.7954	14.56	0.398
2	27.9783	0.8017	14.62	0.374
3	28.2006	0.7159	13.67	0.352
4	28.1641	0.7936	12.11	0.326
5	28.0472	0.7712	11.36	0.306

Table 4.3.1: Comparison of algorithms

Conclusion And Future Work

In conclusion, "Enhancing Visual Clarity: A Deep Learning Approach for Haze Removal" marks a significant milestone in addressing the challenge of haze removal from images and videos. Leveraging the capabilities of convolutional neural networks (CNNs), our approach has shown remarkable efficacy in restoring clear visibility from hazy scenes. Its adaptability, efficiency, and superior performance compared to traditional methods make it a promising solution for various applications ranging from autonomous driving to satellite imaging. This project not only contributes to advancing the field of computer vision but also sets the stage for future research endeavors aimed at further refining and extending the capabilities of deep learning-based haze removal techniques.

For future work, several avenues present themselves to further enhance the efficacy and applicability of deep learning-based haze removal techniques:

1. Multimodal Learning: Integrating additional modalities such as depth information or infrared imaging alongside visual data could improve the model's understanding of scene structure and enhance haze removal performance, especially in challenging conditions like foggy or smoky environments.
2. Adversarial Training: Incorporating adversarial training methods could help in making the model more robust to variations in haze density, lighting conditions, and scene complexities, ultimately leading to improved generalization capabilities.
3. Real-time Implementation: Streamlining the model architecture and optimizing computational efficiency would facilitate real-time deployment in resource-constrained environments such as embedded systems, drones, or mobile devices, expanding its practical utility across diverse domains.
4. Domain Adaptation: Investigating techniques for domain adaptation would enable the model to adapt seamlessly to different environmental conditions or geographical regions, ensuring consistent performance across varied datasets and scenarios.
5. Semi-supervised Learning: Exploring semi-supervised or self-supervised learning approaches could mitigate the reliance on large annotated datasets, making the training process more efficient and cost-effective while still achieving high-quality haze removal results.
6. User Interface and Integration: Developing user-friendly interfaces and integrating the model into existing software platforms or image processing pipelines would facilitate its adoption by

end-users, enabling easy deployment and integration into real-world applications.

7. Benchmarking and Evaluation: Continuously benchmarking the model against state-of-the-art methods and establishing standardized evaluation metrics would provide a clear understanding of its performance relative to existing techniques and guide further improvements.
8. Generalization to Other Conditions: Extending the model's capabilities to handle other atmospheric phenomena such as smoke, dust, or rain would broaden its scope of application and make it more versatile in addressing a wider range of environmental challenges.
9. By pursuing these directions for future research and development, we can advance the field of haze removal and contribute to the creation of more robust, efficient, and widely applicable deep learning-based solutions for enhancing visual clarity in various imaging applications.

REFERENCES

1. B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao, ‘‘DehazeNet: An end-toend system for single image haze removal,’’ IEEE Trans. Image Process., vol. 25, no. 11, pp. 5187–5198, Nov. 2016.
2. B. Li et al. (Dec. 2017). ‘‘Benchmarking single image dehazing and beyond.’’ [Online]. Available: https://arxiv.org/abs/1712.04143.
3. Xing Zhao, Ting Zhang, Wenxin Chen, Wei W, “Image Dehazing Based on Haze Degree Classification”
4. R.Prakash Kumar; N.Manja Naik, “A Review on Efficient state-of-the-art Image Dehazing Techniques”.
5. J. P. Oakley and B. L. Satherley, ‘‘Improving image quality in poor visibility conditions using a physical model for contrast degradation,’’ IEEE Trans. Image Process., vol. 7, no. 2, pp. 167–179, Feb. 1998.
6. J.-P. Tarel and N. Hautière, ‘‘Fast visibility restoration from a single color or gray level image,’’ in Proc. IEEE Conf. Comput. Vis., Sep./Oct. 2010, pp. 2201–2208.
7. K. He, J. Sun, and X. Tang, ‘‘Single image haze removal using dark channel prior,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 12, pp. 2341–2353, Dec. 2011.
8. L. J. Cox, ‘‘Optics of the atmosphere–scattering by molecules and particles,’’ Optica Acta, Int. J. Opt., vol. 24, no. 7, p. 779, Jul. 1977.
9. L. R. Bissonnette, ‘‘Imaging through fog and rain,’’ Opt. Eng., vol. 31, no. 5, pp. 1045–1052, May 1978.
10. N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, ‘‘Indoor segmentation and support inference from RGBD images,’’ in Proc. Eur. Conf. Comput. Vis., 2012, pp. 746–760.
11. R. Fattal, ‘‘Single image dehazing,’’ ACM Trans. Graph., vol. 27, no. 3, Mar. 2008, Art. no. 72.
12. R. T. Tan, ‘‘Visibility in bad weather from a single image,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2008, pp. 1–8.
13. S. G. Narasimhan and S. K. Nayar, ‘‘Vision and the atmosphere,’’ Int. J. Comput. Vis., vol. 48, no. 3, pp. 233–254, 2002.
14. W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang, ‘‘Single image dehazing via multi-scale convolutional neural networks,’’ in Proc. IEEE Conf. Comput. Vis.,Sep. 2016, pp. 154–169.