Studies on Mobile Object Tracking under Adverse Climate Conditions

THESIS SUBMITTED BY

ASFAK ALI

DOCTOR OF PHILOSOPHY (Engineering)

Department of Electronics and Telecommunication Engineering
Jadavpur University
Kolkata - 700032
West Bengal, India

August, 2025

Studies on Mobile Object Tracking under Adverse Climate Conditions

THESIS SUBMITTED BY

ASFAK ALI

DOCTOR OF PHILOSOPHY (Engineering)

Under the guidance of

Dr. SHELI SINHA CHAUDHURI
Professor, Department of Electronics and Telecommunication Engineering,
Jadavpur University, Kolkata-700032
And
Dr. RAM SARKAR
Professor, Department of Computer Science and Engineering,
Jadavpur University, Kolkata-700032

Department of Electronics and Telecommunication Engineering
Jadavpur University
Kolkata - 700032
West Bengal, India

August, 2025

JADAVPUR UNIVERSITY
KOLKATA-700032, INDIA

Index No. 69/22/E

TITLE OF THE THESIS:

Studies on Mobile Object Tracking under Adverse Climate Conditions

NAME, DESIGNATION & INSTITUTION OF THE SUPERVISORS:

- **Dr. SHELI SINHA CHAUDHURI**
  Professor,
  Department of Electronics and Telecommunication Engineering,
  Jadavpur University, Kolkata-700032

- **Dr. RAM SARKAR**
  Professor,
  Department of Computer Science and Engineering,
  Jadavpur University, Kolkata-700032

LIST OF PUBLICATIONS:

a) JOURNAL:

1. **A. Ali**, A. Ghosh and S. S. Chaudhuri, "**Real-Time Tracking of Moving Objects through Efficient Scale Space Adaptation and Normalized Correlation Filtering**". Signal, Image and Video Processing, https://doi.org/10.1007/s11760-023-02758-x. 2023. **Impact Factor- 2.3**

2. **A. Ali**, A. Ghosh and S. S. Chaudhuri, "**LIDN: A Novel Light Invariant Image Dehazing Network**". Engineering Applications of Artificial Intelligence. https://doi.org/10.1016/j .engappai.2023.106830. 2023, **Impact Factor- 8.0**

3. A. Ghosh, **A. Ali** and S. S. Chaudhuri, "**Novel Parametric Based Time Efficient Portable Real-Time Dehazing System**". Journal of Real-Time Image Processing. https://doi.org/10.1007/s11554-023-01283-x,**Impact Fac- tor- 3.0**

4. Md. S. Akhtar, **A. Ali**, and S. S. Chaudhuri, "**MobileUnetGAN: A single Image Dehazing Model**". Signal, Image and Video Processing. https://doi.org/10.1007/s11760-023-02752-3. , **Impact Factor- 2.3**

5. **A. Ali**, R. Sarkar and S. S. Chaudhuri, "**Wavelet based Auto-Encoder for Simultaneous Haze and Rain Removal from Images**". Pattern Recognition, 2024, https://doi.org/10.1016/j.patcog.2024.110370 **Impact Factor- 8.0**

b) CONFERENCE:

1. **A. Ali**, A. Ghosh, and S. S. Chaudhuri, "**Determination of Optimum Dynamic Threshold for Visual Object Tracker**" 2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI), 2021, pp. 1-5, DOI: 10.1109/ACMI53878.2021.9528215. Rajshahi, Bangladesh.

2. **A. Ali**, and S. S. Chaudhuri, "**A Feature Representation Technique For Angular Margin Loss**".Innovations in Computational Intelligence and Computer Vision. ICICV 2022. Lecture Notes in Networks and Systems, vol 680. Springer, Singapore. https://doi.org/10.1007/978-981-99-2602-2_7.

3. **A. Ali**, Md. S. Akhtar, and S. S. Chaudhuri, "**GAN based Image Dehazing On Raspberry PI**". Comsys 2023. https://doi.org/10.1007/978-981-97-2614-1_45.

c) COPYRIGHT:

1. **A. Ali**, and S. S. Chaudhuri, "**Video-Haze100 Dataset**". Diary No. 18933/2023-CO/L. ROC Number: L-8804/2023. Registered

2. A Ghosh, **A. Ali**, S. Banerjee and S. S. Chaudhuri, "**No Reference Dataset for Daytime and Nighttime Synthetic Hazy Image**". Diary No. 18933/2023-CO/L. ROC Number: L-133885/2023. Registered

3. **A. Ali**, D. Hossain, S. Sk, and S. S. Chaudhuri, "**Video-Rain99 Dataset**". Diary No. 8801/2023-CO/L. ROC Number: L-136249/2023. Registered

d) PATENT:

1. A. Ghosh, **A. Ali**, C. Ghorai, S. S. Chaudhuri "**An Energy-Efficient Portable Device for Dehazing**". Application no: 202431005438. Journal Number- 07/2024 **Published** 2024.

2. **A. Ali**, S. Ganguly, C. Ghorai, R. Sarkar and S. S. Chaudhuri "**Portable Real-Time Haze and Rain Removal Device for Enhanced Video Surveillance**". Application no: 202431046425. Journal Number- 26/2024, **Published** 2024.

LIST OF PRESENTATIONS IN INTERNATIONAL CONFERENCES:

1. "**Determination of Optimum Dynamic Threshold for Visual Object Tracker**" 2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI), 2021 Rajshahi, Bangladesh. Published in 2021.

2. "**A Feature Representation Technique For Angular Margin Loss**".Innovations in Computational Intelligence and Computer Vision. ICICV 2022.

3. "**GAN based Image Dehazing On Raspberry PI**". Comsys 2023.

PROFORMA – 1

Statement of Originality

I **ASFAK ALI** (Index No. D-69/22/E) registered on **28/04/2022** do hereby declare that this thesis entitled - **"Studies on Mobile Object Tracking under Adverse Climate Conditions"** contains literature survey and original research work done by the undersigned candidate as part of Doctoral studies.

All information in this thesis have been obtained and presented in accordance with existing academic rules and ethical conduct. I declare that, as required by these rules and conduct, I havefully cited and referred all materials and results that are not original to this work.

I also declare that I have checked this thesis as per the "Policy on Anti Plagiarism, Jadavpur University, 2019", and the level of similarity as checked by iThenticate software is **5 %**.

Asfak Ali

Signature of Candidate
Date: 01/08/25

S.S. Chaudhuri 1/8/25

Prof. Sheli Sinha Chaudhuri
Supervisor
Department of Electronics and
Telecommunication Engineering
Jadavpur University

Ram Sarkar 1/8/25

Prof. Ram Sarkar
Co-Supervisor
Department of Computer
Science and Engineering
Jadavpur University

PROFORMA – 2

CERTIFICATE FROM THE SUPERVISOR/S

This is to certify that the thesis entitled **"Studies on Mobile Object Tracking under Adverse Climate Conditions"** submitted by **Mr. ASFAK ALI**, who got his/her name registered on **28/04/2022** for the award of Ph. D. (Engg.) degree of Jadavpur University is absolutely based upon his own work under the supervision of **Prof. SHELI SINHA CHAUDHURI** (Dept. of E.T.C.E., JU) and **Prof. RAM SARKAR** (Dept. of C.S.E., JU) and that neither his/her thesis nor any part of the thesis has been submitted for any degree/ diploma or any other academic award anywhere before.

S. S. Chaudhuri 1/8/25

Prof. Sheli Sinha Chaudhuri
Supervisor
Department of Electronics and
Telecommunication Engineering
Jadavpur University

Ram Sarkar 1/8/25

Prof. Ram Sarkar
Co-Supervisor
Department of Computer
Science and Engineering
Jadavpur University

To my parents and family, my constant support,
and to all who inspired me along the way
this work is a tribute to your belief in me.

ACKNOWLEDGEMENTS

I am profoundly grateful to the individuals whose contributions have been instrumental in bringing this thesis to fruition. Foremost, I extend my heartfelt appreciation and admiration to my esteemed supervisors, Prof. Sheli Sinha and Prof. Ram Sarkar, whose unwavering guidance, unwavering support, attentive care, invaluable advice, and consistent oversight have been indispensable throughout this journey.

I extend my sincere gratitude to Prof. Sudhabindu Ray, Head of the Department of Electronics and Telecommunication Engineering at Jadavpur University, and Prof. Debesh Kumar Das for their continuous inspiration, unwavering support, and invaluable suggestions throughout my thesis work.

I am also deeply thankful to Prof. Sayan Chatterjee, Prof. Sudipta Chatterjee, and Prof. Abhijit Chandra, whose insightful contributions as members of my research advisory committee have greatly enriched this research. Additionally, I express my heartfelt appreciation to the faculty and staff of the Department of Electronics and Telecommunication Engineering at Jadavpur University for their steadfast support during my research endeavors.

I extend my sincere thanks to Dr. Subhojit Acharjee, Dr. Chinmoy Ghorai, Dr. Nur Amin Haque, and Dr. Nadim Ahamed for their unwavering support and invaluable suggestions, which have been instrumental in the successful completion of my research. I am equally grateful to my fellow researchers—Mr. Avra Ghosh, Mr. Sayan Tripathi, Mrs. Jhelam Jana, Mr. Sambhab Chaki, and Mrs. Nahida Banu—for their steadfast assistance during the research process. I am deeply grateful for the encouragement and camaraderie of my colleagues at the Computer Vision and Data Analytics Laboratory, including Md. Manarul Sk, Mr. Sohel Akhtar, Miss Moumita Hait, Miss Anisha Paul, Miss Rajani Das, Mr. Dilwar Sk, Mr. Singhan Ganguly, Mr. Bibek Das, and Mr. Saifuddin Sk. I would also like to sincerely thank Miss Sayoni Mandal from the IoT and Embedded ML Laboratory, Department of Electronics and Telecommunication Engineering, for her constant support and help, which meant a lot to me. My heartfelt appreciation extends to Mr. Debam Saha, Dr. Neelotpal Chakraborty, Miss Sanchita Das, Mr. Utathya Aich, and Mr. Diptarka Mandal from the CMATER Laboratory, Department of Computer Science and Engineering, for their valuable assistance and encouragement throughout this journey.

Finally, I owe an immense debt of gratitude to my beloved family and close friends. I am forever thankful to my father, Dr. Ayejar Ali; my mother, Mrs. Fatema Khatun; my sister, Miss Farjana Nasrin; my brother, Tanbir Ali; and my dear friend, Miss Wasifa Nazmin, along with others whose unwavering love, support, encouragement, and inspiration have been the driving force behind the completion of this thesis.

Asfak Ali
Electronics & Telecommunication Engineering
Electronics & Telecommunication Engineering Department
Jadavpur University
Kolkata-32, West Bengal, India

Abstract

Adverse climatic conditions—marked by haze, rain, and fluctuating illumination—pose formidable challenges to real-time mobile object tracking by degrading image quality, introducing noise, and increasing computational complexity. In such environments, traditional tracking algorithms struggle with dynamic object appearance, nonrigid deformations, and unpredictable motion, all of which significantly impair performance.

This thesis presents a comprehensive framework that integrates adaptive tracking with advanced image restoration techniques to overcome these challenges. The tracking module fuses the strengths of the Mean Shift algorithm and the Unscented Kalman Filter through an adaptive search region proposal block. A dynamically computed threshold parameter (β), optimized based on factors such as motion blur, high velocity, and nonlinearity, governs the selection between these methods. In addition, a multi-scale template matching strategy anchored by normalized cross-correlation is employed to enhance target localization accuracy while mitigating computational overhead, with a recursive target model update ensuring improved temporal consistency.

To address visibility impairments due to atmospheric aerosols, the framework incorporates a real-time video dehazing pipeline that leverages a novel haze parameter, SATVAL, to selectively process frames based on their saturation-to-value ratios. Complementing this approach, the Light Invariant Dehazing Network (LIDN) employs a Quadruplet loss-trained architecture to achieve consistent dehazing across diverse lighting conditions. Furthermore, a generative adversarial network (GAN) featuring a modified-MobileNet encoder is developed to balance efficiency with high restoration quality, while a wavelet-based deep autoencoder (WAE) simultaneously mitigates both haze and rain effects by exploiting joint spatial and frequency domain features.

Enhancing the practical relevance of this work, the ExtremeTrack dataset is introduced—a synthetic collection of 199 videos that encapsulate a wide range of weather-induced artifacts. Additionally, the ArcTrack algorithm integrates ArcLoss-based similarity matching with an unsupervised detection mechanism to robustly track objects under occlusion and severe visual degradation.

Collectively, the contributions of this thesis advance the state-of-the-art in mobile object tracking and image restoration, offering robust and efficient solutions for computer vision systems operating under challenging adverse climatic conditions.

Keywords: Object tracking, Adverse Climate Condition, Dehazing, De-Rain.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The realm of computer vision is constantly evolving, with object tracking standing out as one of its most complex and captivating areas of study. Its significance lies in its widespread applications across real-time scenarios such as object behavior analysis [1], autonomous drive systems (ADS) [2], robotics [3], sports analysis [4], and video surveillance [5].

Over the past few decades, significant strides have been made in developing algorithms to address object-tracking challenges. Recent advancements in machine learning and deep learning within the computer vision domain have notably propelled research forward. Among these advancements, Convolutional Neural Network (CNN)-based models[6, 7, 8] have emerged as particularly promising, offering superior generalization capabilities compared to traditional methods. Despite these advancements, object tracking presents a myriad of challenges, including occlusion, nonlinear motion, non-rigid object deformation, high velocity, blurriness, illumination variation, in-plane rotation, scale variation, and background clutter etc, as shown in Figure 1.1. To tackle these challenges, numerous algorithms and

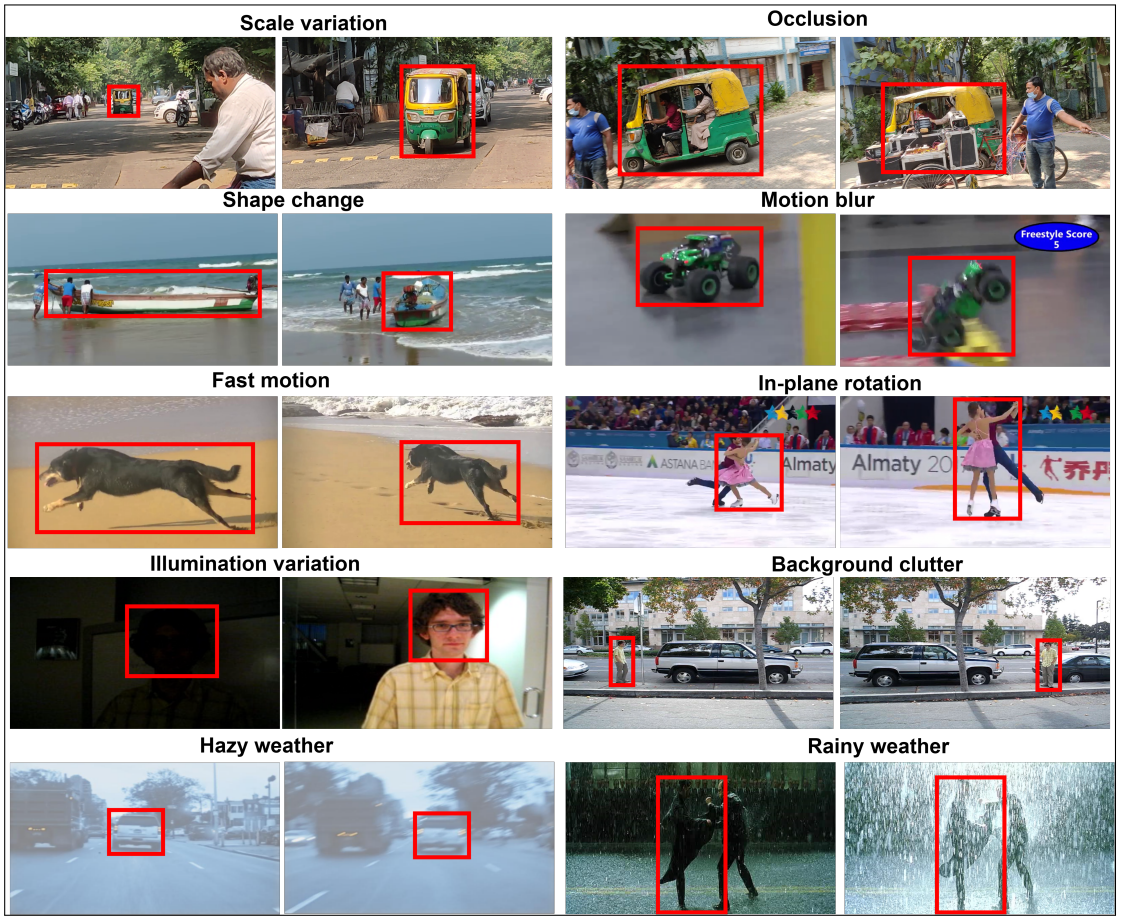

FIGURE 1.1: Different challenging situations.

benchmarks have been devised across various applications, such as object tracking based on unmanned aerial vehicles [9], person tracking [10], large-scale object tracking [11], and video object segmentation [12].

However, a critical gap exists between the performance of existing object-tracking models and their real-world deployment. While these models are typically trained and tested on high-quality videos, real-life scenarios often entail diverse weather conditions like rain and haze [13]. This disconnect between training data and deployment scenarios leads to a domain shift [14], causing many existing models to fall short of achieving the desired objectives of the tracking system, as shown in Figure 1.2. Addressing this domain shift is crucial for enhancing the robustness and effectiveness of object-tracking algorithms in practical applications.

1.1 Research Gaps & Motivation

In real-world applications such as autonomous driving, surveillance, and robotics, object tracking plays a pivotal role in ensuring the effectiveness and safety of systems. However, adverse weather conditions such as fog, rain, snow, and haze present significant challenges for object-tracking algorithms. Despite advancements in object detection under adverse conditions [15, 16, 17], to the best of our knowledge, there are no comprehensive benchmark datasets specifically designed for object tracking in such scenarios. This lack of tailored benchmarks hinders the development and evaluation of robust object-tracking models, highlighting a significant research gap. Object tracking, unlike detection, requires maintaining

FIGURE 1.2: Challenging situation in hazy and rainy weather conditions.

temporal consistency across video frames. This task becomes increasingly challenging in adverse weather, where issues like occlusion, nonlinear motion, and non-rigid object deformation are further exacerbated. Additional factors such as motion blur, illumination variation, scale changes, and background clutter contribute to degraded tracking accuracy. The combined effect of these challenges leads to cluttered, blurry video frames where objects of interest become indistinct, posing a substantial obstacle to current tracking algorithms.

1.1.1 Identified Research Gaps

1. **Lack of Benchmarks for Adverse Weather Object Tracking:** While there are datasets and benchmarks for object detection under adverse weather, no equivalent exists for object tracking. This absence limits the ability to evaluate tracking algorithms effectively.

2. **Domain Shift Between Training and Testing Data:** Existing tracking models trained on clean-weather datasets fail to generalize effectively in adverse weather conditions due to a significant domain shift. This gap underscores the need for domain-adaptive tracking solutions.

3. **Challenges in Image Enhancement for Tracking:** Simultaneous use of image enhancement and tracking models in adverse weather often results in computational overhead and performance degradation. There is a lack of efficient, lightweight solutions for preprocessing weather-affected videos while maintaining high tracking accuracy.

4. **Dependence on Clean Ground Truth (GT) Data for Enhancement Models:** Many enhancement models require clean GT data for training, which is impractical in real-world scenarios. Synthetic data often fails to capture the complexity of real adverse weather, limiting the models' effectiveness.

5. **Insufficient Real-Time and Explainable Tracking Solutions:** Adverse weather scenarios demand real-time processing and explainability, especially in safety-critical applications like autonomous driving. Current approaches often lack the necessary efficiency and transparency.

1.1.2 Motivation for the Research

The absence of tailored benchmarks, domain adaptation challenges, and computational constraints in adverse weather conditions motivate the need for specialized research in object tracking. This research aims to address these gaps by:

- Developing benchmarks specifically designed for object tracking under adverse weather conditions to evaluate and advance tracking algorithms.
- Proposing domain-adaptive object tracking solutions that bridge the performance gap between clean and adverse weather scenarios.
- Designing lightweight image enhancement techniques that seamlessly integrate with tracking pipelines, ensuring real-time performance.
- Exploring self-supervised learning methodologies to eliminate reliance on clean GT data and improve generalization to real-world weather conditions.
- Developing explainable and hardware-optimized tracking architectures capable of operating in safety-critical, real-time applications.

This research has the potential to significantly enhance the robustness and reliability of object tracking systems in adverse weather, contributing to advancements in autonomous systems, surveillance technologies, and beyond. The outcomes of this work aim to bridge the gap between current capabilities and the requirements of real-world applications, thereby establishing a solid foundation for future innovation.

1.2 Scope of the Thesis

With the aforementioned considerations in mind, this thesis approaches the task of object tracking in adverse weather conditions by breaking it down into individual research problems. Each of these problems presents unique challenges, necessitating focused attention on addressing the complexities inherent at different stages of the tracking process. Within this framework, the thesis aims to identify and analyze the diverse challenges associated with object tracking in poor weather conditions, proposing solutions to alleviate these issues. The overarching objective is

to develop and deploy a robust object-tracking algorithm capable of effectively adapting to changes in weather conditions. The thesis is structured around five distinct modules, each addressing specific aspects of the object-tracking process. These modules will be briefly outlined and discussed in the subsequent sections.

1.2.1 Object Tracking Method

The thesis begins with a comprehensive review of existing object-tracking algorithms, setting the stage for the introduction of a novel approach. This innovative method incorporates an adaptive search region proposal block, which collaborates seamlessly with the Mean Shift [18] and Unscented Kalman Filter [19, 20] techniques. This block adeptly navigates the region surrounding the estimated object location, effectively addressing the dynamic changes in appearance and size of moving targets that often pose challenges to tracking algorithms. To tackle these challenges head-on, the thesis introduces the Multi-scale Template Matching technique, which leverages the Normalized Cross-Correlation method within the adaptive search region. This strategic optimization not only reduces computational complexity but also enhances the frame rate, ensuring efficient and accurate object tracking. To validate the efficacy of the proposed algorithm, extensive testing is conducted using the OTB50 dataset [21]. Through rigorous evaluation, the thesis delves into the existing challenges and sheds light on the algorithm's performance in real-world scenarios.

1.2.2 Haze Removal

The adverse weather condition most commonly affecting object tracking performance is haze, particularly in hazy weather. Therefore, it is imperative to develop a haze removal technique to enhance object tracking performance under such conditions. Presently, existing dehazing methods predominantly rely on either daytime or nighttime haze models, which restricts their efficacy in handling haziness across various lighting conditions.

To overcome this limitation, this research introduces the Light Invariant Dehazing Network (LIDN) [22], an end-to-end image dehazing network comprising four key sub-modules: Feature Extractor, Deep Global Atmospheric Light Estimator, Medium Transmission Extractor, and Encoder-Decoder. The proposed model,

trained using Quadruplet loss, demonstrates remarkable effectiveness in reducing artifacts and generating sharper dehazed images. Extensive experiments conducted under diverse lighting conditions validate the superior performance of the proposed LIDN model compared to state-of-the-art daytime and nighttime dehazing approaches. These experiments were conducted using benchmark datasets such as Reside6K [23], further affirming the efficacy and robustness of the proposed approach.

1.2.3 Lightweight Haze Removal Method

The removal of atmospheric haze, known as dehazing, is a crucial aspect of computer vision tasks aimed at enhancing image clarity. Despite significant advancements in dehazing algorithms, which effectively restore contrast-impaired images by eliminating aerosol-induced haze, these methods often fall short when applied to dehazing video sequences in real-time scenarios due to their time-consuming nature.

A novel real-time video dehazing technique is introduced featuring a unique haze parameter called 'SATVAL' [24]. This parameter, derived from the ratio of maximum saturation to the maximum value of an RGB image, is applied within an image scattering model, processing a few video frames per second. Frames with a 'SATVAL' ratio below a predetermined threshold are considered dehazed, while others are passed without dehazing. This approach ensures accurate real-time dehazing of video sequences, rivaling contemporary methods in performance. However, despite its improved processing speed, this method may encounter generalization challenges.

Recently, Generative Adversarial Networks (GANs) [25] have emerged as powerful solutions for dehazing and image restoration tasks. In this chapter, a GAN-based dehazing model is proposed specifically tailored for atmospheric haze removal in images, with a focus on improved generalization. The model integrates generator and discriminator components for adversarial training, enhancing the dehazing process. To capture spatial and contextual information effectively, various architectures such as UNet [26], MANet [27], PSPNet [28], and FPN [29] are explored. Additionally, a Vision Transformer is incorporated as an encoder block to enhance feature extraction.

The proposed technique is evaluated using objective measures such as PSNR and SSIM, trained on the Reside-6K [23] dataset and tested on ITS [23], OTS [23], I-Haze [30], Dense-Haze [31], and NH-Haze [32]. Furthermore, the model is implemented for real-time dehazing on a Raspberry Pi device. The experimental results and comparative studies demonstrate the efficacy of the approach in producing high-quality dehazed images, reaffirming its suitability for practical applications.

1.2.4 Image Restoration

In the previous, the focus was primarily on haze removal from images. However, it's important to note that haze and rain are common weather conditions found in nature. Interestingly, most algorithms in the literature address rain and haze removal separately.

A novel Wavelet-based deep Auto-encoder (WAE) [33] is proposed for simultaneously removing haze and rain effects from images. The proposed network utilizes wavelet transformation and inverse wavelet transformation instead of traditional down-sampling and up-sampling operations, respectively, to enhance network sparsity. By training the model on both spatial and frequency domains, it learns non-stationary features essential for removing haze and rain effects from images.

The proposed model undergoes testing on various datasets containing rain and haze-affected images such as Rain1200[34], RainKITTI2012[35], RainKITTI2015[35], JRSRD[36], Rain14000[37], Rain800[38], RIS[39], RID[39], Reside-6K [23], OTS [23], and ITS [23]. It demonstrates promising performance based on standard evaluation metrics such as structural similarity index measure and peak signal-to-noise ratio. These results confirm the effectiveness of the proposed approach in addressing the challenges posed by rain and haze in images.

1.2.5 Object Tracking in Adverse Weather

In order to establish a comprehensive object-tracking system capable of operating effectively in adverse weather conditions, it is imperative to integrate various individual solutions into a cohesive framework. This research primarily focuses on developing a dataset and an object-tracking algorithm suitable for challenging weather conditions.

The proposed model comprises two main components. Firstly, an image enhancement algorithm is utilized to preprocess weather-degraded video frames. Subsequently, a tracking algorithm is developed to track objects within the enhanced frames. The tracking algorithm incorporates a FastSAM model designed to distinguish between the foreground and background of the target object. Utilizing angular margin loss, the model facilitates feature matching of the target object within each frame.

To evaluate the performance of the proposed model, testing is conducted on a newly curated dataset comprising 200 videos. This dataset includes 100 videos each of rainy and hazy weather conditions, providing a comprehensive test bed for object tracking under adverse weather scenarios.

1.3 Thesis Overview

Each chapter of this thesis is dedicated to a comprehensive exploration of the challenges inherent in object tracking amidst adverse weather conditions, as well as an examination of the currently available datasets and recent standard methods pertaining to this task. Furthermore, proposed solutions are rigorously analyzed and discussed. Except for Chapter 1, which serves as the introduction, the subsequent chapters are organized as follows:

Chapter 2: The following chapter, titled "Foundations of Object Tracking Algorithms" serves as an introduction to various existing methods in object tracking. It delves into novel approaches employing Mean-Shift and Unscented Kalman filter techniques, aiming to address challenges associated with scale variation through multi-scale template matching. The efficacy of these methods is evaluated using the well-established OTB50 dataset.

Chapter 3: This chapter entitled "Enhancing Visual Clarity: Techniques for Image Dehazing" delves into the intricacies of image dehazing, elucidating both the task itself and the array of existing dehazing techniques. Additionally, it introduces a novel deep learning-based haze removal network, meticulously designed to enhance image clarity. To gauge its effectiveness, the proposed network undergoes rigorous evaluation across various benchmarking databases, including the Daytime and Night-time Dehazing Benchmarking Database, Reside6K, I-Haze, O-Haze, Dense-Haze, and NH-Haze Database.

Chapter 4: Entitled "Lightweight Model for Haze Removal" this chapter presents two novel haze removal techniques. The first technique is rooted in traditional methods, while the second employs deep learning for haze removal. These models are meticulously designed to be lightweight, enabling their deployment on edge devices like the Raspberry Pi. This lightweight nature allows them to serve as efficient preprocessing blocks within tracking systems, particularly beneficial for enhancing performance in adverse weather conditions.

Chapter 5: This chapter is titled "A Unified Model for Haze and Rain Removal" which introduces a single-wavelet-based deep learning model for haze and rain removal simultaneously. The model is tested in both real and synthetic haze and rain removal benchmarks, such as Rain14000, Rain800, Rain1200, JRSRD, RIS, RID, Reside-6K, OTS, and ITS.

Chapter 6: This chapter is titled "Object Tracking in Adverse Weather: A Dataset and A Model" wherein a new object tracking dataset in adverse weather conditions is proposed. Additionally, a novel object-tracking model is introduced to track objects under challenging weather conditions and to benchmark the dataset.

Chapter 7: The chapter is titled "Conclusion & Future Directions." which presents a conclusion regarding the contributions made by the proposed datasets and methods toward achieving a compact implementation of object tracking in adverse weather conditions. Additionally, the chapter briefly highlights the limitations of the proposed methods and discusses potential future scopes for further exploration.

Chapter 2

Foundations of Object Tracking Algorithms

2.1 Introduction

Real-time visual object tracking is one of the most important and difficult tasks in computer vision research. It attracts researchers from related domains due to its many applications, like in robotic vision, remote sensing, automated surveillance systems, self-driving vehicles, defense systems, and vision-based control, etc. However, the challenge focused on by researchers is the implementation of the tracking algorithms in real-time due to the dynamic nature of the system, for example, Fast Motion (FM), Nonlinear Motion (NM), Non-Rigid Object Deformation (NOD), full or partial Occlusion (OCC), High Velocity (HV), Motion Blur (MB), Illumination Variation (IV), In-Plane Rotation (IPR), Scale Variation (SV), Low Resolution (LR), Background Cluster (BC), etc. The main tasks of a tracking system are searching and matching, where searching is done by motion tracking models like velocity models[40], Kalman filters[41, 42], Particle filters[20], Optical Flow[43] etc. due to their effectiveness in real-time application. Several matching techniques have been proposed for this objective, like Template matching[44], Convolution Neural network (CNN)[45], Long Short-Term Memory (LSTM)[46], Support Vector Machine (SVM)[47], dictionary-based template matching[48], Mean-shift algorithm[49] etc. In this chapter, an RGB color histogram with feature-based Mean-shift and Unscented Kalman filter (UKF) based robust region proposal block has been introduced. Mean-shift is one of the most efficient techniques against light particle illumination because of its ease of implementation and usefulness regarding computational efficiency. Although Mean-shift algorithms cannot track in scale space, R.T. Collins[50] had proposed Mean-shift blob tracking based on the difference of a Gaussian (DOG), which can track in scale space. However, Mean-shift can not track occluded objects due to its working principle, and blob-based scale space tracking is complex. That is why a combination of UKF[19] and multi-scale template matching technique is introduced in this chapter, as it can track and predict the motion of the moving object. UKF is one of the best state estimators because of its computational efficiency and is also the best for occlusion

A. Ali, A. Ghosh, and S. S. Chaudhuri, "**Determination of Optimum Dynamic Threshold for Visual Object Tracker**" 2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI), 2021, pp. 1-5, DOI: 10.1109/ACMI53878.2021.9528215. Rajshahi, Bangladesh.

A. Ali, A. Ghosh and S. S. Chaudhuri, "**Real-Time Tracking of Moving Objects through Efficient Scale Space Adaptation and Normalized Correlation Filtering**". Signal, Image and Video Processing, https://doi.org/10.1007/s11760-023-02758-x. 2023.

handling and the multi-scale template matching technique is used for scale space tracking.

2.2 Contributions

Considering all the above factors in mind, a novel real-time visual object tracking model is proposed to address the challenges of scale variation and computational complexity. The key contributions of this chapter are as follows:

- The integration of the Mean-Shift and Unscented Kalman filter to estimate the search region and locate the object more efficiently.
- A multi-scale template matching technique improves the accuracy of object tracking by estimating the actual bounding box of the object in the presence of scale variation.
- The utilization of the multi-scale template matching technique within the search region reduces computational complexity and enhances the tracking speed of the model.
- The proposed technique is tested using the OTB50 dataset, and the results obtained are promising.

2.3 Related Work

The object tracking field encompasses various applications, such as security and surveillance systems, which heavily rely on visual data [5, 51]. Other domains benefiting from object tracking include self-driving cars [52], traffic management systems [53], customer behavior analysis [54], interactive game design, and the design of futuristic video effects. Over the years, numerous algorithms have been developed to address the challenges of object tracking, such as Mean-shift [18] and Camshift [55]. However, these algorithms have been found to lack efficient real-time performance. Real-time applications present various scenarios, including FM, NM, MB, OCC, IV, IPR, and SV. These scenarios make it challenging to track objects using a single tracker. To tackle this issue, extensive research has been

conducted on visual tracking algorithms, which have been categorized into five distinct components by Naiyan Wang et al. [56]: Motion Model, Feature Extractor, Observation Model, Model Updater, and Post-processor. Additionally, Yilmaz et al. [57] classified object tracking into three primary types: Point Tracking, Kernel Tracking, and Silhouette Tracking. In point tracking, objects are represented and tracked using their centroid coordinates. The object's location is determined based on its previous state and motion. Point tracking can be classified into two types: the deterministic approach and the probabilistic approach, as proposed by Zhang [5]. Another probabilistic method for single target tracking based on the Bernoulli particle filter was proposed by Bo Li [58].

The kernel tracking method defines one or multiple kernels based on the object's shape, color, and appearance. The object's motion is determined by the kernel's motion, which can be rational, translational, etc. Kernel trackers can be categorized as kernel-based, multi-view-based, and template-based techniques. Silhouette tracking represents an object in a more complex and accurate manner. It can be categorized into two types: contour evolution and shape matching. Feature selection is a crucial task in object tracking as objects are represented by various features in the tracker model. Therefore, a feature extractor is one of the key components of an object tracker. Trackers can be classified into two types based on feature extractors: the generative method and the discriminative method. The generative model calculates the probability density of the object's appearance in the next frame using the extracted features. Common examples of generative models include density estimation [59], Gaussian mixed models [60], dictionary learning [61][62], hidden Markov models [63], and regression networks [64]. The discriminative model does not depend on how the data is generated; it focuses on the error between the object and the background. It involves a binary classification task between the object and the background. In each step, the model learns to differentiate between the object and the background. This type of tracking is also known as "pseudo-tracking" or "tracking-by-detection" because it finds the similarity between the object in the previous frame and the current frame, and detects the object in the current frame. Multiple instance learning, structured output SVM [65], boosting techniques [66], and ensembles of classifiers [67] [68] [69] [70] are common examples of discriminative models. At present, the two most important discriminative models are the correlation model and the deep learning (DL) model. The correlation model utilizes signal processing techniques, specifically convolution between two signals, to determine the presence or absence of

correlation. In the field of computer vision, this method is employed to identify the precise location where correlation occurs. If an image is represented as $f(t)$ and the template is denoted as $h(t)$, the correlation can be expressed as follows:

$$(f \circledast h)(t) = \int_{-\infty}^{+\infty} f(T)h(T+t)dT \tag{2.1}$$

However, as the image pixels are discrete, correlation calculation is carried out as shown below,

$$(f \circledast h)(n) = \sum_{n=-\infty}^{\infty} f(N)h(N+n) \tag{2.2}$$

One of the major challenges in correlation filters is their time complexity, which is $\mathcal{O}(n^2)$. Consequently, the tracker's performance is somewhat slowed down. To address this issue, correlations are computed in the frequency domain using either Fast Fourier Transform (FFT) or Discrete Fourier Transform (DFT) on the images. Following the correlation computation, the images are transformed back to the time domain through Inverse Fast Fourier Transform (IFFT) or Inverse Discrete Fourier Transform (IDFT). This approach enhances computational efficiency, achieving a time complexity of $\mathcal{O}(n \log n)$. Several commonly used correlation models include the Minimum Output Sum of Squared Error (MOSSE) [71], Correlation Filter-based Trackers (CFT) [72], Average of Synthetic Exact Filters (ASEF) [73], Kernel Correlation Filter (KCF) [74], Tracking-Learning-Detection (TLD) [75], P-N learning [70], Collaborative Correlation Tracker (CCT) [76], and Long-term Correlation Tracker (LCT) [77]. These correlation models are applicable for multi-object tracking (MOT). Yang H. et al. [78] proposed a KCF-based multi-object tracker specifically designed for occlusion analysis.

In recent years, DL-based object tracking has gained popularity in the field of computer vision. However, these models require significant computational power, typically provided by a GPU, which limits their portability and makes it challenging to achieve real-time performance. Nevertheless, DL-based trackers are known for their high accuracy due to their excellent decision-making capabilities. There are three main types of DL-based trackers: CNN-based trackers [79, 80], Recurrent Neural Network (RNN)-based trackers [81], and Discriminative DL-based trackers [82, 83].

To enhance the tracker's adaptiveness to different scales, this work proposes a

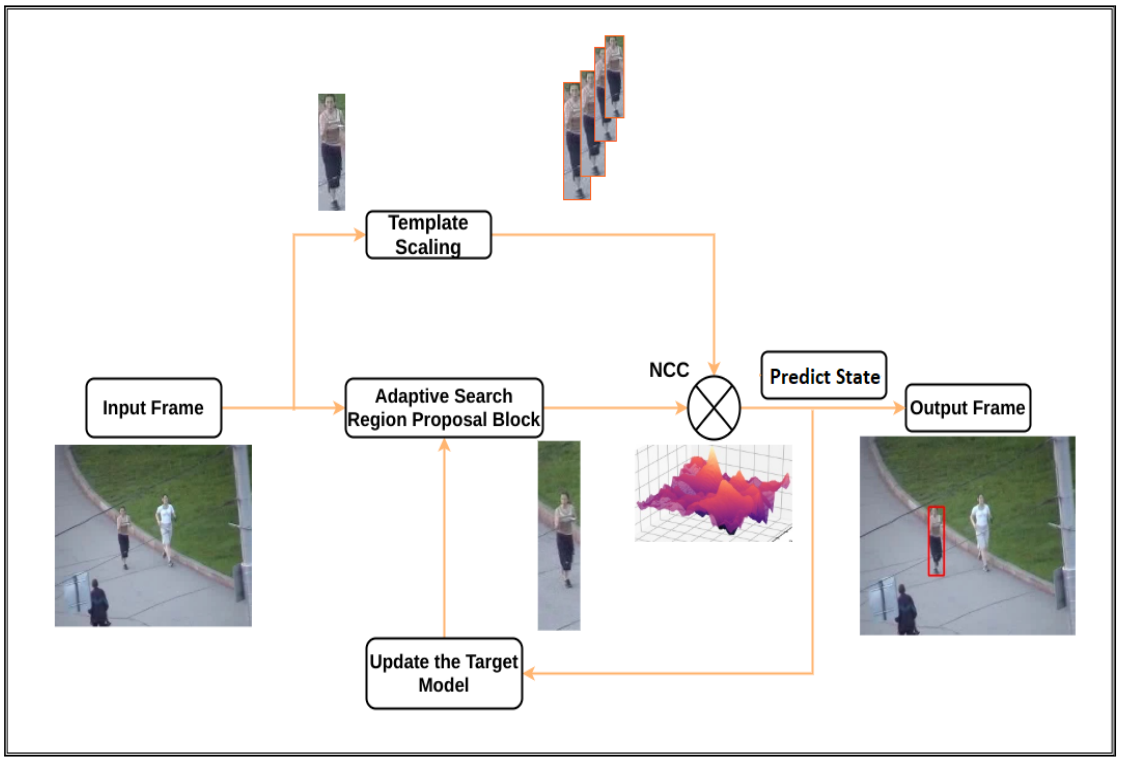

FIGURE 2.1: Flowchart of the proposed tracker. A correlation curve is shown in the figure, where the maximum peak represents the object's location.

generative, point-tracking, kernel-tracking, and correlation-based visual object-tracking system. This tracker utilizes color as a feature and histograms as a feature descriptor to track the object. To evaluate the performance of the proposed tracker, a comparative analysis is conducted against other tracking methods, including KCF[74], BOOSTING[66], MEDIAN-FLOW[84], MIL[85], MOSSE[71], TLD[75], UKF[19], L1-APG2[86], Particle Filter[54], MS[18], GUFIR+KALMAN[87], DCF_{CA}[88], Modified KCF[89], STC[90], MACF[91], EOOT[92], AFAM-PEC[93], Modified STC[94], TNLS-II[95], SiamSSN[96], SiamFC++[97], SNNL[98], TNL2K-2[99], RTTNLD[100], RTTNLD[100], Spikiling SiamFC++[101], VLTTT[102], SiamRPN++[103], JVGT[104]. The results of the comparative analysis demonstrate promising outcomes.

2.4 Proposed Architecture

This section describes the architecture of the proposed tracker. A flowchart of the tracker is shown in Figure 2.1. An overview of the proposed method is described

in algorithm ??. It consists of four parts, namely, model initialization, defining search region, multi-scale template matching, and model update.

2.4.1 Model Initialization

The target is initialized in the first frame by employing manual selection or by a detection algorithm. Then target model is computed by obtaining color histogram, $b(x_i)$ using the following equation,

$$q_u = C \sum_{i=1}^{n} k(||x_i||^2)\delta(b(x_i) - u) \tag{2.3}$$

where C is a normalization constant and k is the kernel. The UKF is initialized by the center location of the object, X_0. An initial sample of the object is stored in a dictionary for scale-space tracking.

2.4.2 Define search region

First, an adaptive search region is predicted using the MS and UKF. An observation model is computed as follows,

$$p_u = C \sum_{i=1}^{n} k(||x_i||^2)\delta(b(x_i) - u) \tag{2.4}$$

and the centroid of the object is estimated using the MS algorithm. An UKF is applied when the object gets occluded. Occlusion is detected by calculating the Bhattacharyya coefficient and the dynamic threshold In practice, trackers have to deal with several scenarios like abrupt motion change, target object change pose, or light conditions, sometimes it will occlude or blur. It is seen that the MS or UKF tracker alone cannot deal with all the conditions. So in the proposed technique, a hybrid model is adopted to offer a better solution. Hence a robust system tracker must have the ability to shift between both trackers simultaneously as per requirement. For that, a dynamic threshold value is needed for selection between the algorithms, which is calculated using the following equation

$$\beta(p(x), q) = 1 - \frac{1}{2} \max(|q_k^u - p_{E(X_k)}^{(u)}|) \tag{2.5}$$

where β is nothing but 1 minus half of the Chebyshev distance between the target model and the color model i.e., the dissimilarity between the target model and color model. Where max(|.|) is Chebyshev distance. Then the Bhattacharyya coefficient $\rho(p(x), q)$ is calculated as

$$\rho(p(x), q) = \sum_{u=1}^{n} \sqrt{p_u(x) q_u} \tag{2.6}$$

thus the value of β is determined and when it is less than the similarity function ρ it means any object with nonlinear motion is detected or an object is occluded. If this condition is satisfied then apply the UKF model else apply MS.

Whenever any object with nonlinear motion or occluded object is detected, the UKF acts as a master tracker. Although the UKF has more time complexity than the MS tracker yet it is best to handle the object with high speed and occluded object. An UKF takes all the previous mean coordinates and predicts the new mean using equations that are described as,

$$\bar{x}_k = \sum_{i=0}^{2N} W_i^m \bar{x}_k^{(i)} \tag{2.7}$$

$$\bar{P}_k = \sum_{i=0}^{2N} W_i^c (\bar{x}_k^{(i)} - \bar{x}_k)(\bar{x}_k^{(i)} - \bar{x}_k)^T + Q_{k-1} \tag{2.8}$$

$$\hat{x}_k = x_k + K_k(y_k - \hat{y}_k) \tag{2.9}$$

$$P_k = \bar{P}_k - K_k P_y K_k^T \tag{2.10}$$

where x_k is the state of the object location or mean of the object at time step k to the measurement function. According to the conventional Kalman prediction algorithm, the predicted value of object location is $\hat{x}_k$. Equation 2.7-Equation 2.8 is called prediction and Equation 2.9-Equation 2.10 is called correction equations of UKF.

On the other hand, the new mean is calculated by mean shift when the object is not occluded or slow motion, defined as

$$z = \frac{\sum_{i=1}^{n} w_i g(||\frac{y - y_i}{h}||^2) y_i}{\sum_{i=1}^{n} w_i g(||\frac{y - y_i}{h}||^2)} - x \tag{2.11}$$

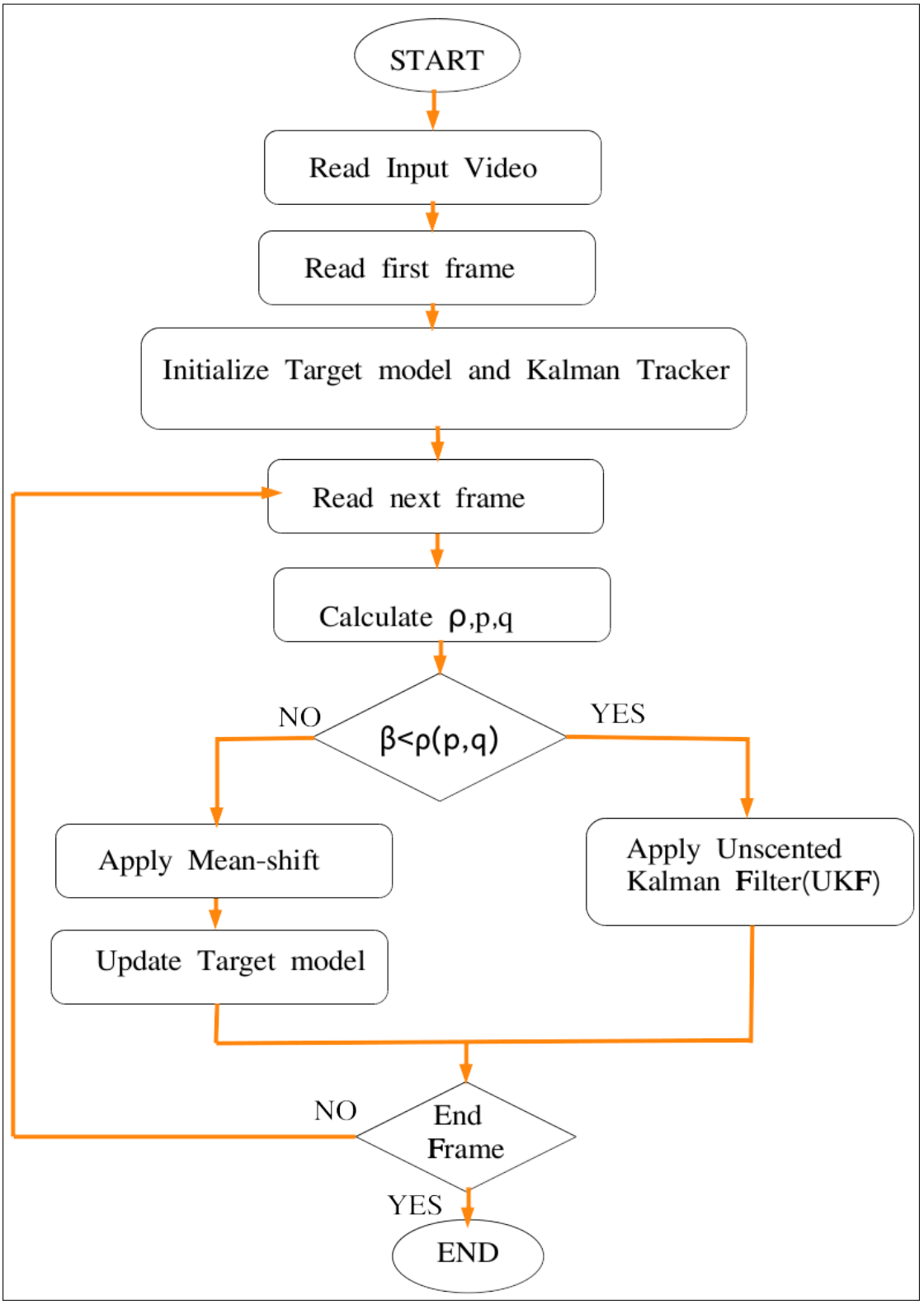

FIGURE 2.2: Flowchart of the proposed adaptive search region block.

where $g(x) = -k'(x)$. $k(x)$is the Epanechikov kernel. After that, the search region is defined as the centroid of the predicted centroid by the mean shift or UKF. The total size of the search region is double the previous object size. This procedure helps the tracker to predict the motion of the object. The flowchart of the adaptive search region block (ASRB) is shown in Figure 2.2.

2.4.3 Multi-scale template matching

Template matching is done by calculating the Normalized Cross Correlation, which is defined as,

$$CCR = \sum_{i=0}^{n-1}\sum_{j=0}^{m-1}[bg(n+i,m+j) - \bar{bg}][t(n+i,m+j) - \bar{t}] \tag{2.12}$$

$$bg\_norm = \sqrt{\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}[bg(n+i,m+j) - \bar{bg}]} \tag{2.13}$$

$$t\_norm = \sqrt{\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}[t(n+i,m+j) - \bar{t}]} \tag{2.14}$$

$$NCC = \frac{CCR}{bg\_norm \times t\_norm} \tag{2.15}$$

where CCR = Cross-correlation, bg_norm = normalization for background, t_norm = normalization for template, bg = background image, t = template. $\bar{bg}$ = mean of the background image intensity. The background of the search region and template are defined in the initial frame. The Correlation calculation has higher complexity in the time domain than the frequency domain. Hence the NCC is calculated in the frequency domain in the proposed model, as follows

$$NCC = \mathcal{F}^{-1}(\frac{BG \times T^*}{|B \times GT^*|}) \tag{2.16}$$

where, BG = $\mathcal{F}(bg)$, T =$\mathcal{F}(t)$, $T*$= Complex Conjugate of T, $\mathcal{F}$ = Fourier Transformation, $\mathcal{F}^-1$= Inverse Fourier Transformation.

After determining the Normalized Cross-Correlation the location of the maximum correlation is found around the position of the true object. Now to solve the scaling problem, the template is being scaled into 15 different scales. For every selected

Input: Object template, Center location of the Object(X_0)
Output: Estimated target location, Estimated size of the object, Updated target model
Initialization:
Target model $q_0(u)$
State of kalman filter X_0
Template of the target
Search Region Estimation:
Calculate Color model $p_k(u)$
Calculate $\rho(p(x), q), \beta(p(x), q)$
if $\beta > \rho$ *then*
Predict next state X_k
else
Compute Target Mean using mean shift
end if
Define search region
Scale Estimation:
$x, y, h, w = TemplateMatching$
Model Update:
Update mean in mean − shift algorithm
Update Kalman Filter
Calculate $q_k(u)$
Update Target model $q_k(u)$

template, NCC is calculated, and the template with the maximum correlation is selected. As per the size of the selected template, the size of the bounding box is estimated. The noise is introduced by different factors in the system during the calculation of NCC. The lemma is presented in this connection. Then the model is updated using Equation 2.19 and Equation 2.20.

Lemma 1: The cross-correlation of a video frame $bg(n)$ and template $t(n)$ from the same video is interdependent on noise $r(n)$.

Proof: Generally, Normalized Cross-correlation is calculated as,

$$NCC = \frac{\mathbb{E}[bg(n)t(n)]}{\sigma_{bg}\sigma_t}$$

Where, $\mathbb{E}$ is the Expectations, σ_{bg} and σ_t are the variance of background and template respectively. The normalization term is ignored for simplicity, which is nothing but cross-correlation, described as,

$$CC = \mathbb{E}[bg(n))(t(n))]$$

Now consider that there is some random noise r(n) present in both the background image and the template then the cross-correlation will change as,

$$CC = \mathbb{E}[(bg(n) + r(n))(t(n) + r(n))]$$
$$CC = \mathbb{E}[bg(n)t(n) + r(n)(bg(n) + t(n)) + r^2(n)]$$
$$CC = \mathbb{E}[bg(n)t(n)] + \mathbb{E}[r(n)(bg(n) + t(n))] + \mathbb{E}[r^2(n)]$$

As the background image bg(n) and template t(n) is independent of the noise r(n), CC can rewritten CC as,

$$CC = \mathbb{E}[bg(n)t(n)] + \mathbb{E}[r(n)]\mathbb{E}[(bg(n) + t(n))] + \mathbb{E}[r^2(n)]$$

In cases where the nature of noise in a system or signal is not known, it may be assumed to be white noise $w(n)$ since the use of white Gaussian noise as a model can offer valuable insights into the behavior of the system or signal. This is true even if the actual noise is not strictly white or Gaussian. The white noise model can be particularly useful for evaluating the impact of noise on the signal-to-noise ratio, measurement accuracy, or the effectiveness of signal processing algorithms.

$$CC = \mathbb{E}[bg(n)t(n)] + \mathbb{E}[w(n)]\mathbb{E}[(bg(n) + t(n))] + \mathbb{E}[w^2(n)]$$

From the properties of white noise it is known that the $\mathbb{E}[w(n)] = 0$, so, the equation is given as,

$$CC = \mathbb{E}[bg(n)t(n)] + \mathbb{E}[w^2(n)]$$

It is also known that $\mathbb{E}[w^2(n)]$ is basically auto-correlation of the white noise, which is given as

$$\mathbb{E}[w^2(n)] = \begin{cases} \sigma_w^2, & \text{if } n = 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.17}$$

Where σ_w^2 is the variance of the white noise. So the CC becomes,

$$CC = \begin{cases} \mathbb{E}[bg(n)t(n)] + \sigma_w^2, & \text{if } n = 0 \\ \mathbb{E}[bg(n)t(n)], & \text{otherwise} \end{cases} \tag{2.18}$$

Here, σ_w^2 can be neglected because it only affects the cross-correlation in the origin of the noise. So cross-correlation is given as,

$$CC = \mathbb{E}[bg(n)t(n)]$$

Thus, the above formulation shows that the cross-correlation of a video frame $bg(n)$ and template $t(n)$ from the same video is independent of noise $r(n)$.

2.4.4 Model update

After finding the object the target model is updated, if an object with linear motion or object is non-occluded, the Mean-shift algorithm works perfectly. However, the target will change due to changes in light illumination, object pose, and changes in viewing angles. So, the target model has to be updated using this equation

$$q_k^{(u)} = (1 - \alpha)q_{k-1}^{(u)} + \alpha p_{E(X_{k-1})}^{(u)} \tag{2.19}$$

where α is a factor on which the color distribution of the current detected object depends $p_{E(X_{k-1})}^{(u)}$ changes the target model at the current frame. This is calculated as

$$\alpha = \frac{1}{n}\sum^{n}|(q_{k-1}^{(u)} - p_{E(X_{k-1})}^{(u)})| \tag{2.20}$$

The target model is not updated when UKF acts as the main tracker in this case, but the target model will update when MS is the master tracker.

2.5 Experiment Results

The experimental results of the proposed method have been shown in Figure 2.3 to Figure 2.6 and in Table 2.2 to Table 2.3. The model has been implemented on Python 3.7.6 and Python OpenCV 4.2.0 library. All experiments were performed on Intel Core i5, 2.3 GHz CPU with 8GB of RAM. Multiple video sequences

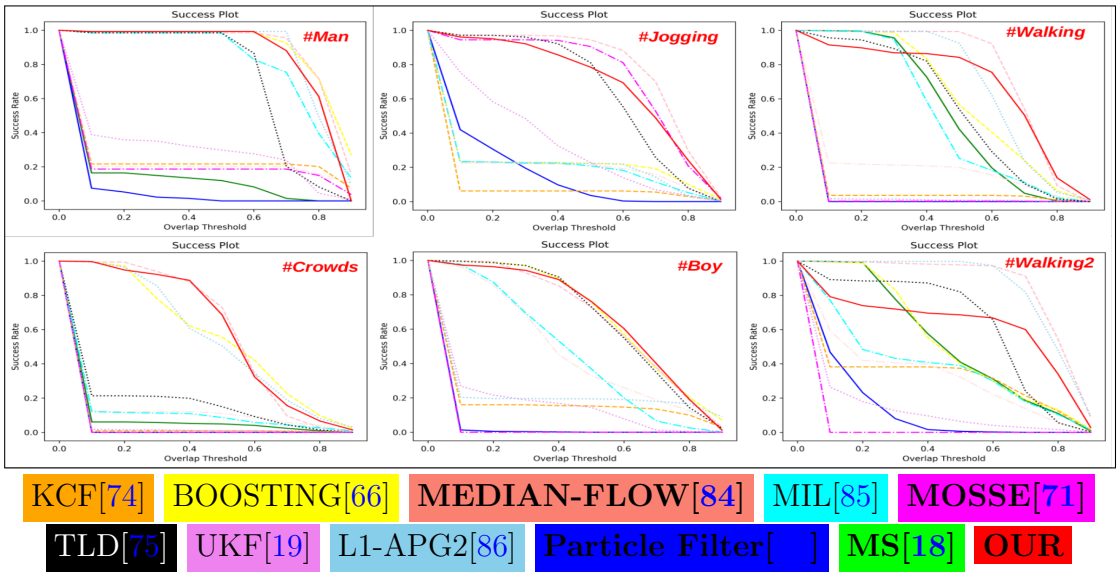

FIGURE 2.3: Success rate comparison at different IOU thresholds of the proposed tracker and different existing trackers on some videos taken from OTB50[21]. The different colors indicate the different trackers as shown above.

with different challenging conditions are tested with the proposed tracker. The proposed method is compared with 30+ existing tracking models like KCF[74], BOOSTING[66], MEDIAN-FLOW[84], MIL[85], MOSSE[71], TLD[75], UKF[19], L1-APG2[86], Particle Filter[54], MS[18], GUFIR+KALMAN[87], DCF_{CA}[88], Modified KCF[89], STC[90], MACF[91], EOOT[92], AFAM-PEC[93], Modified STC[94], TNLS-II[95], SiamSSN[96], SiamFC++[97], SNNL[98], TNL2K-2[99], RTTNLD[100], RTTNLD[100], Spikiling SiamFC++[101], VLTTT[102], SiamRPN++[103], JVGT[104] on OTB50 [21] with different properties. Tracking results are evaluated using some standard qualitative and quantitative methods.

2.5.1 Dataset Description

The evaluation of object tracking algorithms is facilitated by the use of benchmark datasets, known as OTB50[21] datasets. This dataset is comprised of video sequences that have been annotated with ground truth information for object positions. These video sequences present a range of challenges, such as fast motion, occlusion, deformation, and cluttered backgrounds, providing a comprehensive assessment of the performance of object-tracking algorithms in a variety of tracking environments. As a standard benchmark for comparing the performance of different object tracking algorithms, the OTB50 datasets play a crucial role in advancing

the state-of-the-art in the field. To evaluate the proposed model, video sequences were used from this dataset.

2.5.2 Qualitative Comparison

Qualitative comparison of the proposed model is carried out using Precision Plot, Center Location Error, Success Plot, and Tracking Results by Bounding Boxes method in nine challenging conditions: IV, FM, DEF, OCC, MB, LR, SV, BC and IPR. Characteristics of some videos used in this chapter are shown in Table 2.1.

2.5.2.1 Success Plot

Intersection Over Union(IoU) is a method of calculating the percentage overlap area of the actual location of an object and predicted by the tracking model. IoU is given by the equation

$$IOU = \frac{R_t \cup R_g}{R_t \cap R_g} = \frac{TP}{TP + FP + FN} \tag{2.21}$$

where R_t is the area of the target rectangular box, and R_g is the area of the ground truth rectangular box. $\cap$ is known as Intersection, $\cup$ is Union, TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative.

IoU is calculated for every frame with respect to ground truth and predicted bounding box. If the IoU is greater than a certain given threshold then it is called success. By varying the threshold value in the range [0,1], first, calculate the percentage of success then draw the plot threshold vs success rate. The greater the success rate, the better the tracker. Success plots of the proposed model with different video sequences are shown in Figure 2.3. In terms of this success plot, the proposed tracker shows better results in most of the cases.

2.5.2.2 Center location error plot

Center location error is nothing but Euclidean distance between the center of ground truth and the tracking result. Center location error is given by

$$Center\ location\ error = \sqrt{(x_g - x_r)^2 + (y_g - y_r)^2} \tag{2.22}$$

where (x_g, y_g) = coordinate of the centroid of ground truth, (x_r, y_r) = coordinates of the centroid of the tracking system result. The Center location error plot is drawn by the center location error of every frame vs frame number. The Center location error plot is given in Figure 2.4. The figure shows the proposed tracker has less Center location error overall.

2.5.2.3 Precision Plot

Another widely used metric to decide tracker performance is the Precision plot. These plots are helpful to track the performance of the tracker when it is not possible to track the performance of the tracker by simply looking at the center location error plot. The precision plot is calculated using a center location error plot. In any frame, if the center location error is greater than the given threshold value then it is considered a negative frame else it is considered a positive frame. Precision is given by

$$Precision = \frac{Number\ of\ Positive\ of\ Frames}{Total\ number\ of\ Frames} \tag{2.23}$$

A precision plot is drawn after calculating the precision parameter for different threshold values. In this case, the threshold value varies from 0 to 50. A threshold value above 50 is not considered because a higher threshold value results in too much deviation of the center location error, which makes the values useless. The precision plot of different existing tracker models and the proposed model using different video sequences are shown in Figure 2.5. As the area under the curve (AUC) is greater than other trackers, it shows the proposed tracker performs better than other trackers.

2.5.2.4 Tracking result

Object with Motion Blur(MB) and Non-rigid Object Deformation(DEF)

In this chapter, the proposed model is evaluated using a video sequence of a Boy in order to assess its robustness against various challenges, including blur effects and non-rigid object deformation caused by actions in the video sequences named such as Jogging, Jogging2, Walking, Dog, Crowds, and David3. The Mean-Shift algorithm struggled to detect the object when the frames were highly blurred and required a significant amount of time to calculate the location of the target object.

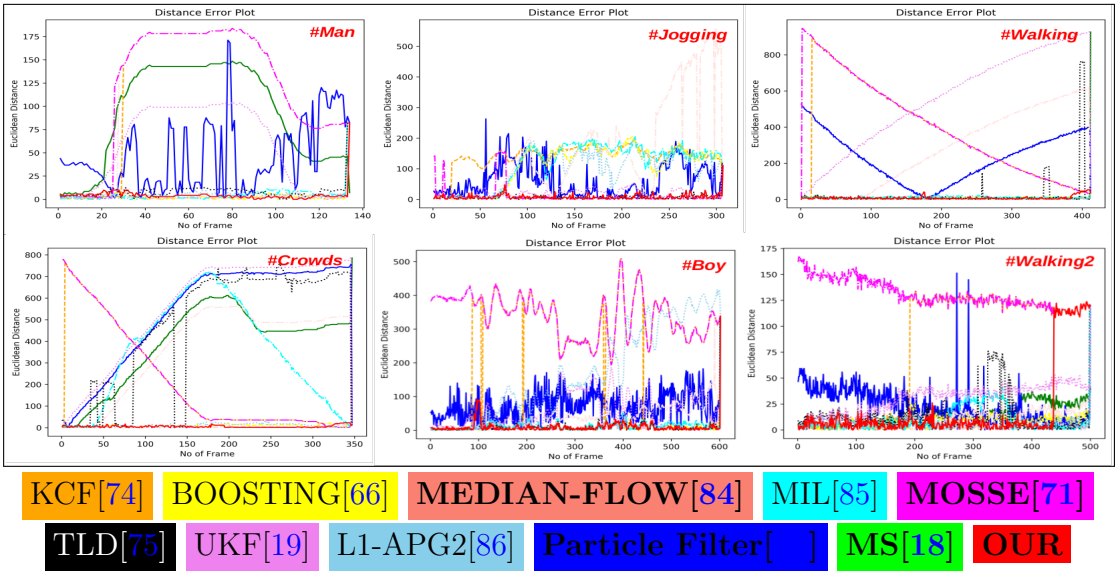

FIGURE 2.4: Center location error (Euclidean distance) comparison of the proposed tracker and different existing trackers on some videos taken from OTB50[21]. The different colors indicate the different trackers as shown above.

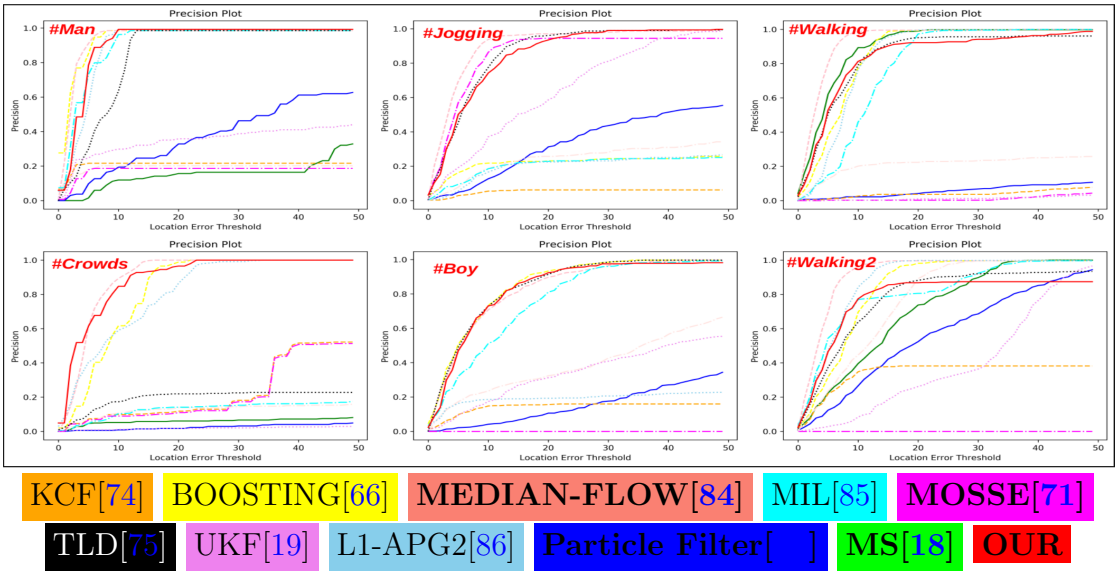

FIGURE 2.5: Precision at different Center location error threshold comparison of the proposed tracker and different existing trackers on some videos taken from OTB50[21]. The different colors indicate the different trackers as shown above.

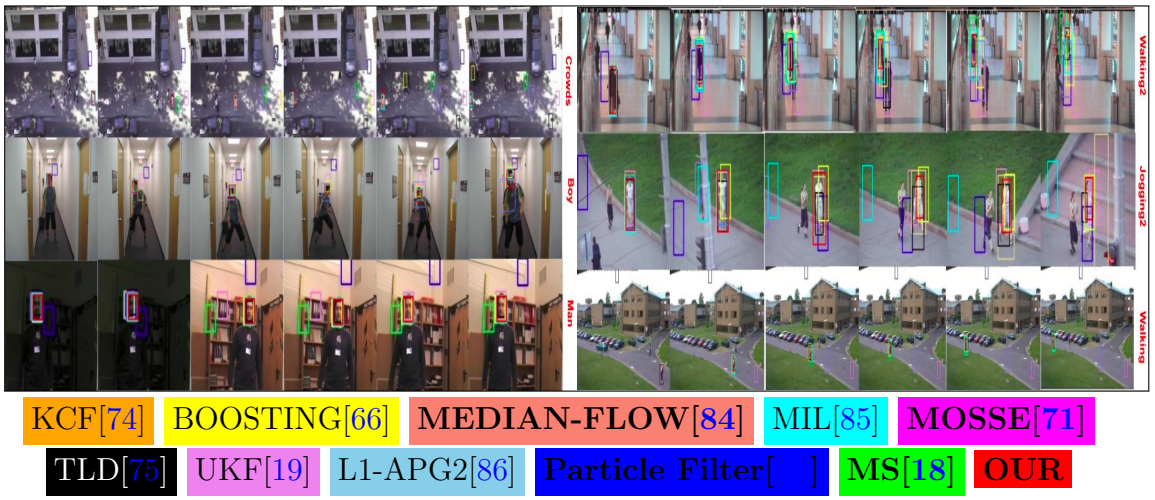

FIGURE 2.6: A tracking result comparison of the proposed tracker and different existing trackers on some videos taken from OTB50[21]. The different colors indicate the different trackers as shown above.

Although the Mean-Shift algorithm performed well for non-rigid object deformation, its performance decreased when the size of the object became small. To overcome this limitation, the multi-scale template matching technique is applied, which improves the results significantly.

Nonlinear Motion and Fast Motion(FB) In order to verify the robustness of the proposed model against the non-linear and fast motion, a video sequence of a boy is used. The performance of the proposed model is compared with state-of-the-art tracking methods. The target in the video had a non-linear motion with high velocity, making it a challenging test case. For instance, at the beginning of the Car24 video sequence, the object moved linearly and at a low velocity, allowing the Mean-Shift algorithm to perform well. However, after the 25th frame, the object increased its speed, causing the Mean-Shift algorithm to lose track of the target. The UKF could handle non-linear motion and high velocity, but it could not track the target perfectly on its own. The proposed method improved tracking efficiency by switching between the MS algorithm and UKF based on the performance of other trackers, as shown in Figure 2.6. The result indicates that the TLD tracker performs best for non-linear motion, followed by the proposed model in this chapter.

Partially or fully occluded Object Tracking(OCC) The efficacy of the proposed model in handling occluded objects is demonstrated using video sequences

like Jogging, David3, Walking, Walking2, and Jogging2. The results showed that the object tracker is able to successfully track the target even when it is fully occluded in the Jogging1 and Jogging2 sequences.

Low Resolution(LR) The proposed model utilizes the color histogram feature, which is optimized for low-resolution video frames. To assess the performance of the tracker in such conditions, the video of Walking2 is used. The results are found to be satisfactory for low-resolution video sequences, as demonstrated in Figure 2.6.

Illumination Variation(IV) One of the most persistent challenges in visual object tracking is variations in illumination, often caused by shadows from trees, buildings, or changes in lighting conditions. To assess the performance of the proposed model under such circumstances, the video sequences of Car24, Man, and Crowds are used. The results of these tracking examples are presented in Figure 2.6. For instance, in the Man video sequence, the model could track the object perfectly even when the lighting conditions changed significantly in the 50th frame. This is in contrast to the Particle filter, which displayed the worst tracking performance in scenarios with large changes in illumination.

TABLE 2.1: Different video sequences and their attributes used in this chapter from the CVLab Visual Tracker Benchmark dataset[21] and average processing time using the proposed method of the different video sequences are presented.

Sequence	*Image size*	*Object Size*	*Challenging conditions*	*Number of frames*	*Average processing time (s/f)*
David3	640 × 480	35 × 131	OCC, BC, DEF, OPR	252	0.0308
Jogging	352 × 288	25 × 101	OCC, DEF, OPR	307	0.0217
Walking	768 × 576	24 × 79	SV, OCC, DEF	412	0.0328
Dog	352 × 240	56 × 48	SV, DEF, OPR	127	0.0201
Dog1	320 × 240	51 × 36	SV, PR, OPR	1350	0.0070
Boy	640 × 480	35 × 42	SV, MB, FM, IPR, OPR	602	0.0380
Man	241 × 193	26 × 39	IV	134	0.0089
Crowds	600 × 480	22 × 51	IV, DEF, BC	347	0.0226
Jogging2	352 × 288	37 × 114	OCC, DEF, OPR	307	0.0248
Twinings	320 × 240	74 × 55	SV, OPR	472	0.0594
Walking2	384 × 288	31 × 115	SV, OCC, LR	500	0.0302
Car24	360 × 240	27 × 24	IV, SV, BC	3059	0.0144

Background Cluster(BC) In some cases, the background can resemble the target objects, creating what are known as background clusters. This can pose a challenge for color histogram-based tracker models. To address this issue, multi-scale template matching is employed. While it does not yield a perfect solution,

it performs well. The performance of the tracker is evaluated using the David3, Car24, and Crowds video sequences, as illustrated in Figure 2.6.

In-plane Rotation(IPR) and Outer-plane Rotation(OPR) The proposed model is evaluated in terms of its ability to handle the rotational movements of target objects in video sequences. Two types of rotations are considered: in-plane rotation, which refers to the rotation of the object within the image plane, and outer-plane rotation, which refers to the rotation of the object outside the image plane. The video sequences of David3, Jogging, Dog, Dog1, Jogging2, Twinnings, and Boy are utilized to demonstrate the robustness of the proposed tracker against these rotational movements. The comparison of the performance of the proposed tracker with other state-of-the-art trackers is presented in Figure 2.6.

Scale variation To evaluate the ability of the proposed tracker in handling scale variations, various video sequences that include changes in object size, such as Dog1, Walking2, Twinings, Boy, Car24, Dog, and Walking, were utilized. The implementation of multi-scale template matching has proven to be effective in addressing this issue. As observed in the Walking2 and Dog video sequences, where the objects move away from the camera and their sizes become smaller, the proposed tracker outperforms other existing trackers, such as MUKF, in terms of maintaining the appropriate size of the bounding box. The results of the comparison between the proposed tracker and other trackers in terms of robustness against scale variation are shown in Figure 2.6.

2.5.3 Quantitative Comparison

2.5.3.1 State-of-the-Art Comparison

The proposed method is compared with nine state-of-the-art tracking models like KCF[74], BOOSTING[66], MEDIAN-FLOW[84], MIL[85], MOSSE[71], TLD[75], UKF[19], L1-APG2[86], Particle Filter[54], GUFIR + KALMAN[87], Modified KCF[89], STC[90], MACF[91], EOOT[92], DCF$_{CA}$[88], AFAM-PEC[93], Modified STC[94], TNLS-II[95], SiamSSN[96], SiamFC++[97], SNNL[98], TNL2K-2[99], RTTNLD[100], RTTNLD[100], Spikiling SiamFC++[101], VLTTT[102], SiamRPN ++[103], JVGT[104]. The three metrics are used, namely, Success Rate, Precision,

and Average Object Tracking Error to compare all the trackers shown in Table 2.2 and Table 2.3 tested on the OTB50 [21]. The best three results are denoted in red, blue, and green in descending order.

2.5.3.2 Precision

Precision (PRE) in object tracking refers to the exactness of the bounding box that surrounds the object. It is calculated by determining the number of frames, where the bounding box precisely encloses the object, divided by the total number of frames in a video sequence. The precision is represented by a Precision plot that is generated based on a specific threshold for location error. A higher Precision value indicates a better performance of the model. The results show that Modified KCF and GUFIR+KALMAN got the third place in terms of Precision. The proposed model achieved second place with 73% precision, and MACF got the first with 76% precision.

2.5.3.3 Success Rate

The success rate (SUCC) of an object tracking algorithm is the proportion of frames in which the item is successfully tracked by the algorithm. The ratio of the number of frames in which the item is tracked to the total number of frames in a video sequence is used to compute it. Equation 12 can be used to calculate success. In this scenario, success is measured using a 0.4 overlap threshold. The success rate of suggested models and other state-of-the-art models are shown in Table 2.2. L1-APG2 and MACF both got third place in terms of Success Rate. The DCF_{CA} came at second, while the proposed model came at first, scoring 68%.

2.5.3.4 Object Tracking Error

The object tracking error (OTE) is the difference between the item's real location and its estimated location in each frame of the video sequence. The average of the Euclidean distance between the actual and estimated item positions for the whole

TABLE 2.2: Success Rate, Precision and Object tracking error comparison of proposed tracker in the CVLab Visual Tracker Benchmark dataset (OTB100)[21]. The three best results are shown in red, blue, and green, respectively. The proposed tracker achieves the best performance compared to the other existing methods.

Tracking	**SUCC**	**PRE**	**OTE**	**Tracking**	**SUCC**	**PRE**	**OTE**
KCF[74]	0.47	0.54	45.10	GUFIR+KALMAN[87]	-	**0.71**	**27.50**
BOOSTING[66]	0.61	0.50	47.30	Modified KCF[89]	0.62	**0.71**	44.54
MEDIAN FLOW[84]	0.42	0.29	146.01	STC[90]	0.59	0.65	273.91
MIL[85]	0.61	0.45	64.10	MACF[91]	**0.65**	**0.76**	46.72
MOOSE[71]	0.55	0.22	193.70	EOOT[92]	-	0.59	32.01
TLD[75]	0.62	0.66	68.50	DCF_{CA}[88]	**0.67**	0.70	56.09
UKF[19]	0.29	0.10	163.00	AFAM-PEC[93]	0.49	0.69	**26.95**
L1-APG2[86]	**0.65**	0.66	52.90	Modified STC[94]	0.55	0.60	63.48
Particale Filter[54]	0.20	0.33	135.82	**PROPOSED**	**0.68**	**0.73**	**24.30**

video sequence is used to compute it, which is

$$OTE = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(x_g^i - x_r^i)^2 + (y_g^i - y_r^i)^2} \tag{2.24}$$

where (x_g^i, y_g^i) = coordinate of the centroid of ground truth for i^{th} frame, (x_r^i, y_r^i) = coordinates of the centroid of the tracking system result for i^{th} frame, N is the number of frame. GUFIR+KALMAN, AFAM-PEC achieved the third, and second positions respectively in terms of object tracking error and the proposed model got the first position with the OTE 24.30 as shown in Table 2.2. A lower average object tracking error number implies that the object tracking algorithm performs better.

2.5.3.5 Comparison with recent Deep Learning Models

In this chapter, a traditional computer vision-based model is proposed for object tracking. However, in this section, the proposed model is compared with DL-based models. Computer vision models, which leverage handcrafted features and explicit rules, offer interpretability and effectiveness in tasks with well-defined patterns. On the other hand, DL models excel at learning complex patterns from abundant data, achieving state-of-the-art performance but often sacrificing interpretability and requiring significant computational resources. The proposed model is designed for real-time applications, achieving a frame rate of 53 FPS on the CPU. In comparison to existing DL models such as VLTTT and SiamRPN++, the model shows

competitive results, outperforming SiamFC++ and Spikiling SiamFC++ in terms of success rate as shown in Table 2.3.

TABLE 2.3: Comparison of existing deep learning models and proposed model using the CVLab Visual Tracker Benchmark dataset(OTB100)[21]. The three best results are shown in red, blue, and green, respectively.

Method	**Year**	**SUCC**	**PRE**
TNLS-II[95]	2017	0.550	0.720
SiamSSN[96]	2020	0.443	0.528
SiamFC++[97]	2020	0.682	**0.884**
SNNL[98]	2021	0.666	0.804
TNL2K-2[99]	2021	0.666	0.804
RTTNLD[100]	2022	0.610	0.790
Spikiling SiamFC++[101]	2022	0.664	0.854
VLTTT[102]	2022	**0.764**	**0.931**
SiamRPN++[103]	2023	**0.696**	**0.914**
JVGT[104]	2023	0.653	0.856
OUR	2023	**0.683**	0.732

2.5.3.6 Computational Complexity

The proposed method is mainly developed using three different efficient algorithms, MS, UKF, and Template matching technique. MS and UKF are used for defining adaptive search regions and Template matching is for localization and scaling purposes. MS has time complexity $\mathcal{O}(w \times h)$, where w is the width of the image and h is the height of the image array of the target object. On the other hand, UKF has time complexity $\mathcal{O}(N^2 + K^{2.376})$, where N is the state matrix and K is the measurement dimension. But in the proposed method dimension of the state and measurement dimension are constant. Template matching technique has time complexity $\mathcal{O}(w \times h)$, but in the frequency domain, it becomes $\mathcal{O}(hlog(h))$, where $h > w$. n_t number of templates are used for multi-scale template matching, so time complexity is $n_t \times \mathcal{O}(hlog(h))$. So total time complexity of the proposed tracker is $\mathcal{O}(w \times h) + \mathcal{O}(N^2 + K^{2.376}) + n_t \times \mathcal{O}(hlog(h))$. As n_t is a constant, as well as $n_t <<< h$ and $n_t <<< w$, the computational complexity of the model is the same as MS, which is $\mathcal{O}(w \times h)$. The proposed method is tested on the different video sequences. The average processing time of the proposed method of some video sequences and their properties is shown in Table 2.1. The proposed method got 53.4fps on the OTB50 dataset.

TABLE 2.4: Comparison of the performance of different blocks used in the proposed method.

Component	SUCC	PRE
MS	0.35	0.26
UKF	0.24	0.10
ASRB	0.47	0.44
SRB + Template Scaling	0.68	0.73

2.5.3.7 Ablation Studies

The proposed method consists of four main components: MS, UKF, Template Scaling, and ASRB. This study presents the effects of tracking the precision and success of each component. Firstly, MS is applied to track the object, utilizing a color model for tracking. While color-based models are highly effective in tracking objects, they tend to fail when objects become occluded. As a result, it can be observed only a 35% success rate and 26% precision, as indicated in Table 2.4. The second component is the UKF, a probabilistic method used for object tracking. However, UKF fails if measurement data is absent in multiple iterations. Consequently, the proposed method achieves only a 24% success rate and 10% precision on the OTB100 dataset. To address the measurement issues in UKF and the occlusion problems in MS, an ASRB is proposed. This block combines MS and UKF using the proposed method, leading to a 47% success rate and 44% precision, as shown in Table 2.4. However, the ASRB is unable to estimate the object's size accurately in the scale space since it focuses on estimating the centroid of the target object. To tackle this issue, the chapter introduces a Template Scaling Block, which overcomes the limitations of other methods, as demonstrated in Table 2.4 and achieves 73% success rate and 68% precision.

2.6 Discussion

This chapter is aimed at designing a reliable real-time object-tracking system that combines the benefits of MS, UKF, and Normalized Cross-Correlation for scale-space tracking. The proposed methods are shows significant improvements in terms of accuracy, tracking error, and precision compared to existing tracking algorithms. The use of multi-scale template matching utilizing Normalized Cross-Correlation further enhances the accuracy of the algorithm. The model is

rigorously evaluated and compared to other tracking systems, and its real-time performance is found to be impressive with a frame rate of 53.4.

It is important to acknowledge that the proposed tracker has some limitations. Firstly, it can handle occlusion in most cases but may not perform as well as MACF in occlusion handling. Secondly, due to the lack of proper datasets, the model is only tested on datasets with good weather conditions, and it is not possible to evaluate the model's performance in challenging weather conditions at this time.

Chapter 3

Enhancing Visual Clarity: Techniques for Image Dehazing

3.1 Introduction

Clear images are essential for computer vision and artificial intelligence systems to function. However, the presence of suspended particles in the atmosphere refracts, reflects, and diffuses light rays affecting image acquisition and loss of image information. Image acquisition becomes affected and image information is often lost to some extent. This results in a loss of brightness, contrast, and visibility of the acquired image leading to a decline in the performance of computer vision applications to which these images act as input. There are primarily two major

FIGURE 3.1: Example of nighttime and daytime images of same scenes. The images show the difference between night and daylight conditions and light scattering variation.

processes to restore such images. First, to employ contrast enhancement-based technique [105],[106], as the main issues caused by haze are low brightness, low contrast, and significant black areas. Consequently, if brightness and contrast can be improved in a suitable ratio, the result can be a clear image. However, it has a drawback; the image's information may be lost if the brightness is increased. Secondly, there are the prior-based techniques [107], [108], [109], in which several priors are proposed by various authors, such as the dark channel prior, the color attenuation prior, etc. Prior-based approaches produce significantly better results than contrast-based enhancement techniques but with higher time complexity as well.

A. Ali, A. Ghosh and S. S. Chaudhuri, "**LIDN: A Novel Light Invariant Image Dehazing Network**". Engineering Applications of Artificial Intelligence. https://doi.org/10.1016/j.engappai.2023.106830. 2023.

There are primarily two types of image dehazing techniques based on the source of light: one is for daylight image dehazing [107, 108, 109], and the other is for nighttime image dehazing[110, 111, 112, 113]. Since the light conditions for day and night are different, the same algorithm often is not suitable for both light conditions. Figure 3.1 illustrates how the airlight varies for the same landscape during the day and at night.

Image dehazing techniques can be classified into two major types depending on the presence or absence of daylight. In the daylight, the sky is totally blue and with high brightness and contrast, whereas, at nighttime, the same sky is pitch dark, with localized light sources showing the difference in the color distribution in the images for the same scenes. There has been limited research conducted on the topic of dehazing for both daytime and nighttime conditions[114].

3.2 Contributions

The main contributions of this chapter are mentioned below:

- A single deep learning-based model providing daytime and nighttime image dehazing techniques. Four sub-modules are introduced for image dehazing in the dehaze network namely Light Invariant Image Dehazing Network (LIDN), such as Feature Extractor, Medium Transmission Extractor (MTEN), Deep Global Atmospheric Light Estimator (DGALE), Encoder-Decoder module. The effectiveness of this proposed dehaze network is demonstrated in various scenarios.
- Reduced inference time than existing models with better qualitative outputs.
- Quadruplet Loss function is introduced to train the LIDN, which gives a sharper dehazed image and reduces the artifacts.

3.3 Related Works

In 1924, H. Koschmieder [115] proposed the following representation of hazy images.

$$I(x) = J(x)t(x) + \alpha(1 - t(x)) \tag{3.1}$$

where, $J(x)$ denotes the haze-free image, $t(x)$ denotes the transmission coefficient, α denotes the atmospheric light and $I(x)$ denotes the hazy image. Analysis of the Equation 3.1 shows there is only one known term, the hazy image $I(x)$, whereas others are unknown terms. Determination of the three parameters, therefore, involves three steps. The calculation of atmospheric light (α) is divided into two parts: dark channel calculation was presented by K. He et al. in 2009[108], followed by atmospheric light calculation. The dark channel prior equation is presented as,

$$I^{dark}(x) = \min_{y \in \Omega(x)} (\min(I^C(y))) \tag{3.2}$$

where $\Omega(x)$ represents a minimum intensity square kernel at x and C symbolizes each color channel in Red, Green, Blue. A dark channel represents the shaded area in the total image and 0.1% of the brightest pixels in the dark channels are represented as Atmospheric Light (α). This model assumed homogeneous haze, with the transmission coefficient represented as -

$$t(x) = e^{-\beta d(x)} \tag{3.3}$$

where β and $t(x)$ are the scattering coefficients of the atmosphere and scene depth respectively. This expression clearly represents the attenuation of haze depending upon distance d. Considering the aerial perspective, the transmission coefficient is modified with w, which is a constant parameter ($0 < w \leq 1$) based on the application used and represented as

$$t(x) = 1 - w * \min_{y \in \Omega(x)} (\min_{C \in r,g,b} \frac{I^c(y)}{\alpha^c}) \tag{3.4}$$

Therefore, once the atmospheric light and transmission coefficient are calculated, the scene is recovered using the following equation

$$J(x) = \frac{(I(x) - \alpha)}{max\{t(x), t_0\}} + \alpha \tag{3.5}$$

where, $J(x)$ is the recovered haze-free scene and restricts the lower limit of the transmission coefficient, which helps to recover the scene properly. Also, t_0 is taken as a lower bound to avoid zero division error in the calculation of the transmission coefficient. The traditional H. Koschmieder equation of image dehazing is adopted to implement the proposed deep learning model called LIDN.

J.P. Oakely et al. [116] presented a method on the image dehazing technique for contrast degradation in the hazy image. Y. Y. Schechner, G Narshimhan, and Shree K. Nayar [117] shared a technique of real-time dehazing of images with the help of polarization. A method was proposed that removes the haze due to polarized atmospheric particles, from images. A contrast restoration of weather-degraded images method was proposed by S.G. Narasimhan and Shree K. Nayar presented [118], which worked in uniform-degraded conditions. S. Shwartz, E. Namer, Y. Y. Schechner [119] presented a paper on blind haze separation, the main idea of which is to estimate and separation of airlight. Though it is an effective algorithm for polarized images, it does not give a good result for the normal image. R. T. Tan [120]proposed a single-image dehazing technique based on a physical model with three parameters. R. Fattal [121] presented a method for a single-image dehazing technique using a transmission function. J.P. Tarel and N. Hautière [122] proposed a method for real-time applications like surveillance, object detection, and tracking using single image dehazing, like atmospheric veil inference, image restoration, and tone mapping. K. He et al [107] introduced the concept of dark channel prior which was based on observations that the intensity of the pixel is minimum in one of the color channels. This prior in combination with the haze imaging model work to estimate the thickness of the haze for recovering the dehazed image. This method also gives the depth map of the input image. However, some issues were not addressed like detecting the sky region pixels and the time complexity of this method.

All of the research work discussed above are classical image dehazing methods. The introduction of deep learning methods enhances the image-dehazing process drastically. For example, in 2016, B. Cai et al. [123] proposed a CNN network based upon a dark channel prior, known as "DehazeNet". However, this method works only for daytime images. In 2018, Patricia L. Suárez et al [124]presented a conditional Generative Adversarial Network(GAN) based method on Single image dehazing. Ancuti et al.[125] proposed a fusion-based methodology based on two original hazy picture inputs, using a white balance and a contrast enhancement mechanism. The proposed method takes as input deteriorated pictures to be restored by adjusting their contrast and then accounting for the color shift. Y. Li et al. [126] designed a unique nighttime haze model that accounts for the glow of various light sources. The model works well at night, but as the image's haziness grows, so does the quantity of noise and artifacts. D. Berman et al. [127] discovered that pixels in a particular cluster are frequently non-local, meaning

that they are distributed throughout the whole picture plane and are placed at varying distances from the camera. They hypothesized that the colors of a haze-free image could be well represented by a few hundred unique colors arranged in compact clusters in RGB space. Z. Li et al. [128] developed a unique technique to picture defog and enhancement based on a replaceable plug-in segmentation module and region-adaptive processing. Their color lines, on the other hand, model the medium transmission, which specifies the proportion of ambient light to scene brightness, whereas color lines represent an object's surface reflection qualities, namely the relationship between illumination chromaticity and diffusion chromaticity.

Based on the Retinex theory, Yu. et al.[129] proposed a nighttime single-image dehazing method. The transmission map is estimated with pixel-wise alpha blending, where the transmissions from dark channel priors and bright channel priors are blended into one transmission map guided by a brightness-aware weight map. Nonetheless, it suffers from the same flaw as most other approaches that rely on atmospheric scattering models. M. Ju. et al.[130] introduced a new parameter named Enhanced Atmospheric Scattering Model (EASM), which can address the dim effect and better model outdoor hazy scenes. They proposed a simple yet effective grey-world assumption-based technique called image Dehazing and Exposure (IDE) to enhance the visibility of hazy images. However, the dehazing method produces inadequate information to assess transmission and also causes color variations.

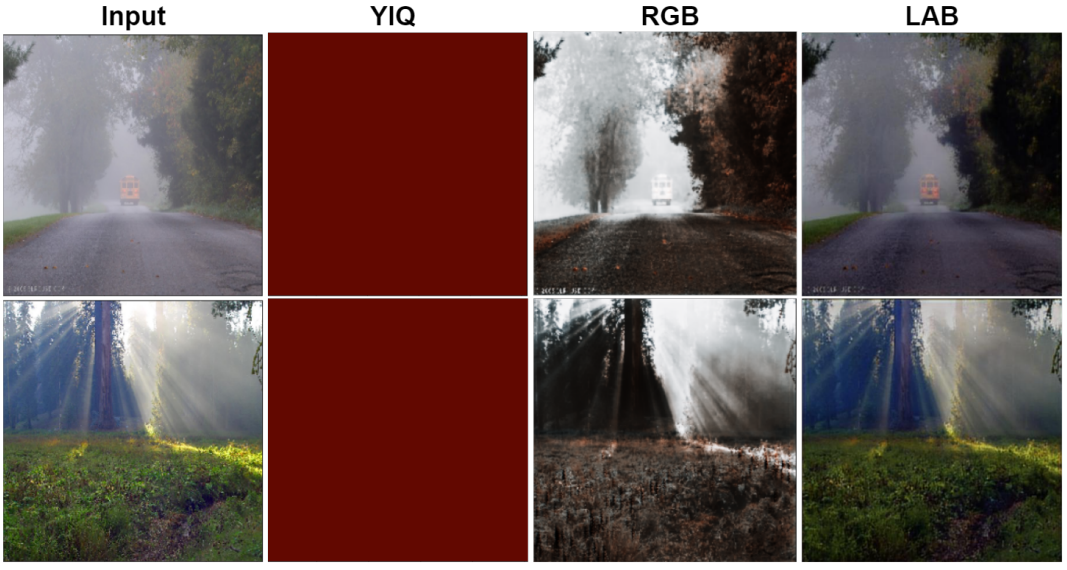

FIGURE 3.2: Output of the proposed model train in YIQ, RGB and LAB color space, where in LAB color space the proposed model shows best results.

3.4 Proposed Method

The proposed model architecture is described in this section. The relationship between the traditional method with the proposed model and the training process is briefly discussed here. The proposed method is based on an end-to-end deep encoder-decoder dehazing architecture. The proposed model takes a hazy image as input and gives a dehaze image as output.

TABLE 3.1: Deep Global Atmospheric Light Estimator Network configurations.

Type	Input Size	Number of Filters	Filter Size	Stride
Convolution + ReLU + Instance Norm +Max Pooling	(224,224,1)	8	(3,3)	1
	(224,224,8)	16	(3,3)	1
	(112,112,16)	32	(3,3)	1
	(56,56,32)	64	(3,3)	1
	(28,28,64)	128	(3,3)	1
	(14,14,128)	256	(3,3)	1
	(7,7,256)	512	(7,7)	1
Flatten	(1,1,512)	-	-	-
Fully connected	512	-	-	-
	256	-	-	-
	128	-	-	-
	1	-	-	-
SoftMax	1	-	-	-

3.4.1 Data Pre-processing

The pre-processing step consists of three parts: in the first part hazy images are transformed from RGB to LAB color space and each channel is normalized, in the second step the RGB image is transferred to the HSV color space and the haze maps $d_1(x)$ and $d_2(x)$ are calculated as follows,

$$d_1(x) = \theta_0 + \theta_1 V(x) + \theta_2 S(x) + p(x|\mu, \sigma^2) \tag{3.6}$$

According to the proposal by Q. Zhu et al. [131], the model is trained using 500 training samples consisting of 120 million scene points. The best learning outcome is obtained with constant values for θ_0, θ_1, and θ_2 are constant. $\theta_0 = 0.121779$, $\theta_1 = 0.959710$, $\theta_2 = 0.780245$, $\mu = 0$ and $\sigma = 0.041337$. It is important to note that any changes in θ will result in corresponding changes in the depth map, leading to artifacts in the dehazed image or a degradation in image quality.

$$d_2(x) = S(x) - V(x) \tag{3.7}$$

where $S(x)$ is the saturation channel and $V(x)$ is the value channel of the input image in HSV color space. In the last step, the model takes the LAB hazy image and the two haze maps as inputs. Ground truth RGB images are scaled into [0.0, 1.0] range for training of the model. It is to be noted that the training of the same network is done using the input images in RGB and YIQ color space, which does not produce good dehazing output as shown in Figure 3.2, because there is a disparity between the LAB model and the RGB model. with the architecture giving the best results employing human response, i.e., LAB model. Diaz-Cely J. et al. [132] showed that models with human-level color perception perform better in LAB color space. So pre-processing is necessary for this network. Figure 3.2 shows the dehazing result on different color spaces, where the LAB color space shows the best performance.

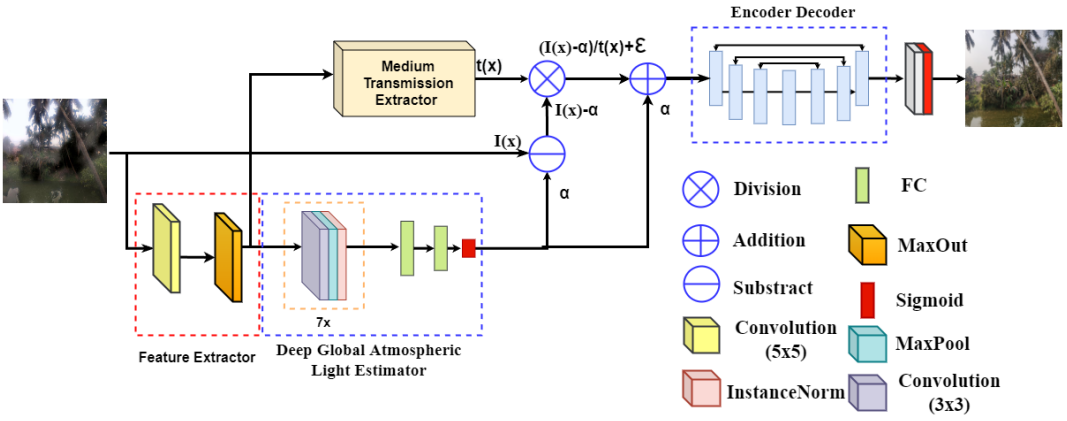

FIGURE 3.3: The overall framework of the proposed LIDN method. The network consists of four sub-blocks Feature Extractor, Medium Transmission Extraction, Deep Global Atmospheric Light Estimator and Encoder Decoder module.

3.4.2 Model Architecture

The proposed model consists of four subsections named Feature Extraction, Medium Transmission Extraction Network, Deep Global Atmospheric Light Estimator and Encoder-Decoder head. The design of the proposed model is shown in Figure 3.3.

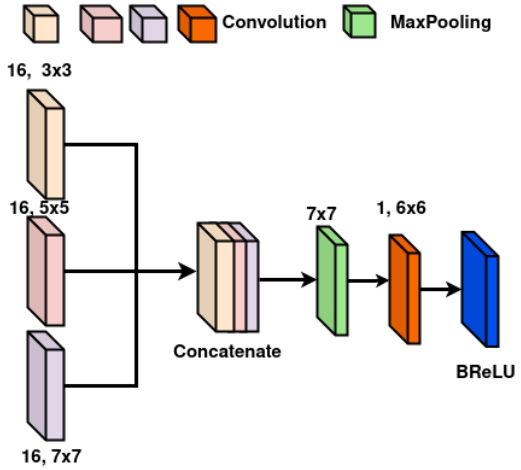

FIGURE 3.4: Schematic diagram of Medium Transmission Extractor. The boxes are denoted by the feature maps.

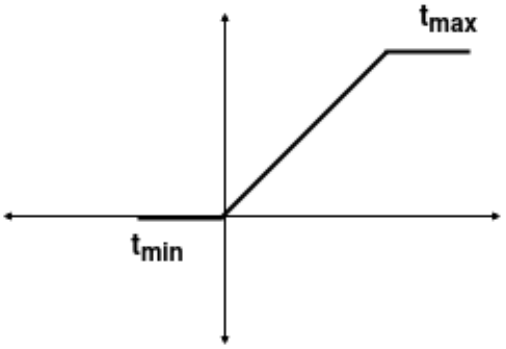

FIGURE 3.5: BReLU Activation function. t_{max}, t_{max} represent the maximum and minimum transmission possible in dehazing, respectively.

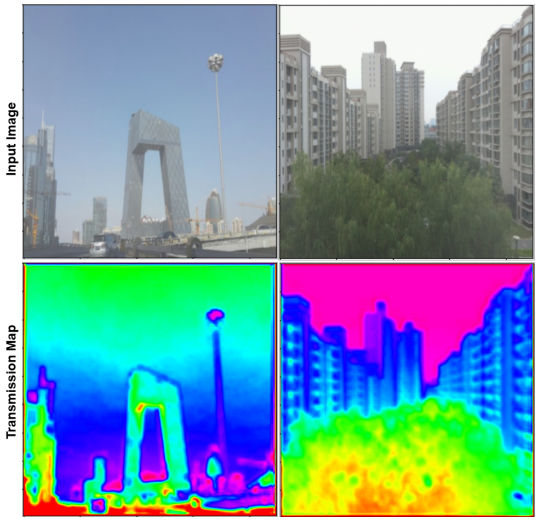

FIGURE 3.6: Extracted transmission-map, $t(x)$ from Medium Transmission Extractor.

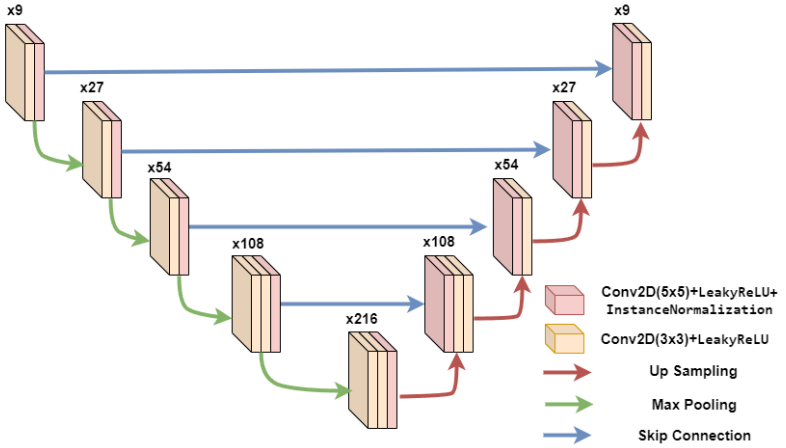

FIGURE 3.7: Schematic diagram of Encoder-Decoder module. The boxes are denoted by the feature maps. On the box's top, it says how many channels are there. The various operations are shown by the arrows.

3.4.2.1 Feature Extraction

The feature extraction layer of the proposed model extracts the haze-related features like dark channel, hue disparity, and color attenuation and takes the pre-processed five-channel input image and passes it through a convolutional layer with 16, (5x5) filters that learnt those haze-related features. Then a MaxOut unit is used for the selection of the most important feature needed for the dehazing purpose. The output of the feature extraction layer is as follows,

$$F^l_{i,j,k} = W^l_{i,j} \circledast I_{i,j,k} \tag{3.8}$$

$$F^l_{i,j,g} = \max_x F^l_{i,j,g,k||g} \tag{3.9}$$

where, $W^l_{i,j}$ is the l^{th} filter, $\circledast$ denotes the convolution operation, and I is the input image. MaxOut unit breaks the last dimension of the extracted features into some value g, which is called the group, and performs the max-reduce operation. Note that k is always divisible by g. "||" denotes the integer division operation. Inspired by [133], the proposed feature extraction layer is developed. The MaxOut unit of the feature extraction layer does the extremum processing in color channels to find the haze-related features[123] by non-linear transformation of features and selecting relevant features using dimension-reduction.

3.4.2.2 Medium Transmission Extraction

To remove the haze from an image using Equation 3.1 two unknown factors must be determined, Medium Transmission $\hat{t}(x)$ and Global Atmospheric Light ($\hat{\alpha}$). To extract the Medium Transmission $\hat{t}(x)$, Medium Transmission Extractor Network (MTEN) is developed as illustrated in Figure 3.4. MTEN takes the output of the feature extractor $F^l_{i,j,g}$ as input and passes it through three parallel convolution layers with different filter sizes, 3×3, 5×5, 7×7, each with 16 number of filters. In this layer, different sizes of the receptive field are created, which can help learn distinct features and depth of the objects given as,

$$ {}^{p}F^t_{i,j,k} = {}^{p}W^t_{i,j,k} \circledast F^l_{i,j,k} \tag{3.10} $$

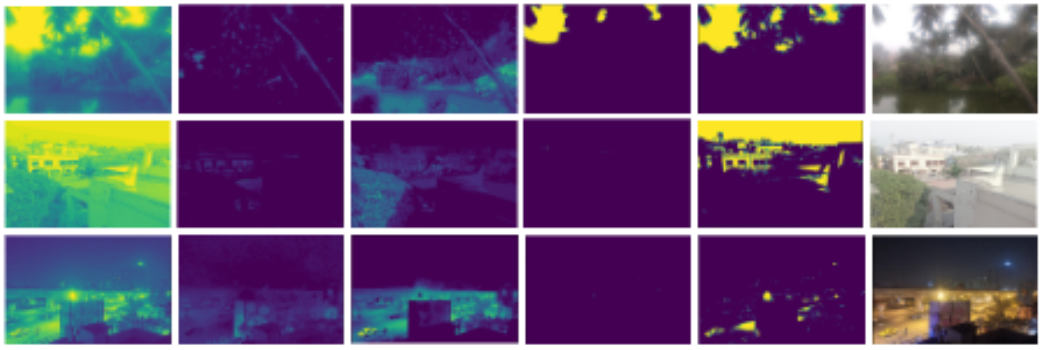

FIGURE 3.8: Extracted haze-related features based on the light condition extracted by Equation 3.15. Right-sided images show the input of the network.

where p represents the three parallel convolution layers and ${}^{p}W^t_{i,j,k}$ is the learn-able weight of the parallel convolution layers. Then the model stacks the feature map using Equation 3.11 and passes the stacks through a 7×7 max-pooling layer with stride 1 and, padding 6. Then a convolution layer is used, which consists of 6×6 filter and stride 1, which is written as,

$$ F^t_{i,j,3k} = [{}^{1}F^t_{i,j,k}, {}^{2}F^t_{i,j,k}, {}^{3}F^t_{i,j,k}] \tag{3.11} $$

$$ F^t_{i,j,3k} = {}^{p}W^t_{i,j,k} \circledast F^t_{i,j,3k} \tag{3.12} $$

Lastly, the Bilateral Rectified Linear Unit (BReLU) activation function is used as the activation on the final layer of MTEN, which is,

$$ \hat{t}(x) = BReLU(F^t_{i,j,3k}) \tag{3.13} $$

BReLU was proposed by Bolun Cai et al.[123] for image dehazing, which is inspired by ReLU and Sigmoid activation. ReLU is mostly useful for regression, however, it has a vanishing gradient problem. On the other hand, Sigmoid does not have this problem but is useful for classification. A plot of BReLU is shown in Figure 3.5. This work uses $t_{max} = 1$ and $t_{min} = 0$, which can be easily implemented using the Sigmoid function followed by ReLU activation. The output of the proposed Medium Transmission Extraction, $\hat{t}(x)$ is shown in Figure 3.6, which depicts that the proposed LIDN model extracts truly good transmission maps.

3.4.2.3 Deep Global Atmospheric Light Estimator

To extract the global atmospheric light, the Deep Global Atmospheric Light Estimator (DGALE) module is introduced in this chapter. Global atmospheric light is obtained using the DGALE in an unsupervised manner. DGALE is designed using 7 deep convolution layers followed by an instance normalization and max pooling layer with 2×2 pixel window, with stride 2. 3×3 filter is used to create a small receptive field in the network. After convolution layer 3 fully connected layers are used to estimate the atmospheric light. As it does not activate all the neurons at the same time, ReLU activation is used in each layer to add non-linearity helping to find the general global atmospheric light. The implementation details of the DGALE module are shown in Table 3.1. Finally, the output layer of DGALE has a Softmax activation. The output of DGALE can be obtained by,

$$\hat{\alpha} = DGALE(F^{l}_{i,j,g}) \tag{3.14}$$

3.4.2.4 Encoder-Decoder Head

Encoder-Decoder is the last layer of the proposed dehazing framework. Before applying the encoder-decoder, the DGALE and MTEN are combined. For this, Equation 3.1 is rewritten as,

$$J(x) = \frac{I(x) - \hat{\alpha}}{\hat{t}(x) + \epsilon} + \hat{\alpha} \tag{3.15}$$

where ϵ is a small constant, which is introduced for eliminating zero division error. On applying Equation 3.15, the combined feature maps are obtained from

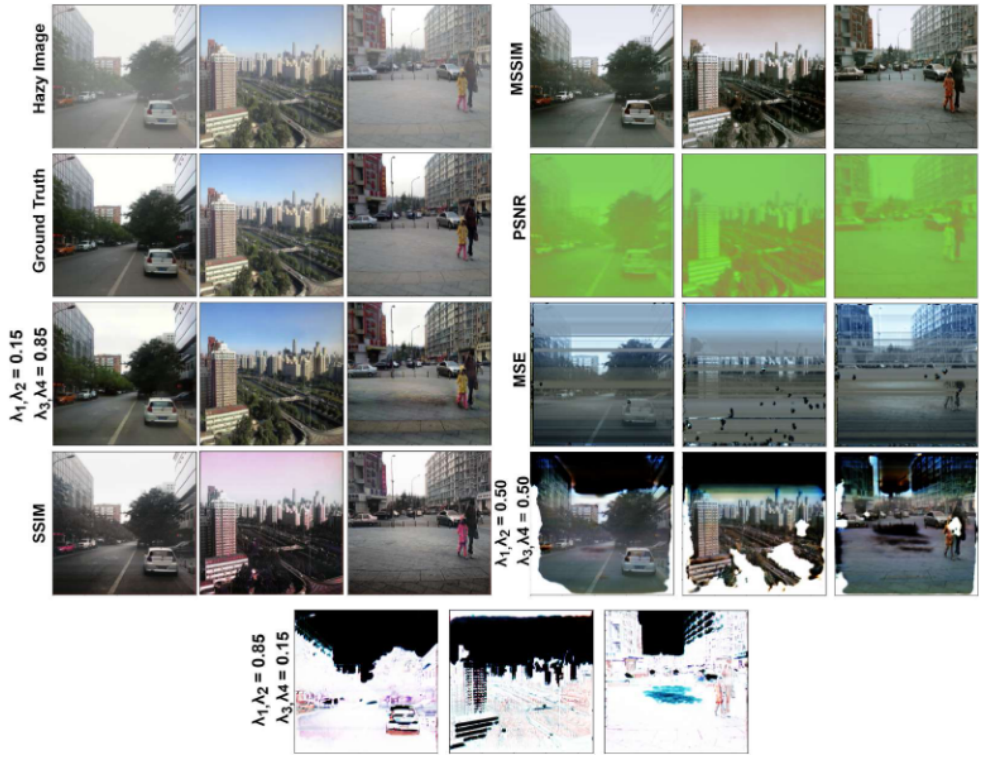

FIGURE 3.9: Dehazing results of the proposed LIDN model trained using MSSIM, PSNR, SSIM, MSE, Quadruplet Loss of different λ values.

the MTEN and DGALE, as shown in Figure 3.8. In Figure 3.8 right most images represent input images and the other five are the feature maps. It is observed that the different feature maps give different features depending on the light conditions, and the haziness of the input. Traditional image dehazing approaches use Equation 3.15 to obtain the final dehazed image, however, the proposed method uses the equation as a feature combiner of the MTEN and DGALE module. The output features are then passed through an Encoder-Decoder module, to obtain the following features,

$$F_{i,j,k}^{EDN} = EDN(J(x)) \tag{3.16}$$

Then the output is estimated using a convolution layer followed by a Sigmoid activation function because the output is scaled in the range [0.0, 1.0], which is shown in Figure 3.3. Output is written as,

$$O(x) = Sigmoid(W_{i,j,k}^{F} \circledast F_{i,j,k}^{EDN}) \tag{3.17}$$

The Deep Global Atmospheric Light Estimator module and the Transmission Coefficient Estimator work together to effectively handle daytime and nighttime conditions. The DGALE module is responsible for estimating the atmospheric light in the scene. The LIDN model is trained to learn and estimate the atmospheric light accurately. To ensure effective learning, the output of DGALE is restricted to a range of [0,1]. By applying a Sigmoid activation function to the output, the model can effectively capture the haze features present in both daytime and nighttime scenes. The Transmission Coefficient Estimator utilizes a 5-channel input LAB image and two haze maps, as described in subsection 3.4.1. This input configuration enables the model to generate good transmission maps, as demonstrated in Figure 3.6. By incorporating these multiple inputs and the haze maps, the proposed model can accurately estimate the transmission coefficients, thereby enhancing the dehazing performance for both daytime and nighttime scenarios. Additionally, the final encoder-decoder block in the proposed model helps improve the output of Equation 3.15.

3.4.3 Training of the Model

This section describes training and testing data, loss function selection, and hyperparameter settings for training LIDN.

3.4.3.1 Training and Testing Data

To train the network, I-HAZE[134], O-HAZE[135], NH-HAZE[136], DENSE-HAZE [137] are used, which contain high resolution hazy and haze-free ground truth. I-HAZE is made up of 35 pairs of interior pictures with and without artificial haze. O-HAZE includes 45 pairs of photographs of actual outdoor haze and haze-free scenes. 55 pairs of non-homogeneous realistic artificial haze and haze-free outdoor images are included in NH-HAZE. 33 pairs of realistic hazy and correspondingly haze-free images of diverse outdoor settings are included in Dense-Haz. N-HAZE, a nighttime dehazing dataset that contains 10 nighttime synthetic haze and haze-free ground-truth image pairs.150 pairs are randomly chosen for training and 28 for validation of the model. Next for data augmentation, high-resolution images are randomly cropped into 448x448 pixels and they are augmented using different augmentation techniques like random-flip, random-rotate, random-crop and

25% random-zoom, to generate 5000 images for training and 1000 for validation. daytime and nighttime dehazing benchmarking database[138] is used for testing the model as it contain real and synthetic images in both day and nighttime hazy and dehaze ground-truth pair, which is useful to test the robustness of the proposed model. The dataset contains 16 images in different light conditions. Also, trained the model on 12,000 randomly selected images from Reside-Full [23] for 25 epochs, which has been tested on SOTS-indoor, OTS-outdoor, OTS-mix, I-HAZE, O-HAZE[135], Dense-HAZE and NH-HAZE.

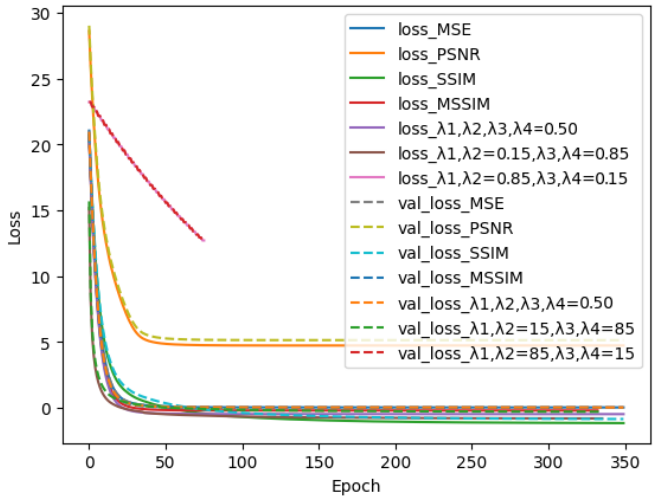

FIGURE 3.10: Training and validation loss curves of the LIDN model trained on LAB, RGB and YIQ color space.

3.4.3.2 Loss Function Selection

The LIDN network has been trained using a combination of four different types of loss functions as follows,

- **Mean Squared Error (MSE)[139]:** Mean squared error is the most used loss function in image dehazing, image reconstruction, etc. It calculates the mean of L2 loss of the channel-wise difference between the target and the predicted image, which allows the consideration of the total distance between the target and the generated image. MSE loss is given as,

$$\mathcal{L}_{MSE} = \frac{1}{N} \sum_{x\epsilon(r,g,b)} (I(x) - O(x))^2 \tag{3.18}$$

where I is the ground-truth image, O is the reconstructed image and x is the image pixel of red, blue, and green channels, N is the total no of images. The gradient of the MSE error is also simple. For each pixel patch, it is given as,

$$\frac{\partial \mathcal{L}_{MSE}}{\partial I(y)} = \frac{-2}{N} \sum_{x \epsilon (r,g,b)} (I(x) - O(x)) \tag{3.19}$$

where x represents a particular pixel and y represents the total pixels in the image. Using MSE as a loss function can introduce a blur or smoothing effect in the reconstructed image.

- **Peak Signal-to-Noise Ratio (PSNR)[140] :** To introduce sharpness to the generated image PSNR loss has been used. It also reduces noise in reconstructed images and improves the signal-to-noise ratio. PSNR is given as,

$$PSNR = 10 \log_{10} (\frac{I_{max}^2}{MSE}) \tag{3.20}$$

In the proposed scenario I_{max} is 1. as, it re-scale the ground truth in the range [0.0, 1.0]. As the PSNR is higher the quality of the reconstructed image is better, so by maximizing the PSNR, and PSNR loss is described as,

$$\mathcal{L}_{PSNR} = -1 \times PSNR \tag{3.21}$$

The gradient of the PSNR loss can be written as,

$$\frac{\partial \mathcal{L}_{PSNR}}{\partial I(y)} = \frac{20 \ln 10}{N \sum_{x \epsilon (r,g,b)} (I(x) - O(x))} \tag{3.22}$$

In [141] Lin Zhang et al. show MSE and PSNR do not accurately reflect the complex features of the human senses and do not correspond well with how people perceive the quality of images (HVS).

- **Structural Similarity Index Measure (SSIM)[142] :** Structural Similarity Index Measure is mainly used for finding the similarity between two images. It can reflect more complex features like textures and boundaries. The SSIM is written as,

$$SSIM = \frac{(2\mu_I \mu_O + C_1)}{(\mu_I^2 + \mu_O^2 + C_1)} . \frac{(2\sigma_{IO} + C_2)}{(\sigma_I^2 + \sigma_O^2 + C_2)} \tag{3.23}$$

$$= l(x).cs(x) \tag{3.24}$$

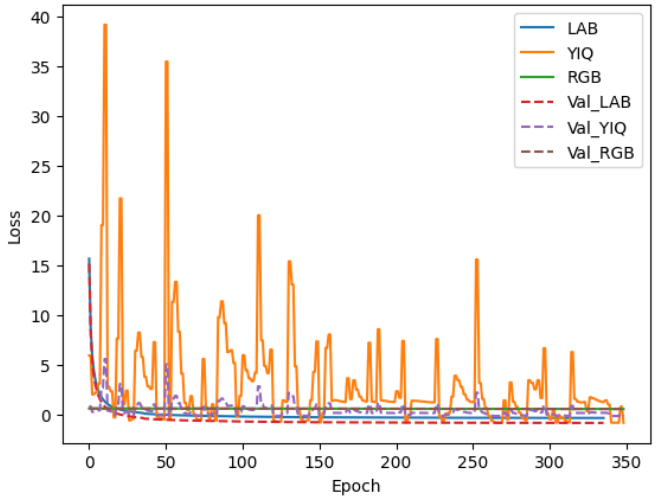

FIGURE 3.11: Training and validation loss curves of the LIDN model trained using MSSIM, PSNR, SSIM, MSE, Quadruplet Loss of different λ values.

where μ_I is the average of I; μ_O is the average of O; σ_I^2 is the variance of I; σ_O^2 is the variance of O; σ_{IO} is the covariance of I and O; $C1$, $C2$ are constant for stabilizing the division. The SSIM loss is defined as,

$$\mathcal{L}_{SSIM} = -1 \times SSIM = -1 \times l(x) \times cs(x) \tag{3.25}$$

Now, the gradient of the SSIM loss is,

$$\frac{\partial \mathcal{L}_{SSIM}}{\partial I(y)} = cs(x)\frac{\partial l(x)}{\partial I(y)} + l(x)\frac{\partial cs(x)}{\partial I(y)} \tag{3.26}$$

where,

$$\frac{\partial l(x)}{\partial I(y)} = 2.G_\sigma(y-x)\frac{(\mu_O - \mu_I l(x))}{(\mu_I^2 + \mu_O^2 + C_1)} \tag{3.27}$$

and,

$$\frac{\partial cs(x)}{\partial I(y)} = \frac{2}{(\sigma_I^2 + \sigma_O^2 + C_2)}.G_\sigma(y-x).[(O(y)-\mu_O) - cs(x)(I(y)-\mu_I)] \tag{3.28}$$

where G_σ is Gaussian kernel of standard deviation σ.

- **Multi-Scale Structural Similarity Index Measure (MSSIM)[143] :** MS-SSIM is a similar type of loss function to SSIM, which reduces the artifacts that are introduced by using other loss. In a dyadic pyramid with M

levels, MS-SSIM is defined as,

$$MS\text{-}SSIM = l_M^{\alpha}(x) \prod_{i=1}^{M} cs_j^{\beta_j}(x) \tag{3.29}$$

where $l(x)$ and $cs(x)$ are same as the SSIM, α and β are constant. In this case, it is 1 and $j = 1, 2, 3, ..., M$. Like SSIM, MS-SSIM can also be maximized for better results. MS-SSIM loss is defined as,

$$\mathcal{L}_{MS\text{-}SSIM} = -1 \times MS\text{-}SSIM \tag{3.30}$$

Therefore, the gradient of the MS-SSIM loss is given by,

$$\frac{\partial \mathcal{L}_{MS\text{-}SSIM}}{\partial I(y)} = (\frac{\partial l_M(x)}{\partial I(y)} + l_M(x).\sum_{i=1}^{M} \frac{1}{cs_i(x)}.\frac{\partial cs_i(x)}{\partial I(y)}).\prod_{i=1}^{M} cs_j(x) \tag{3.31}$$

- **Quadruplet Loss Function:** In this chapter, a combination of $\mathcal{L}_{MSE}$, $\mathcal{L}_{PSNR}$, $\mathcal{L}_{SSIM}$, $\mathcal{L}_{MS\text{-}SSIM}$ are used. $\mathcal{L}_{SSIM}$ helps to learn more complex features, $\mathcal{L}_{MS\text{-}SSIM}$ reduces the artifacts, however, if only these two functions are used for the training, the reconstructed image may look dull. The $\mathcal{L}_{MSE}$ and $\mathcal{L}_{PSNR}$ preserve the color effect and sharpness, and reduce dullness. Total loss is defined as

$$\mathcal{L}_Q = \lambda_1 \mathcal{L}_{MSE} + \lambda_2 \mathcal{L}_{PSNR} + \lambda_3 \mathcal{L}_{SSIM} + \lambda_4 \mathcal{L}_{MS\text{-}SSIM} \tag{3.32}$$

where, λ_1, λ_2, λ_3 and λ_4, are constant. In this chapter λs are set as $\lambda_1 = 0.15$, $\lambda_2 = 0.15$, $\lambda_3 = 0.85$ and $\lambda_4 = 0.85$. When it comes to measuring image similarity, Mean Squared Error (MSE) and Peak Signal-to-noise ratio (PSNR) have been widely used in the past. However, as previously discussed, these metrics fail to capture complex visual features and can be inaccurate in evaluating perceived quality. This is where the Structural Similarity Index (SSIM) and Multi-Scale Structural Similarity Index (MS-SSIM) come into play. SSIM and MS-SSIM take into account human perception of visual quality and incorporate factors such as luminance, contrast, and structural similarity. As a result, they provide more accurate and reliable quality metrics compared to MSE and PSNR. To highlight the importance of SSIM and MS-SSIM, to assign more weight to them in the proposed loss. The model is also trained using different loss functions, SSIM, PSNR, MSE, MSSIM,

and the different weight values of λ_1 to λ_4. The output of the different losses are shown in Figure 3.9. Based on the results, it can be said that SSIM loss and MSSIM loss outperform MSE and PSNR in the proposed network, indicating their superiority in capturing proper color. Assigning a higher weight to SSIM and MSSIM compared to PSNR and MSE would be beneficial. Conversely, training the model with a higher weight for PSNR and MSE, and equal weight for each loss type, resulted in poor results, as depicted in the Figure 3.9. The training and validation loss curves of the LIDN model using different losses are shown in Figure 3.10. Experiments are conducted using three different color spaces, namely LAB, RGB, and YIQ, to assess the performance of the LIDN model during backpropagation. The evaluation involves analyzing the training and validation loss curves, as depicted in Figure 3.11. Observing the loss curves, it becomes apparent that both the RGB and LAB color space models outperform the YIQ color space model in terms of optimization of the model. Specifically, the loss curve of the YIQ color space displays significant fluctuations, while the loss curves of RGB and LAB exhibit smooth and gradual decreases. These results suggest that the LIDN model trained on LAB images achieves superior performance compared to the models trained on RGB and YIQ color spaces. The consistent and smooth reduction in loss within the LAB color space indicates improved convergence and model effectiveness.

TABLE 3.2: Hyper-parameter details for training LIDN

Hyper-parameter	**Value**
Optimizer	Adam $(\beta_1 = 0.9, \beta_2 = 0.99)$
Loss Function	Quadruplet Loss $(\lambda_1, \lambda_2 = 0.15, \lambda_3, \lambda_4 = 0.85)$
Initial Learning Rate	0.01
Learning Rate Decay	0.95
Lower Bound of learning rate	10^{-8}
Patience	7
Cooldown	5
Monitor	Validation Loss
Delta	0.001
Batch Size	1

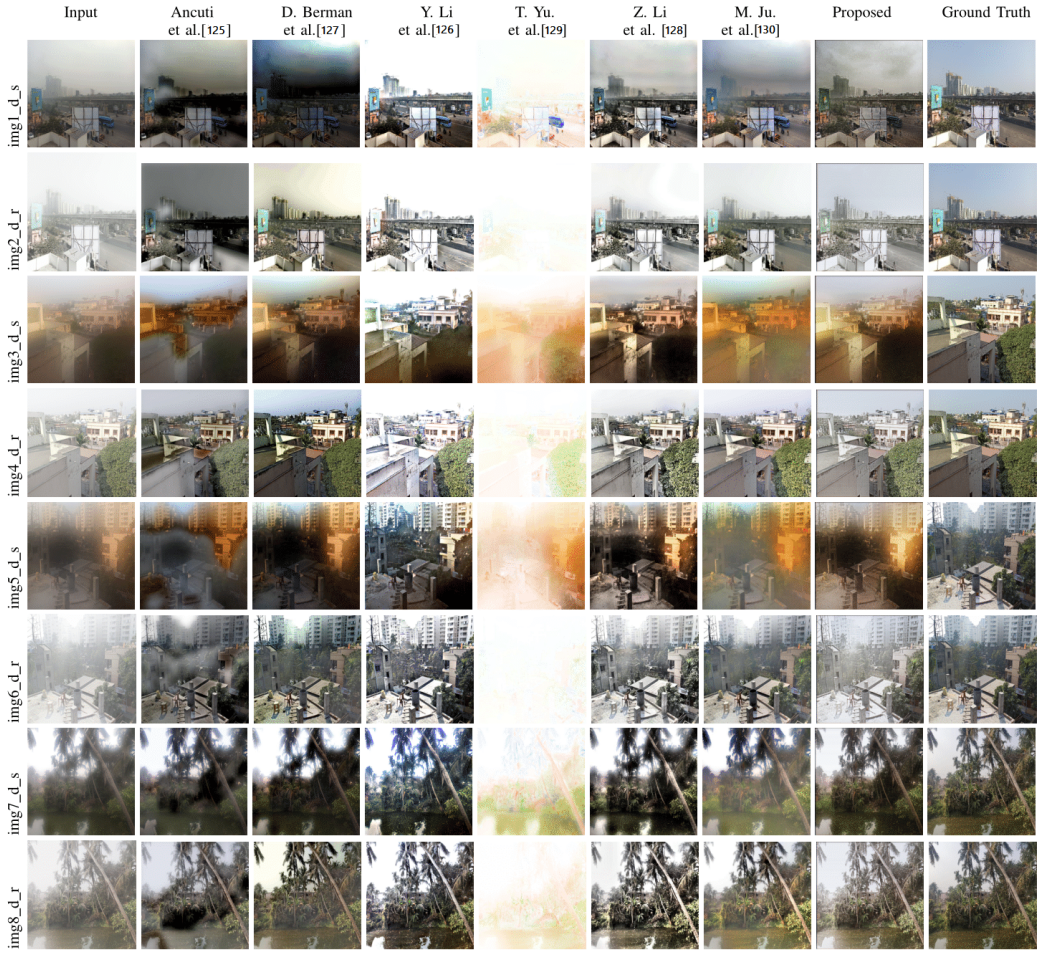

FIGURE 3.12: Qualitative comparison of the proposed method and different existing models in daytime images of daytime and nighttime dehazing benchmarking database.

TABLE 3.3: SSIM, MSSIM, PSNR, MSE comparison of the proposed method and different existing models in **daytime images** from the daytime and nighttime dehazing benchmarking database. The three best results are shown in red, blue, and green, respectively. The proposed model achieves the best performance compared to the other existing methods in most cases.

Method	SSIM	MSSIM	PSNR	MSE
Proposed model	**0.413**	**0.585**	**13.583**	**11.382**
Ancuti et al.[125]	**0.415**	0.559	**14.413**	**10.575**
D. Berman et al.[127]	**0.392**	**0.581**	**13.797**	13.942
M. Ju. et al.[130]	0.374	**0.582**	11.621	18.133
Y. Li et al. [126]	0.355	0.551	13.215	12.650
Z. Li et al. [128]	0.389	0.553	13.244	**12.420**
T. Yu.et al. [129]	0.237	0.324	5.290	86.183

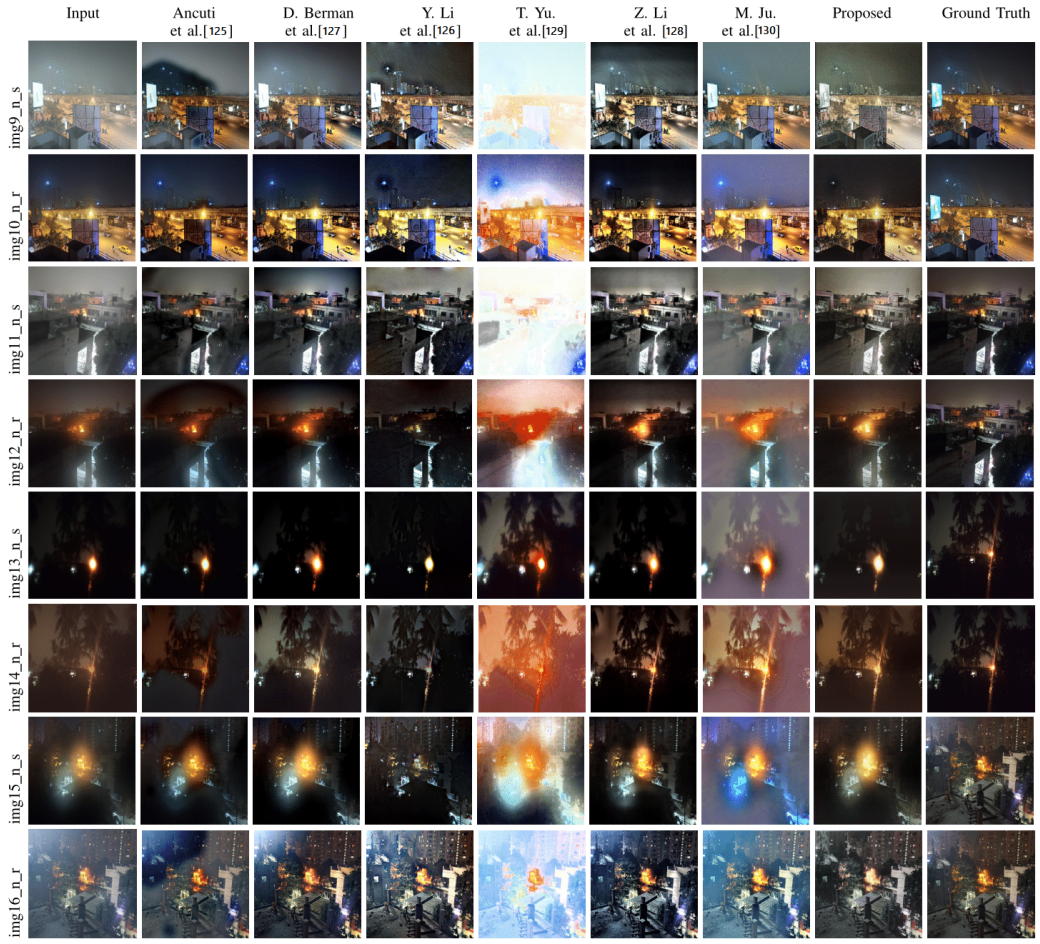

FIGURE 3.13: Qualitative comparison of the proposed method and different existing model in nighttime images of daytime and nighttime dehazing benchmarking database.

TABLE 3.4: Comparison of SSIM, MSSIM, PSNR, MSE parameters of the proposed method and different existing models in **nighttime images** from the daytime and nighttime dehazing benchmarking database. The three best results are shown in red, blue, and green, respectively. The proposed model shows the best performance compared to the other existing methods in most cases.

Method	SSIM	MSSIM	PSNR	MSE
Proposed model	**0.393**	**0.500**	**14.222**	**11.094**
Ancuti et al.[125]	**0.375**	**0.468**	**14.083**	**12.635**
D. Berman et al.[127]	0.322	**0.488**	13.769	**13.080**
M. Ju. et al.[130]	0.304	**0.468**	9.177	20.409
Y. Li et al. [126]	**0.346**	0.450	**14.505**	16.992
Z. Li et al. [128]	0.299	0.466	13.620	13.553
T. Yu.et al. [129]	0.299	0.369	7.889	52.875

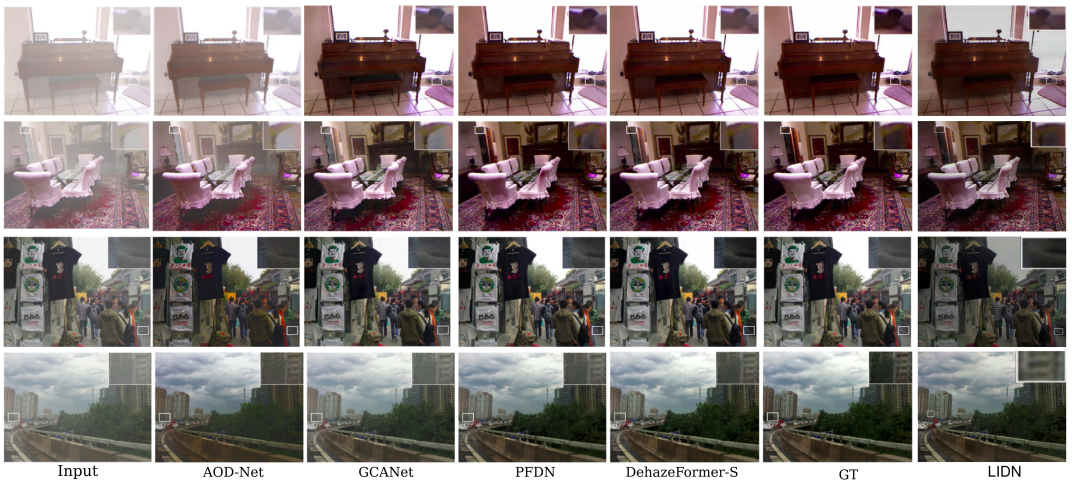

FIGURE 3.14: Qualitative comparison of image dehazing methods on SOTS-mix dataset, where the first two rows are indoor images, and the last two rows are the outdoor images. The first column is the hazy images, the second last column is the corresponding ground truth and the last column is the corresponding result.

TABLE 3.5: Comparison of SSIM, MSSIM, PSNR, MSE parameters of the proposed method and different existing models in **real hazy images** from the daytime and nighttime dehazing benchmarking database. The three best results are shown in red, blue, and green, respectively. The proposed model shows the best performance compared to the other existing methods in most cases.

Method	SSIM	MSSIM	PSNR	MSE
Proposed model	**0.409**	**0.547**	**14.127**	**10.790**
Ancuti et al.[125]	**0.402**	0.526	**13.777**	**12.28**
D. Berman et al.[127]	0.326	0.526	**13.571**	14.18
M. Ju. et al.[130]	0.353	**0.535**	11.710	18.81
Y. Li et al. [126]	**0.355**	0.517	12.495	16.66
Z. Li et al. [128]	0.336	**0.528**	13.219	**13.39**
T. Yu.et al. [129]	0.315	0.354	7.382	58.09

3.4.3.3 Hyper-parameter Tuning and Training Setting

Tensorflow 2.4.2 is used to implement the proposed model, and a Colab notebook equipped with an NVIDIA Tesla T4 15GB GPU and 13GB RAM is used for training. With exponential decay rates of 0.9 and 0.999 for β_1 and β_2, respectively, the models are trained using the Adam optimizer. The batch size and the initial learning rate are set to 1 and 0.01 respectively at the beginning. The model is trained in such a way that if the validation loss does not decrease less than 0.001, for more than 7 epochs (Patient), then the learning rate will be reduced by the factor of 0.95 i.e., the new learning rate, $lr_i = factor \times lr_{i-1}$. The lower

TABLE 3.6: Comparison of SSIM, MSSIM, PSNR, MSE parameters of the proposed method and different existing models in **synthetic hazy images** from the daytime and nighttime dehazing benchmarking database. The three best results are shown in red, blue, and green, respectively. The proposed model shows the best performance compared to the other existing methods in most cases.

Method	SSIM	MSSIM	PSNR	MSE
Proposed model	**0.397**	**0.537**	13.594	**11.80**
Ancuti et al.[125]	**0.387**	0.500	**14.786**	**10.83**
D. Berman et al.[127]	**0.393**	**0.545**	**14.026**	12.74
M. Ju. et al.[130]	0.323	**0.519**	11.232	19.80
Y. Li et al. [126]	0.346	0.481	**13.713**	12.72
Z. Li et al. [128]	0.353	0.488	13.644	**12.53**
T. Yu.et al. [129]	0.214	0.338	5.684	82.61

TABLE 3.7: Comparison of SSIM, MSSIM, PSNR, MSE performance of the proposed method and different existing models in daytime and nighttime dehazing benchmarking database. The three best results are shown in red, blue, and green, respectively. The proposed model achieves the best performance compared to the other existing methods in most cases.

Metric	Proposed model	Ancuti et al.[125]	D. Berman et al.[127]	M. Ju. et al.[130]	Y. Li et al. [126]	Z. Li et al. [128]	T. Yu. et al.[129]
SSIM	**0.403**	**0.395**	**0.357**	0.339	0.350	0.344	0.268
MSSIM	**0.542**	**0.513**	**0.534**	0.527	0.500	0.509	0.347
PSNR	**13.90**	**14.24**	**13.78**	11.49	13.06	13.42	6.59
MSE	**11.24**	**11.60**	13.51	19.27	14.82	**12.99**	69.53
Runtime(s)	**0.240**	**0.300**	0.352	**0.257**	0.359	0.631	0.305

TABLE 3.8: SSIM, PSNR, MSE, MSSIM comparison of the proposed model trained using MSE, PSNR, MSSIM, SSIM and Quadruplet loss with different λ values in LAB color space and Quadruplet loss in RGB, LAB and YIQ on Reside6K database.

Loss	Color Space	MSE	PSNR	SSIM	MSSIM
MSE	LAB	0.0141	18.5079	0.7001	0.743
PSNR	LAB	0.2779	5.5606	0.3976	0.4296
SSIM	LAB	0.0017	27.4790	0.9620	0.994
MSSIM	LAB	0.0076	21.1462	0.8988	0.9378
Quadruplet loss($\lambda_1, \lambda_2 = 0.15, \lambda_3, \lambda_4 = 0.85$)	LAB	0.1253	9.0199	0.3697	0.4047
Quadruplet loss($\lambda_1, \lambda_2 = .50, \lambda_3, \lambda_4 = 0.50$)	LAB	0.5069	2.9502	0.1844	0.2074
Quadruplet loss($\lambda_1, \lambda_2 = 0.85, \lambda_3, \lambda_4 = 0.15$)**(proposed)**	LAB	**0.000001**	**68.32**	**0.997**	**1.0**
Quadruplet loss($\lambda_1, \lambda_2 = 0.85, \lambda_3, \lambda_4 = 0.15$)	YIQ	0.5335	2.5753	0.1536	0.1690
Quadruplet loss($\lambda_1, \lambda_2 = 0.85, \lambda_3, \lambda_4 = 0.15$)	RGB	0.1778	15.302	0.606	0.652

TABLE 3.9: PSNR, SSIM, MSE, MSSIM comparison of the proposed method and different existing models in I-Haze, O-Haze, Dense-Haze, and NH-Haze databases. The three best results are shown in red, blue, and green, respectively. The proposed model achieves the best performance compared to the other existing methods in most cases.

Dataset →		**I-Haze**				**O-Haze**		
Method ↓	**PSNR**	**SSIM**	**MSE**	**MSSIM**	**PSNR**	**SSIM**	**MSE**	**MSSIM**
DCP[107]	14.43	0.7516	10.44	0.8916	16.78	0.6532	7.97	**0.8916**
CAP[144]	12.24	0.6065	13.44	0.7665	16.08	0.5965	8.64	0.7365
AOD-Net[145]	13.98	0.7323	10.99	0.8523	15.03	0.5385	9.75	0.8723
QCNN-H[146]	14.118	0.612	10.825	-	17.440	0.545	7.9852	-
EDN-Net[147]	**22.9**	**0.827**	**3.93**	**0.997**	**23.46**	**0.8198**	**3.69**	**1.0**
LIDN(Proposed)	**63.32**	**0.995**	**0.038**	**1.0**	**63.26**	**0.996**	**0.04**	**1.0**
Dataset →		**Dense-Haze**				**NH-Haze**		
Method ↓	**PSNR**	**SSIM**	**MSE**	**MSSIM**	**PSNR**	**SSIM**	**MSE**	**MSSIM**
DCP[107]	10.06	0.3856	2.70	0.5256	10.57	0.5196	16.29	0.6596
DehazeNet[123]	13.84	0.4252	0.87	0.5652	16.62	0.5238	8.12	0.6638
AOD-Net[145]	13.14	0.4144	1.07	0.5544	15.4	0.5693	9.34	0.7093
EDN-Net[147]	**15.43**	**0.52**	**0.54**	**0.74**	**20.24**	**0.7178**	**5.35**	**0.9278**
LIDN(Proposed)	**68.32**	**0.988**	**0.004**	**1.0**	**59.24**	**0.992**	**0.06**	**1.0**

s

TABLE 3.10: PSNR, SSIM, MSE, MSSIM and Overhead comparison of the proposed method and different existing models in ITS, OTS database. The three best results are shown in red, blue, and green, respectively. The proposed model achieves the best performance compared to the other existing methods in most cases.

Dataset →	**ITS-Haze**				**OTS-Haze**				**Overhead**
Method ↓	**PSNR**	**SSIM**	**MSE**	**MSSIM**	**PSNR**	**SSIM**	**MSE**	**MSSIM**	**#Param**
DCP[107]	16.62	0.818	0.378	0.958	16.62	0.815	0.378	0.845	-
DehazeNet[123]	19.82	0.821	0.145	0.961	24.75	0.927	0.033	0.957	0.009M
MSCNN[148]	19.84	0.833	0.144	0.973	22.06	0.908	0.074	0.938	**0.008M**
AOD-Net[145]	20.52	0.816	0.118	0.956	24.14	0.92	0.040	0.95	0.002M
GFN[149]	22.3	0.88	0.069	0.89	21.55	0.844	0.086	0.874	0.499M
GCANet[150]	30.23	0.98	0.006	0.99	19.95	0.866	0.139	0.89	0.702M
GridDehazeNet[151]	32.16	0.984	0.003	0.994	30.86	0.982	0.0053	0.998	0.956M
MSBDN[152]	33.67	0.985	0.002	0.995	33.48	0.982	0.0024	**0.999**	31.35M
PFDN[153]	32.68	0.976	0.003	0.986	23.62	0.948	0.046	0.972	11.27M
FFA-Net[154]	36.39	0.989	0.001	0.992	33.57	**0.984**	0.002	0.997	4.456M
AECR-Net[155]	37.17	0.99	0.0008	0.993					2.611M
DehazeFormer-T[156]	35.15	0.989	0.001	0.994	33.71	0.982	0.002	0.995	0.686M
DehazeFormer-S[156]	36.82	0.992	0.0009	0.995	34.36	0.983	0.002	0.996	1.283M
DehazeFormer-B[156]	37.84	0.994	0.0006	0.997	34.95	**0.984**	0.0019	0.997	2.514M
DehazeFormer-M[156]	38.46	0.994	0.0005	0.997	**34.29**	0.983	**0.0017**	0.993	4.634M
DehazeFormer-L[156]	**40.05**	**0.996**	**0.0003**	**0.999**					25.44M
LIDN(Proposed Method)	**68.75**	**0.998**	**0.000001**	**1**	**67.89**	**0.996**	**0.000001**	**1**	5.074M

bound of the learning rate is 10^{-8}. After reducing the learning rate validation loss monitoring stops for 5 epochs, and it is called cooldown. All the hyper-parameters need for training LIDN are shown in Table 3.2.

TABLE 3.11: PSNR, SSIM, MSE, MSSIM comparison of the proposed method and different existing models in Reside6K database. The three best results are shown in red, blue, and green, respectively. The proposed model achieves the best performance compared to the other existing methods in most cases.

Dataset →	**Reside6K**			
Method ↓	**PSNR**	**SSIM**	**MSE**	**MSSIM**
DCP[107]	17.88	0.816	0.259	0.846
DehazeNet[123]	21.02	0.87	0.101	0.9
MSCNN[148]	20.31	0.863	0.125	0.893
AOD-Net[145]	20.27	0.855	0.127	0.885
GFN[149]	23.52	0.905	0.048	0.935
GCANet[150]	25.09	0.923	0.029	0.953
GridDehazeNet[151]	25.86	0.944	0.024	0.974
MSBDN[152]	28.56	0.966	0.011	**0.996**
PFDN[153]	28.15	0.962	0.012	0.992
FFA-Net[154]	29.96	0.973	0.007	0.99
AECR-Net[155]	28.52	0.964	0.011	0.981
DehazeFormer-T[156]	30.36	0.973	0.006	0.99
DehazeFormer-S[156]	30.62	0.976	0.006	0.993
DehazeFormer-B[156]	**31.45**	**0.98**	**0.004**	**0.997**
DehazeFormer-M[156]	**30.89**	**0.977**	**0.005**	0.994
DehazeFormer-L[156]				
LIDN(Proposed)	**68.32**	**0.997**	10^{-6}	**1**

3.5 Experimental Results

To verify the proposed model, comparison is done with six state-of-the-art dehazing techniques proposed by Ancuti et al.[125], D. Berman et al.[127], Y. Li et al.[126], Z. Li et al. [128], M. Ju. et al.[130], T. Yu. et al.[129]. The efficiency of the proposed model has been compared with that of the above-mentioned models, using the daytime and nighttime dehazing benchmarking database[138] for both daytime and nighttime images. The model is also compared using 14 state-of-the-art models, DCP[107], DehazeNet[123], MSCNN[148], AOD-Net[145], GFN[149], GCANet[150], GridDehazeNet[151], MSBDN[152], QCNN-H[146] PFDN[153], FFA-Net[154], AECR-Net[155], DehazeFormer-T[156] on SOTS-indoor, OTS-outdoor, OTS-mix, and DCP[107], DehazeNet[123], AOD-Net[145], CAP[144], and EDN-Net[147] are compared on I-HAZE[134], O-HAZE, Dense-HAZE[137, 157] and NH-HAZE[136] datasets.

3.5.1 Quantitative Comparison

To demonstrate the effectiveness of the proposed dehazing method, the results of the proposed method are compared with the output of the model proposed by Ancuti et al.[125], D. Berman et al.[127], Y. Li et al. [126], Z. Li et al. [128], M. Ju. et al.[130], T. Yu. et al.[129] methods, in for both nighttime and daytime images. Figure 3.12 and Figure 3.13 show the daytime and nighttime hazy images, ground-truth, and respective dehazed images obtained by the six state-of-the-art models, respectively. The naming convention of those images is, for example, in 'img1_d(n)_r(s)', img1 is the image number, d/n represents the daytime/nighttime and r/s represents the real haze/synthetic haze. It is clear that the proposed approach, in both the daytime and nighttime cases, produces the best results for real haze compared to other techniques. The model proposed by T. Yu. et al.[129] produces several artifacts, leading to poor results. This model performs better at night than it does during the day. For synthetic hazy images, the proposed model displays some artifacts, and M. Ju. et al.[130] method performs more effectively. However, for both nighttime and daytime images, the proposed model is more robust than the others. A qualitative comparison of image dehazing methods on the SOTS-mix dataset is shown in Figure 3.14, where the first two rows show the indoor and the last two rows show the outdoor result comparison of AOD-Net, GCANet, PFDN, DehazeFormer-S models with the proposed LIDN method.

3.5.2 Quantitative Evaluation

Since all algorithms operate quite correctly, comparing various models alone by seeing the results is not helpful. The quantitative comparison is thus highly beneficial. To compare with other results, this work includes four widely used complete reference metrics: MSE, SSIM, MS-SSIM, and PSNR. In section IV, all four metrics are covered. The suggested model is put to test in both the presence and absence of daylight, as well as with genuine and artificial haze. Table 3.3 compares six state-of-the-art models for daytime dehazing. The suggested model achieves the best results in terms of MS-SSIM and PSNR, but the model proposed by Ancuti et al.[125] performed best in terms of SSIM and MSE, and the proposed model ranks second. Table 3.4 compares nighttime dehazing outcomes. The suggested model performs the best in terms of MS-SSIM, MSE, and SSIM. However, the

suggested model is placed second in terms of PSNR, and Table 3.5 shows a comparison with real-haze dehazing outcomes. Regarding each of the four metrics, the recommended model yields the best results. For synthetic dehazing results, see Table 3.6. For MS-SSIM and MSE, the proposed model comes in second, and it produces the best outcomes for all SSIM. From all of the findings, it is clear that the suggested approach is more accurate than previous methods for real-haze and robust for both day and night conditions. The average result on the daytime and nighttime dehazing benchmarking database is shown in Table 3.7. The suggested technique yields the best results in terms of SSIM, MSSIM, and MSE, while in terms of PSNR, Ancuti et al.[125] method performs best, followed by the proposed method. The proposed model outperforms previous state-of-the-art models on ITS, OTS, and Reside6K datasets in terms of SSIM and PSNR as shown in Table 3.10 and Table 3.11. The proposed LIDN model achieves the best dehazing performance in both PSNR and SSIM on I-HAZE, O-HAZE, Dense-HAZE, and NH-HAZE datasets, which is shown in Table 3.9.

3.5.3 Ablation Studies

The purpose of this ablation study is to compare the performance of different loss functions and color spaces on the Reside6K database. The study aims to determine the most effective combination for image reconstruction. The proposed model utilizes the Quadruplet loss with specific weight parameters ($\lambda_1, \lambda_2 = 0.85, \lambda_3, \lambda_4 = 0.15$) and achieves the best results when applied to the LAB color space. The evaluation metrics, including MSE, PSNR, SSIM, and MSSIM, are used to compare the different approaches, and the results are summarized in Table 5.9. The findings indicate that the Quadruplet loss with the specified weight parameters performs exceptionally well in the LAB color space. It yields an impressively low MSE value of 0.000001, indicating high accuracy in the reconstructed images. The PSNR value of 68.32 indicates a good signal-to-noise ratio, while the SSIM and MSSIM values of 0.997 and 1.0, respectively, demonstrate a close similarity between the reconstructed and original images.

The proposed model is evaluated using alternative color spaces. In the YIQ color space, it achieves an MSE of 0.5335, PSNR of 2.5753, SSIM of 0.1536, and MSSIM of 0.1690. In the RGB color space, the corresponding values are 0.1778 for MSE, 15.302 for PSNR, 0.606 for SSIM, and 0.652 for MSSIM. These results highlight

the consistent superiority of the proposed model in the LAB color space across all evaluation metrics. This ablation study provides insights into the selection of loss functions and color spaces for image reconstruction tasks. The results provide the effectiveness of the Quadruplet loss with the specified weight parameters in the LAB color space. These findings have important implications for image dehazing.

3.6 Discussion

This chapter introduces the LIDN, an end-to-end single network designed to effectively combat haziness in both daytime and nighttime images. The model's training on RGB, YIQ, and LAB color spaces reveals the superiority of LAB images in achieving optimal dehazing results, outperforming RGB and YIQ images with noticeable artifacts. Extensive evaluation of the daytime and nighttime dehazing benchmarking database confirms LIDN's superior performance in real-haze scenarios, albeit with some artifacts in artificial haze conditions. The model consistently surpasses other methods in popular evaluation metrics, excluding PSNR, and demonstrates an efficient runtime of 0.240 seconds on the NVIDIA Tesla T4 15GB GPU. Furthermore, testing on the NVIDIA GEFORCE 940MX GPU yields a runtime of 0.331 seconds. The LIDN model's state-of-the-art performance is further validated on multiple datasets, although it does exhibit a color shift issue, necessitating further research for improvement.

Chapter 4

Lightweight Model for Haze Removal

4.1 Introduction

Existing algorithms have limitations, such as the loss of image details with enhancement-based methods and the need for manual feature extraction with physical model-based approaches. Deep learning-based methods still rely on physical scattering models and may cause image smoothing, loss of detail, and reduced clarity due to the use of mean square loss. Another problem in the existing dehazing methods need high processing power.

To address these issues, this chapter proposes a real-time video dehazing technique has been proposed with a novel haze parameter 'SATVAL' which is the ratio of maximum saturation to maximum value of a RGB image applied on an image scattering model using a few video frames processing in a second. A frame with a 'SATVAL' ratio below the threshold value is considered to be dehazed or else passed without dehazing. This makes a dehaze video sequence perform accurately in real-time comparable to other contemporary methods. Low-computation device like Raspberry Pi Model 4B is used to test the performance of the proposed technique. Although the proposed technique performs haze removal in low computing devices it has less generalization ability. To address this, a GAN-based dehazing model uses different generator architectures (UNet [26], PSPNet [28], MANet [27], FPN [29]) to effectively capture spatial and contextual information. The Vision Transformer serves as an encoder block, and the most efficient model is implemented on a Raspberry Pi, demonstrating practical applicability. The research contributes to real-world dehazing advancements, providing enhanced visibility in resource-constrained environments.

A. Ghosh, **A. Ali** and S. S. Chaudhuri, "**Novel Parametric Based Time Efficient Portable Real-Time Dehazing System**". Journal of Real-Time Image Processing. https://doi.org/10.1007/s11554-023-01283-x

Md. S. Akhtar, **A. Ali**, and S. S. Chaudhuri, "**MobileUnetGAN: A single Image Dehazing Model**". Signal, Image and Video Processing. https://doi.org/10.1007/s11760-023-02752-3.

A. Ali, Md. S. Akhtar, and S. S. Chaudhuri, "**GAN based Image Dehazing On Raspberry PI**". COMSYS 2023. Accepted in Springer, 2023. https://doi.org/10.1007/978-981-97-2614-1˙45

A Ghosh, **A. Ali**, S. Banerjee and S. S. Chaudhuri, "**No Reference Dataset for Daytime and Nighttime Synthetic Hazy Image**". Diary No. 18933/2023-CO/L. ROC Number: L-133885/2023. Registered

A. Ghosh, **A. Ali**, C. Ghorai, S. S. Chaudhuri "**An Energy-Efficient Portable Device for Dehazing**". Application no: 202431005438. **Published** 2024.

4.2 Contributions

The principal contributions of this chapter are outlined as:

- Introduced a haze parameter called 'SATVAL', which determines whether dehazing algorithm needs to be applied on the frame or not. This avoids the complex dehazing algorithm to be applied on all the frames leading to a reduction in computation time.
- This chapter presents a GAN-based model for effectively removing haze from images. The model utilizes different generator architectures, including PSP-Net, MANet, and FPN.
- The proposed model is evaluated on diverse datasets, including I-Haze, Dense-Haze, and NH-Haze, providing comprehensive insights into the generalization and robustness of the proposed approach across various haze conditions.
- Extensive evaluation and comparison with state-of-the-art methods consistently demonstrate the superiority of the proposed approach in terms of both quantitative metrics and visual quality. The GAN model based on MANet contributes significantly to advancements in haze removal.
- Additionally, implemented the most efficient MANet architecture on a Raspberry Pi device and evaluated its performance and real-time capabilities using live camera input. This research aims to contribute practical and effective haze removal solutions for applications such as surveillance systems and autonomous devices.

4.3 Related Work

In 2019, Li et al.[105], presented a simulation method on a hardware-efficient model of a single image dehazing technique. DSP-based image real-time dehazing optimization for improved dark-channel prior algorithm. [124]In 2019, a DSP-based real-time fog optimization technique was proposed. In 2020, Ghosh et al.[158] proposed a parallel architecture-based method on dark channels prior to a single image

dehazing technique. It is divided into basically two parts, one to calculate the atmospheric light and the second to calculate the transmission coefficient parallelly to apply in real-time application. In 2021, Ghosh et al.[159] presented a Raspberry Pi-based single-image dehazing machine, which acquires the hazy images and processes them locally. In 2021, Y Cimtay[160] proposed orientation sensors on mobile-based dehazing, where atmospheric light is estimated with the help of the differences between rotation and a predefined threshold. Most of the dehazing models are simulation-based, and not appropriate for real-time implementation. Real-time implementation is required for autonomous car driving systems so that they can run the car in real-time scenarios. A portable system makes it easier to attach the device anywhere as per requirement. The Raspberry Pi-based device makes real-time application possible as well as portable, which can be mounted with any system by peripheral devices.

4.4 Proposed Methods

In this chapter, two methods have been proposed for dehazing in low-computation devices. First, a traditional image processing method is introduced and secondly, a Patch-GAN approach for single-image dehazing is proposed for better generalization of haze removal task. The proposed approaches aim to efficiently remove haze from images and enhance their clarity. A detailed description of the network components, including the generator, discriminator, and loss functions is provided in the following subsections.

4.4.1 Method 1

The amount of haze in a weather-degraded image is determined by saturation and brightness[161]. The difference between brightness and saturation for a haze-free image is almost negligible. It is possible to assume that the depth of the scene and the haze concentration are positively associated since, generally speaking, the haze concentration rises as the scene depth changes. This relationship is represented by,

$$d(x) \propto c(x) \propto (v(x) - s(x)) \tag{4.1}$$

where s is the saturation, v is the scene's brightness, c is the haze concentration, and d is the scene's depth.

The vector space representation of the HSV color model is shown in Figure 4.1[162]. Vector I, which passes through the origin, represents the hazy image with the colors H, S, and V (value). V represents the center-line projection of I in the vertical of an inverted cone, as shown in Figure 3, where I forms an angle with a horizontal plane (H-S) of alpha (α). As a result, H and S represent I's projection in the horizontal plane, where α varies in degree [0-90]. α tangent value increase with increasing α value. As a result, when α grows, so does the angle between V and S, increasing the difference between the two. Furthermore, as V rises and S falls with an increase in α, the relationship between V and S shows the image's depth (d).

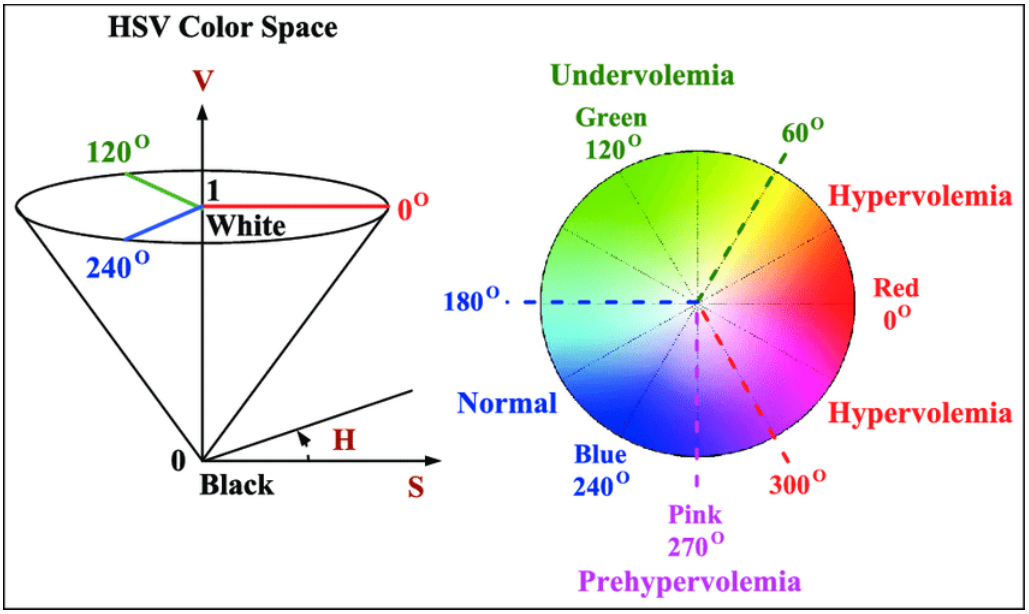

FIGURE 4.1: Geometric description of HSV color model

$$S(x) \propto (1/V(x)) \tag{4.2}$$

As a result, α and image depth are precisely proportional. To put it briefly, V, S, and d are the image I's intrinsically significant characteristics. Equation 4.1 leads to the following deduction:

$$V(x) - S(x) = K; [Constant] \tag{4.3}$$

The following equation can be derived from the one above:

$$\frac{S(X)}{V(x)} = K1 \tag{4.4}$$

K1 is a variable determined by the level of haze in each image. It follows that there are differences among the V, S, and d of a degraded image and its equivalent clean image. It is therefore impossible to calculate pixel-by-pixel whether S and V have zero values, or any value at all. Strong algorithms can reconstruct the original image from a weather-damaged one. In light of the aforementioned cue, a unique method has been developed in this chapter.

From the above discussion, a formula is proposed that represents the haze quality of the frame of the video, which is the ratio of maximum saturation to maximum brightness of the value of the frame, as represented in Equation 4.3. This ratio is named as Haziness Factor (HF). Therefore, when calculating the values for a whole image, use the highest value of S and V. This ensures that the value cannot be zero for an entire image, and if it is, it is devoid of information.

Thus, based on this attribute, this chapter proposes a formula, where the hazy quality of the image frame is expressed as follows: the ratio of the maximum saturation and maximum brightness or value,

$$SATVAL = H.F. = \frac{max(S(x))}{max(V(x))} \tag{4.5}$$

Where the value of S(x) and V(x) is represented as below

$$S(x) = \begin{cases} 0 & ;\text{if } \max\limits_{C\in R,G,B} I^C(x) = 0 \\ 1 - \dfrac{\min\limits_{C\in R,G,B} I^C(x)}{\max\limits_{C\in R,G,B} I^C(x)} & ;\text{otherwise} \end{cases} \tag{4.6}$$

$$V(x) = \max_{C\in R,G,B} I^C(x) \tag{4.7}$$

The HF is calculated and then compared to a threshold value to determine whether to display the haze as is or remove it. Testing with various images allows for the detection of the threshold value. It will be dehazed if it is less than the threshold value; otherwise, dehazing will be required. Real-time video processing is greatly

aided by this reduction in processing time and complexity. This approach, as seen below, will lessen information loss as well.

Algorithm 1: SATVAL Algorithm for Dehazing Hazy Frames

Input: Hazy frames
Output: Dehazed frames
1 **for** *each frame* **do**
2 Calculate $S(x)$ and $V(x)$ for all pixels in the image;
Calculate $\max(S(x))$ and $\max(V(x))$ of the image;
Calculate SATVAL=$\frac{\max(S(x))}{\max(V(x))}$ parameter;
if $SATVAL < threshold$ **then**
Calculate Atmospheric Light;
Calculate Transmission Coefficient;
Calculate Recover the scene;
end
3 **end**

The detailed process is explained in detail by the line 2. It will process each frame of a video one at a time, dehazing just the necessary frames. The algorithm selects the frames that require processing after examining them.

This method achieves 10FPS for 128×128 video frames, the detailed comparison is shown in Table 4.1. Although the proposed technique uses less computing power it has a less generalization ability for haze removal tasks, the result is shown in Table 4.2.

TABLE 4.1: Performance comparison of the video processing with different sizes of frames in Raspberry Pi Model.

Video ID	FPS (128 Kbps)	FPS (256 Kbps)	FPS (512 Kbps)
1	1.319	0.347	0.086
2	3.138	0.8359	0.419
4	10.544	10.556	10.5193
5	1.8541	0.9655	0.1684
8	1.873	0.5604	0.1699
9	14.652	13.8793	10.4403
10	2.6676	0.9354	0.3243
11	4.1096	1.2419	0.351
12	9.8844	5.149	2.3399

TABLE 4.2: Experimental results of the proposed method deployed in Raspberry Pi 4B Model on different video sequences

Frame No.	Hazy Image	Dehazed Image
1		
2		
4		
5		

4.4.2 Method 2

Improved learning algorithms are needed to overcome the shortcomings of the earlier research. Generative Adversarial Networks (GANs) and conditional GANs are two promising methods that have shown promise in a variety of image translation and reconstruction challenges. It is also possible to enhance the visual quality of images by using perceptual loss mechanisms. These factors lead to the development of a novel GAN-based haze removal model that is trained on a network capable of removing haze from a single image using Patch-GAN.

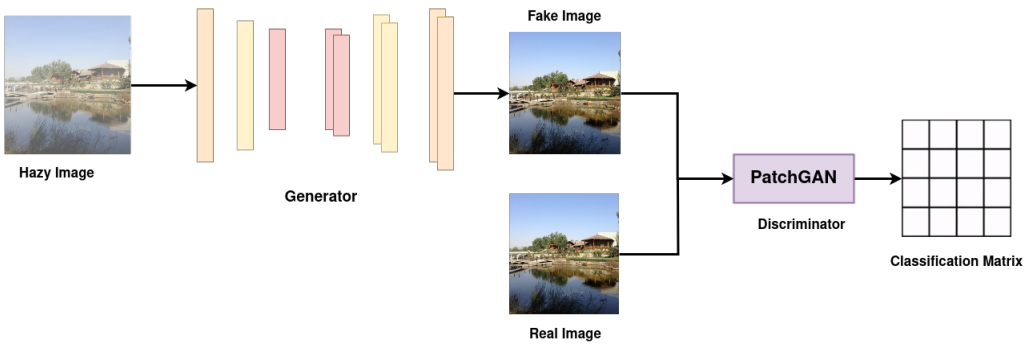

FIGURE 4.2: An illustration of proposed GAN architecture

4.4.2.1 Generator

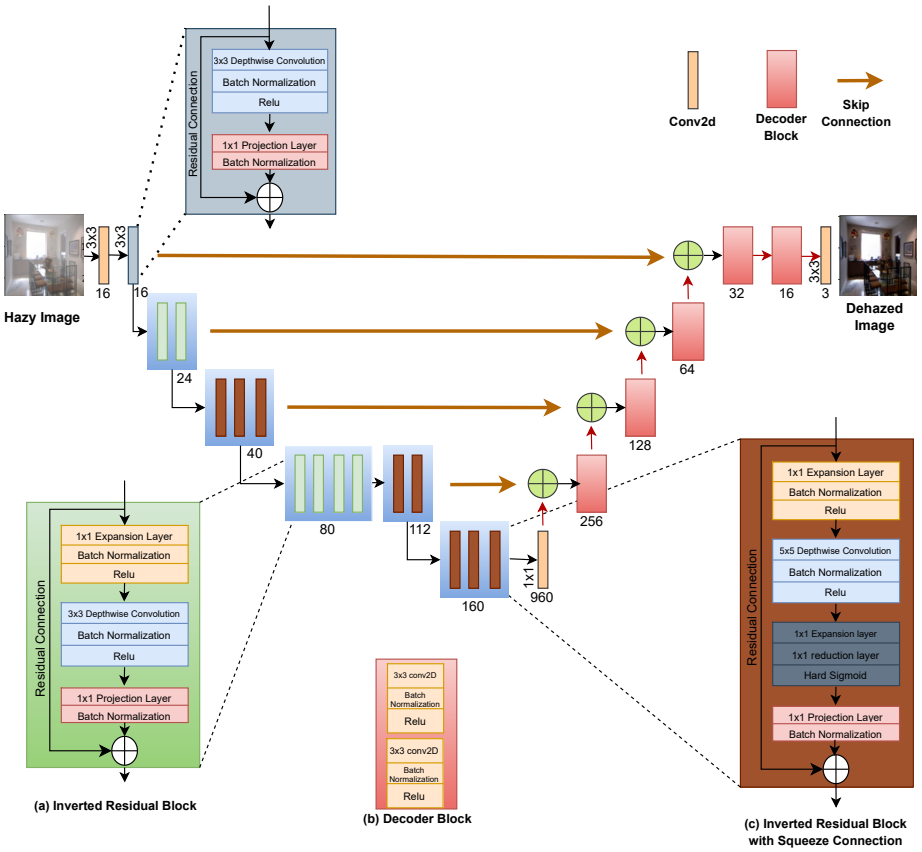

FIGURE 4.3: Illustration of proposed generator architecture-(a)Inverted Residual block. (b) Decode block. (c)Inverted Residual block with Squeeze connection.

The generator in the proposed method is a patch-based MANet and U-Net network designed to produce high-quality dehazed images from hazy inputs. It consists of an encoder and a decoder. The encoder acts as a feature extractor, capturing relevant information from the hazy image. Different backbone architectures, such as modified-MobileNet, Mix Vision Transformer, ResNet, VGG16, and others, are experimented with to extract robust features from the hazy input. Among these, the MobileNet backbone is found to provide the best performance in terms of generating high-quality dehazed images. The decoder then reconstructs the dehazed output by upsampling the encoded features through transposed convolutions, matching the original input image's size. Skip connections are utilized during the upsampling process, allowing the generator to leverage low-level and high-level features simultaneously for better dehazing results. To train the generator, a combination of loss functions, including MSSIM loss, L1 loss, and Binary Cross-Entropy with Logits (BCEWithLogits) loss, measures the difference between

the generated dehazed image and the ground truth clear image, encouraging perceptually similar outputs. Through iterative training, the generator's parameters are optimized to produce high-quality dehazed outputs for unseen hazy images during inference.

4.4.2.2 Discriminator

The discriminator plays a vital role in distinguishing real (clear) images from fake (generated dehazed) ones. It acts as an adversarial network, providing feedback to the generator to enhance the overall output quality. The discriminator architecture consists of four convolutional layers followed by fully connected layers. It takes the generator's output and the corresponding original image as input and predicts the probability of the input being real or fake. During training, the discriminator is optimized to minimize the BCEWithLogits, which helps accurately classify real and fake images. Employing an adversarial framework with the generator, the proposed method effectively learns to generate high-quality dehazed images closely resembling the ground truth (clear) images.

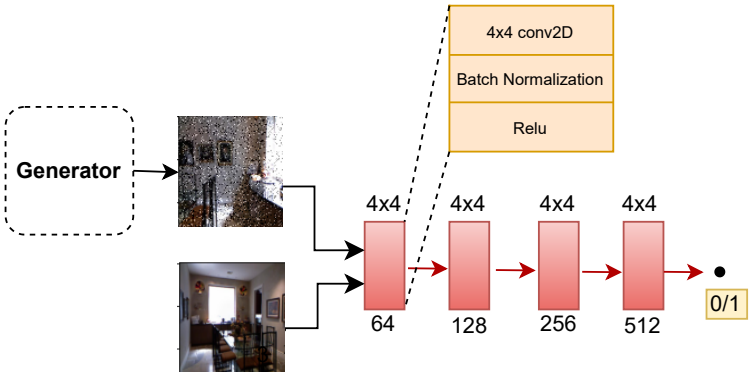

FIGURE 4.4: Discriminator architecture

4.4.2.3 Loss Function

The proposed method uses multiple loss functions to optimize the generator and discriminator networks during the training process. The generator's total loss is formulated as follows:

$$L_G = \lambda_1 L_{L1} + \lambda_2 L_{MS-SSIM} + \lambda_3 L_{BCE} \tag{4.8}$$

where, λ_1, λ_2, λ_3 are constants. In this chapter λs are set as, $\lambda_1 = 200$, $\lambda_2 = 200$, $\lambda_3 = 1$. The higher value (200) of λ_1, and λ_2 reduces these artifacts[163]. The L1 loss, sometimes referred to as the mean absolute error (MAE) loss, calculates the pixel-wise difference between the dehazed image and the original image. It is defined as:

$$L_{L1} = \frac{1}{N} \sum_{i=1}^{N} \|G(I_i) - J_i\| \tag{4.9}$$

where $G(I_i)$ is the dehazed, output of the generator network G for the input hazy image I_i, J_i is the corresponding ground truth, and N is the sample size.

The ground truth image and the dehazed image are compared at various scales using a method called multi-scale structural similarity (MS-SSIM) loss. It is defined as:

$$L_{MS-SSIM} = 1 - \frac{1}{N} \sum_{i=1}^{N} \frac{1}{M} \sum_{j=1}^{M} MS\_SSIM\left(G(I_i), J_i\right) \tag{4.10}$$

where $MS\_SSIM\left(G(I_i), J_i\right)$ is the MS-SSIM score between the generated dehazed image and the original image, calculated at M different scales.

The BCEWithLogits loss quantifies the dissimilarity between the predicted probability generated by the discriminator network and the corresponding ground truth label. It is defined as:

$$L_{BCE} = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log(D(J_i)) + (1 - y_i) \log(1 - D(G(I_i)))\right] \tag{4.11}$$

where D is the discriminator network, y_i is the ground truth label (1 for clear images and 0 for hazy images), and N is the training size.

The discriminator loss is computed solely using the BCEWithLogits loss function. Its purpose is to enable the discriminator to differentiate between the dehazed images generated by the generator and the original ground truth images. The loss is defined as:

$$L_D = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(D(J_i)) + (1 - y_i)\log(1 - D(G(I_i)))] \tag{4.12}$$

where D is the discriminator network, y_i is the ground truth label (1 for clear images and 0 for hazy images), and N is the training size.

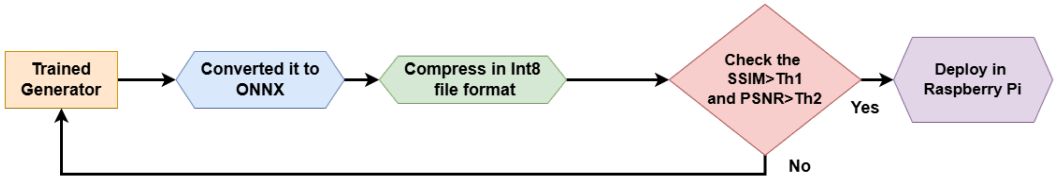

FIGURE 4.5: Flowchart of real-time deployment steps in Raspberry Pi

4.4.2.4 Real-Time Implementation

For real-time implementation on the Raspberry Pi, the MANet architecture with the mit_b3 encoder has been chosen for its excellent performance and computational efficiency. To facilitate integration, the model has been converted the generator component to the ONNX format. Additionally, model compression techniques, like reducing precision to int8, are employed to optimize size and complexity while maintaining dehazing performance. The real-time deployment steps are shown in Figure 4.5. To further enhance processing speed, the Movidius Neural Compute Stick accelerator is utilized, enabling real-time dehazing at an impressive rate of approximately 11 frames per second (FPS) on the Raspberry Pi.

4.5 Experiments and Analysis

4.5.1 Datasets

To overcome the challenges of acquiring a large-scale dataset of real atmospheric hazy images, the researchers leveraged the Reside6K dataset[23]. The Reside6K dataset provides a diverse and comprehensive collection of hazy images for training and evaluation. The training set of the Reside6K dataset consists of 6,000 hazy images, including an equal distribution of 3,000 indoor hazy images and 3,000

outdoor hazy images, along with their corresponding ground truth (clear) images. For a fair assessment of their proposed method, a separate test set comprising 1,000 hazy images is employed. This test set includes 500 indoor hazy images and 500 outdoor hazy images, enabling the researchers to evaluate the generalization and performance of their approach. The hazy images in the Reside6K dataset are synthetically generated by introducing atmospheric haze effects to the corresponding clear images. This process ensures that the hazy images exhibit realistic and diverse characteristics, such as varying intensities of haze, distinct color tones, and spatial distributions. By training and evaluating their model on this dataset, it is possible to effectively address different hazy conditions encountered in real-world scenarios.

To facilitate a comprehensive comparison of their proposed method with existing techniques, the I-Haze [30], Dense-Haze [31], and NH-Haze [32] datasets are used for testing purposes. These datasets allowed them to evaluate the performance of their model across different hazy scenarios and environments. Widely used evaluation metrics such as peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) were employed to quantitatively measure the effectiveness of the proposed dehazing model.

Utilizing the Reside6K dataset with its diverse hazy images and balanced training and test splits, ensures a robust evaluation of the proposed method's performance. This dataset serves as a valuable resource for training, evaluating, and comparing their model with other state-of-the-art dehazing techniques, while the additional datasets provide further insights into the generalizability of the proposed approach.

4.5.2 Training Details

In this section, an overview of the experimental setup and training details for the proposed method are provided. The model is initialized by incorporating pre-trained ImageNet weights into the encoder block, enabling it to effectively capture relevant visual patterns from the hazy input images. During the training process, a combination of loss functions is employed to guide the learning of both the generator and discriminator components of the GAN. The generator loss encompasses a diverse set of loss functions, including MSS loss, L1 loss, and BCEWithLogits loss. This combination of loss functions collectively facilitates the generation of high-quality clear images. On the other hand, the discriminator loss solely focuses on

TABLE 4.3: The outcomes of comparing proposed dehazing techniques on Reside6K with various state-of-the-art techniques.

Method		DCP[164]	DehazeNet[165]	AOD-Net[166]	DCPDN[167]	GFN[168]	Ours(UNet)	Ours(MANet)
Indoor	PSNR	16.62	21.14	19.06	15.85	22.3	**25.82**	**27.58**
	SSIM	0.8179	0.8472	0.8504	0.8175	0.88	**0.926**	**0.927**
Outdoor	PSNR	19.13	22.46	20.29	19.93	21.55	**29.37**	**24.89**
	SSIM	0.8148	0.8514	0.8765	0.8449	0.8444	**0.965**	**0.893**
Run-time(s)		3.002	0.831	0.106	6.053	-	**0.100**	**0.060**

TABLE 4.4: PSNR and SSIM comparison of the proposed dehazing techniques using various encoders on Reside6K, Reside6K-indoor (ITS), and Reside6K-outdoor (OTS) dataset.

Model	Encoder	Reside6K		ITS		OTS		Run-time(s)
		SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	
Unet	**Mix Vision Transformer**	0.903	26.37	0.879	24.74	0.928	27.99	0.478
Unet	**Resnet152**	0.935	25.94	0.919	24.85	0.952	27.03	0.408
Unet	**Vgg19**	0.930	25.85	0.912	25.09	0.947	26.60	0.437
Unet	**modified-MobileNet**	0.949	**27.86**	0.926	25.82	**0.965**	**29.37**	0.100
Unet	**Efficientnet**	**0.950**	27.81	**0.935**	**26.51**	0.965	29.10	0.346

BCEWithLogits loss, aiding the discriminator in distinguishing between real and generated clear images. To optimize the learning process, the learning rate is set to 0.001, and the Adam optimizer with default parameters is utilized to optimize the network. The training is conducted on the Reside6K dataset, which consists of 6,000 hazy images for training and 1,000 hazy images for testing. Mini-batch training with a batch size of 16 is employed to enhance the training efficiency. This approach allowes for effective adjustment of the model weights based on the computed losses and gradients.

TABLE 4.5: PSNR and SSIM comparison of the proposed dehazing techniques using various encoder on I-Haze, Dense-Haze and NH-Haze datasets

Model	Encoder	I-Haze[30]		Dense-Haze[31]		NH-Haze[32]	
		SSIM	PSNR	SSIM	PSNR	SSIM	PSNR
Unet	**Mix Vision Transformer**	0.716	15.98	0.408	**11.97**	0.497	12.72
Unet	**Resnet152**	0.684	14.96	0.353	11.19	0.490	**12.92**
Unet	**Vgg19**	0.711	15.67	0.366	10.90	0.464	12.27
Unet	**modified-MobileNet**	0.756	16.59	0.384	11.21	0.482	12.68
Unet	**Efficientnet**	**0.775**	**16.84**	**0.411**	11.79	**0.512**	12.72

4.5.3 Quantitative and Qualitative Evaluation

A thorough evaluation of the proposed dehazing method is conducted, comparing it against several state-of-the-art approaches, namely DCP, AOD-Net, Cycle-dehaze, GridDehazeNet, and FFA-Net. The evaluation encompasses both quantitative and qualitative assessments to validate the performance of the method.

TABLE 4.6: Outcomes of comparing different dehazing architectures with various Mix Vision Transformers encoder on the OTS dataset.

Encoder	MANet		FPN		PSPNet	
	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR
mit_b0	0.875	26.65	0.724	22.87	0.617	21.26
mit_b1	0.875	25.92	**0.756**	**24.03**	**0.624**	**21.71**
mit_b2	0.892	26.30	0.726	22.84	0.619	21.51
mit_b3	**0.927**	**27.58**	0.723	22.78	0.616	21.60
mit_b5	0.888	27.16	0.749	23.93	0.561	19.76

TABLE 4.7: Outcomes of comparing different dehazing architectures with various Mix Vision Transformers encoder on the ITS dataset.

Encoder	MANet		FPN		PSPnet	
	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR
mit_b0	0.823	23.31	0.672	20.21	0.543	18.73
mit_b1	0.817	23.44	0.707	20.96	0.551	18.73
mit_b2	0.863	**24.89**	0.668	20.30	0.550	18.25
mit_b3	**0.893**	24.51	0.674	20.47	**0.560**	**19.01**
mit_b5	0.829	24.52	**0.712**	**21.00**	0.538	18.62

TABLE 4.8: Comparison of MANet architecture with Mix Vision Transformer Encoder on various datasets

Encoder	ITS		OTS		I-Haze		Dense-Haze		NH-Haze	
	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR
mit_b0	0.823	23.31	0.874	26.65	0.680	16.04	**0.362**	**11.33**	0.442	**12.48**
mit_b1	0.816	23.43	**0.874**	**25.91**	**0.700**	**16.55**	**0.404**	11.56	**0.473**	12.31
mit_b2	0.862	24.89	**0.891**	26.29	0.691	**16.28**	0.403	**12.00**	0.447	12.54
mit_b3	0.892	**24.51**	**0.926**	**27.58**	**0.720**	**16.10**	**0.395**	11.62	0.458	**12.68**
mit_b5	**0.829**	**24.51**	**0.888**	**27.16**	**0.7051**	16.23	**0.392**	11.50	**0.442**	12.30

For quantitative evaluation, two widely used metrics, PSNR and SSIM, are employed. PSNR measures the fidelity of reconstruction between the dehazed image and its corresponding ground truth clear image, while SSIM evaluates the structural similarity between the dehazed image and the ground truth. Higher PSNR and SSIM values indicate better performance in terms of reconstruction accuracy and similarity to the ground truth.

The evaluation results, as depicted in Table 4.3, provide quantitative evidence of the outstanding performance of the proposed method. On the outdoor dataset of Reside6K, the approach achieves an average PSNR of 29.37 and an average SSIM of 0.965. On the indoor dataset of Reside6K, it attains an average PSNR of 25.82 and an average SSIM of 0.926. These results demonstrate the superiority of the method compared to other methods in terms of both metrics. Overall,

the approach achieves exceptional results in image quality and similarity to the ground truth. From Table 4.3 it can be seen that the proposed model required less run-time i.e., it achieves higher performance than other state-of-the-art dehazing models.

The quantitative evaluation results, as presented in Table 4.4, showcase the performance of the proposed method with different encoder blocks. Specifically, when evaluated on the outdoor dataset of Reside6K, the modified MobileNet encoder achieves the highest average PSNR of 29.37 and an average SSIM of 0.965. Conversely, when evaluated on the indoor dataset of Reside6K, the EfficientNet encoder achieves the highest average PSNR of 25.82 and an average SSIM of 0.926. These results unequivocally demonstrate the superiority of the MobileNet encoder over other encoders in terms of both PSNR and SSIM metrics, highlighting its effectiveness in preserving image details and capturing structural similarities. The results presented in Table 4.5 demonstrate the performance of various encoder blocks on different datasets, including I-Haze, Dense Haze, and NH Haze. It is evident from the table that the models exhibit favorable performance, particularly on the I-Haze dataset. Table 4.4 shows the the proposed modified-MobileNet encoder has less run-time than other encoders.

For qualitative evaluation, a visual comparison of the dehazed results is conducted. Figure 4.6 presents a selection of sample images from the Reside6K dataset, where the first column depicts the hazy input images, and the subsequent columns showcase the dehazed outputs generated by the method in question and the compared approaches. The last column shows the ground truth images corresponding to the hazy images. Through visual inspection, it becomes apparent that the proposed method effectively reduces haze, enhances image clarity, and preserves crucial details when compared to the other methods. Figure 4.7 shows the output of dehazing results of the different encoders and the proposed modified-MobileNet encoder. The result depicts that the proposed model generates better visual quality and higher haze removal ability than other models. Efficientnet also achieves significant performance but it has a lot of artifacts than other models. Although the proposed model generates a good visual result, however sometimes the images get blurred as shown in Figure 4.7. The combined quantitative and qualitative evaluations provide strong evidence for the effectiveness of the proposed dehazing method. The best performance in terms of PSNR and SSIM, coupled with the visually pleasing dehazed results, validates the efficacy of the approach in mitigating

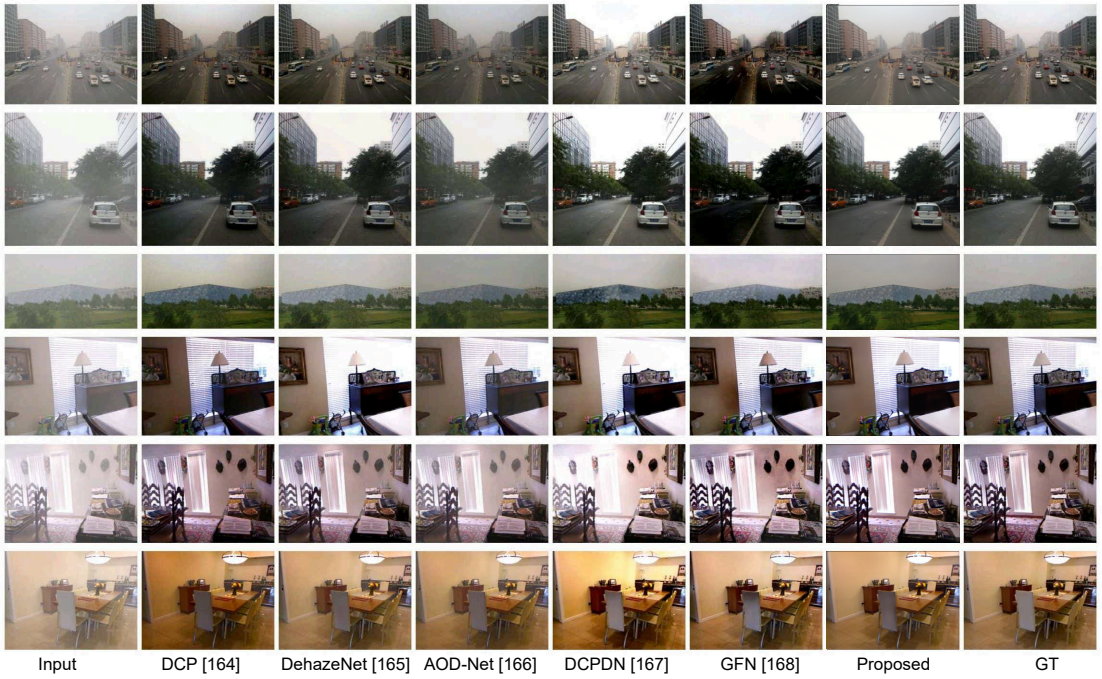

FIGURE 4.6: Comparison of the state-of-the-art dehazing methods on Reside6K. The upper three rows show the dehazing results on outdoor images and the bottom three rows show the dehazing results on indoor images.

haze, improving image clarity, and preserving important image details.

For quantitative evaluation, two widely used metrics, PSNR and SSIM are utilized, to measure reconstruction accuracy and structural similarity between the dehazed and ground truth images. The results in Table 4.3 clearly demonstrate the outstanding performance of the proposed approach, achieving higher PSNR (24.89 for outdoor and 27.58 for indoor) and SSIM (0.893 for outdoor and 0.927 for indoor) values compared to other methods.

Additionally, the performance of different dehazing architectures with Mix Vision Transformers on Reside outdoor and indoor datasets is evaluated. MANet consistently outperforms other architectures, especially when paired with the mit_b3 encoder, achieving the highest SSIM and PSNR values, indicating its ability to produce high-quality dehazed images. The choice of encoder also plays a crucial role, with mit_b5 and mit_b3 encoders demonstrating better performance.

Furthermore, Table 4.8 presents the outcomes of comparing MANet with various Mix Vision Transformer encoders on multiple datasets, where MANet exhibited relatively higher performance on the I-Haze dataset.

TABLE 4.9: PSNR and SSIM comparison of the proposed dehazing technique using various values of λ_1, λ_2 and λ_3 on Reside6K NH-Haze dataset. Final values of λ_1, λ_2 and λ_3 are shown in **bold**.

λ_1	λ_2	λ_3	**PSNR**	**SSIM**
1	1	1	10.61	0.357
1	1	200	8.42	0.292
200	1	1	11.08	0.358
200	-	1	10.91	0.361
1	200	1	18.04	0.569
-	200	1	10.42	0.355
200	**200**	**1**	27.86	**0.950**

Overall, the combined quantitative and qualitative evaluations provide strong evidence for the efficacy of the proposed dehazing method, demonstrating its superior performance in mitigating haze and improving image clarity.

4.5.4 Ablation Study

The ablation study focused on evaluating dehazing techniques through a comprehensive analysis of the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM). The study utilizes the Reside6K dataset and investigates the impact of the constant values λ_1, λ_2, and λ_3, on the proposed dehazing methods as shown in Table 4.9. The main objective is to identify the most effective combination of these parameters to enhance dehazing performance. The findings reveal interesting insights into the impact of these parameters on dehazing performance. Specifically, the last row values represent the most promising outcome, as it achieves the highest PSNR of 27.86 and an impressive SSIM of 0.950. tab:abstudy shows that the higher λ_1 and λ_2 values are more effective than the higher values of λ_3.

4.6 Discussion

The chapter introduces a novel parameter called 'SATVAL' for detecting the haziness in a particular frame, which reduces the computation time for haze removal. By applying the technique, the complexity is reduced, but the method lacks generalization. To address this, a patch-GAN based on a novel approach for single-image

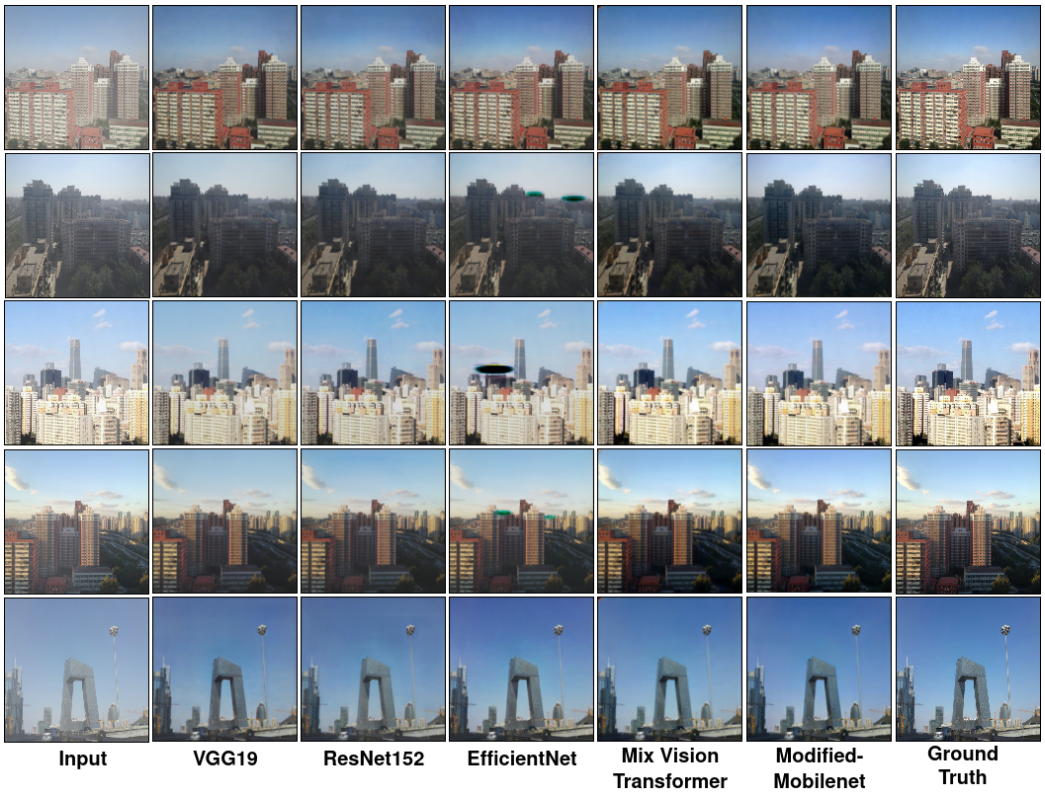

FIGURE 4.7: Comparison of the state-of-the-art dehazing methods on Reside6K. The first column and the last column are the input hazy and haze-free ground truth, respectively. Other columns are the output of dehazing results of the different encoders.

dehazing. The model explores and compares different generator architectures (U-Net, PSPNet, MANet, and FPN) with different encoder blocks of a mix of Vision Transformers, ResNet121, VGG19, modified-MobileNet, and EfficientNet. Extensive evaluations on Reside6K, I-Haze, Dense-Haze, and NH-Haze datasets demonstrate that the MANet architecture with the mix of Vision Transformers encoder consistently outperforms other configurations, both quantitatively and qualitatively, in terms of image restoration and visibility improvement. By utilizing a combination of loss functions, the generator and discriminator components are effectively trained. The experimental results on the Reside6K dataset demonstrate the effectiveness of the proposed method compared to several advanced dehazing techniques, both quantitatively and qualitatively. Real-time implementation on resource-constrained devices, like the Raspberry Pi, achieved a processing speed of 11 FPS. The proposed approach can be useful as a preprocessing block for other computer vision applications in hazy weather.

Chapter 5

A Unified Model for Haze and Rain Removal

5.1 Introduction

Degradation in the quality of outdoor images due to adverse weather conditions poses a significant challenge for various computer vision tasks, including object detection, object tracking, and video object segmentation. Ensuring robustness against weather-induced image or video deterioration is vital, particularly in computer vision-based surveillance systems and autonomous driving applications. The most common infractors for reduced image or video visibility are haze and rain.

The degradation caused by rain can be categorized into two primary types. First, rain streaks in proximity to the camera resemble salt-and-pepper noise, while second, rain streaks farther from the camera manifest as haze or mist. Although there are distinct similarities between the effects of haze and rain, researchers often treat them as separate challenges for image enhancement or noise removal. While numerous studies have addressed image restoration affected by either haze or rain individually, relatively few have focused on restoring images impacted by both phenomena using a single method, despite their natural co-occurrence.

Recent methods for addressing this challenge can be broadly classified into two categories: single-image-based and video-based approaches. It is worth noting that single-image-based approaches are more challenging due to the absence of essential background and foreground information when compared to video-based methods.

Current strategies for haze and rain removal can further be categorized into two distinct approaches: data-driven and prior-based methods. The data-driven approach leverages extensive datasets to optimize its algorithms, resulting in superior generalization capabilities. Conversely, prior-based approaches rely heavily on mathematical and physical models, often exhibiting a reduced generalization capacity. A prominent example of a prior-based dehazing method estimating the transmission coefficient and atmospheric light [169]. Some techniques, such as the one proposed by Hautiere et al. [170], emphasized maximizing local contrast degraded by haze. Similarly, He et al. [171] introduced a dark channel prior-based

A. Ali, R. Sarkar and S. S. Chaudhuri, "**Wavelet based Auto-Encoder for Simultaneous Haze and Rain Removal from Images**". Pattern Recognition, 2024. https://doi.org/10.1016/j.patcog.2024.110370

A. Ali, C. Ghorai, S. Ganguly, R. Sarkar and S. S. Chaudhuri "**Portable Real-Time Haze and Rain Removal Device for Enhanced Video Surveillance**". Application no: 202431046425. Journal Number- 26/2024, **Published** 2024.

physical model that achieved commendable results. However, these prior-based assumptions may not always be applicable in real-time scenarios and can be less robust due to their dependence on specific assumptions and constraints related to hazing and atmospheric conditions. Consequently, when confronted with deviations from these assumptions, such as the presence of multiple light sources, non-uniform haze distributions, or intricate scene geometries, prior-based methods may struggle to accurately estimate and remove haze artifacts.

On the one hand, traditional approaches to rain removal have predominantly focused on addressing the challenge of eliminating rain streaks from images, as exemplified by previous methods such as [172]. Additionally, methods grounded in prior-based strategies have attracted interest within the research community. Notably, Barnum et al. [173] delved into frequency prior-based rain removal techniques. More recently, the field has witnessed a surge in convolutional neural network (CNN)-based data-driven models dedicated to deraining [174], achieving state-of-the-art results, especially on synthetic datasets. In supervised training for deep learning models targeting haze removal, conventional techniques rooted in standard atmosphere scattering models have found favor [175]. Li et al. [174], for instance, proposed a channel attention-based model capable of effectively removing multiple rain streaks from a single image.

However, despite the myriad of methods proposed for dehazing and deraining as separate tasks, the literature offers only a limited number of approaches that simultaneously address both challenges using a unified methodology. Xiaohong et al. [176] proposed the Densely Scale-Connected Attentive Network (DSCAN), leveraging physical model principles. DSCAN employs multi-scale networks to enhance information exchange and aggregation effectively. Wenhan et al. [177] introduced a multi-task deep learning model proficient in binary rain-streak detection and removal from degraded images. Their approach incorporated a contextualized dilated network, enabling robust contextual information utilization for improved accuracy. Vijay et al. [178] harnessed a multi-scale encoder, incorporating domain-aware filtering modules within a generative adversarial network (GAN) framework. Furthermore, they incorporated recurrent features from previous frames to ensure temporal consistency and attained state-of-the-art results for video-based dehazing and deraining. Xiao et al. [179] introduced a Selective Attention Module (SAM) to capture both haze and rain effects, enhancing estimation precision through channel-wise and spatial-channel attention mechanisms.

Lastly, Hao et al. [180] devised an end-to-end CNN-based network, which served the dual purpose of dehazing and deraining tasks.

Previous networks have predominantly relied on deep CNN models operating solely in the spatial domain. However, limited attention has been given to the fusion of frequency domain information with CNN models. Weather-degraded images consist of high-frequency components, encompassing details like rain-streaks and edge characteristics, alongside low-frequency components that manifest as hazy, blurry effects with limited informational content.

Dong et al. introduced FHRR-Net [181], a deep neural network rooted in frequency-based processing. This network, adopting an encoder-decoder architecture, leverages guided filters based on frequency decomposition. Similarly, Liang et al. [182] proposed a wavelet-based deep CNN framework, which employs the Haar wavelet transformation to establish an end-to-end mapping between hazy/rainy images in the wavelet domain and their respective ground truth wavelet-transformed counterparts. Wavelet transformation, distinct from Fourier transformation, is found to be advantageous due to its capacity to localize signal features in both time and frequency domains.

Wavelet-transformed images possess a unique characteristic:- they encompass both local and global spatial as well as frequency features. This attribute enhances the model's capacity to capture contextual information, surpassing the capabilities of existing pooling-based networks. Consequently, the integration of wavelet transformations augments the learning process, contributing to more effective handling of weather-induced image degradation.

Adverse environmental conditions, such as haze and rain, exert a detrimental influence on image quality, which is a challenge to computer vision applications. Traditional solutions often struggle with the simultaneous mitigation of both haze and rain effects, necessitating a novel approach. To address this, the Wavelet-based Auto-Encoder (WAE) is introduced, a method that uses wavelet transformations as an alternative to standard pooling and upsampling operations. This not only streamlines the process of feature dimension reduction and restoration but also introduces an effective way for capturing spatial and frequency features within the hidden layers. The WAE's distinctive capability to conduct multi-resolution analysis empowers the identification of both low- and high-frequency features, providing a nuanced and comprehensive understanding of intricate scenes.

5.2 Contributions

In this chapter, a wavelet-based deep auto-encoder network is proposed, which uses wavelet and inverse-wavelet transformation to design encoder and decoder networks, respectively. The key contributions include:

- A novel Wavelet-based Auto-Encoder is proposed, called WAE, which can remove both haze and rain effects from images.
- The network uses wavelet transformation as an alternative to the max-pooling operation in order to reduce the feature dimension, and inverse wavelet transformation as an alternative to the up-sampling operation.
- Wavelet transformation on each layer helps multi-resolution analysis, which is useful for identifying both low-frequency and high-frequency features.
- The model has the ability to localize features in both time and frequency domains, thereby helping to analyze non-stationary features.
- The proposed model achieves state-of-the-art performance, while evaluated on rain and haze-affected image datasets.

5.3 Related Work

This section demonstrates the physical rain model, haze model, and joint rain-haze model, along with a comprehensive analysis of each type of physical model.

5.3.1 Rain Model

Rainy images comprise mainly two types of components such as rain-free output images and rain-streaks[183]. The physical rain model is mathematically described as:

$$I(x) = O(x) + R(x) \tag{5.1}$$

where $I(x)$ is a input rainy image, $O(x)$ is the rain-free image, $R(x)$ is the rain-streaks, and x is the pixel intensity of the image.

5.3.2 Haze Model

McCartney [184] proposed the physical model for haze removal depending upon atmospheric scattering, called the atmosphere scattering model (ASM), which is mathematically described as,

$$I(x) = O(x)t(x) + \alpha[1 - t(x)] \tag{5.2}$$

where $I(x)$ is the input hazy image, $O(x)$ is the haze-free image, $t(x)$ is transmission coefficient, α is the global atmospheric light, and x is the pixel intensity of the image. The transmission coefficient, $t(x)$ depends on the atmospheric scattering of the light due to aerosols present in the atmosphere and the camera distance from the object. The mathematical description of the transmission map is as follows,

$$t(x) = e^{-\beta d(x)} \tag{5.3}$$

where β is a constant, called scattering coefficient, and $d(x)$ is the scene depth.

5.3.3 Rain-Haze Model

Due to the rain veiling effect caused by heavy rain, long-distance rain looks like haze. By integrating ASM with the rain model, the researchers were able to develop a rain-haze model[177]. Depending on the superposition order of rain and haze effects, there exist two different integration approaches. The first integration approach can be written as,

$$I(x) = O(x)t(x) + \alpha[1 - t(x)] + R(x) \tag{5.4}$$

The above integration is called haze-first approach[185].

Another method of integration is called the rain-first approach [186], which is expressed as:

$$I(x) = [O(x) + R(x)]t(x) + \alpha[1 - t(x)] \tag{5.5}$$

To account for heavy rain factors, an extension was made to Equation 5.5 by Wenhan et al. [177], resulting a new model, which is described as:

$$I(x) = [O(x) + \sum_{i=1}^{L} R_i(x)]t(x) + \alpha[1 - t(x)] \tag{5.6}$$

where L is the number of rain-streak layers present in a heavy rain situation. Finally, the clear output image can be obtained from Equation 5.7, which is described as:

$$O(x) = \frac{[I(x) - \alpha]}{t(x)} - [\sum_{i=1}^{L} R_i(x) + \alpha] \tag{5.7}$$

Although there are several methods developed for simultaneously removing haze and rain effects from images using Equation 5.7, however, there are three problems exist in this approach that include:

- Rain streaks exhibit non-homogeneous characteristics, which means that their distributions across different regions of an image can vary.
- The separation of rain and haze may lead to an over-smoothing issue in areas that are not affected by rain or haze.
- The CNN models relying on Equation 5.7 operate on local image patches or have a limited receptive field. As a result, they lack spatial contextual information from broader regions, leading to the occurrence of artifacts and a blurred effect in the output image.

5.3.4 Wavelet for joint Rain and Haze Removal

Frequency-based methods are effective for contextual information learning, however, lots of local spatial information by transferring an image into the frequency domain by Fourier transform, may not be desirable. There is another method called Wavelet transform, which is very useful in this situation. Liang et al. [182] developed a Haar wavelet-based deep CNN model, which maps between an input wavelet-transformed noisy image to wavelet transformed noise free image.

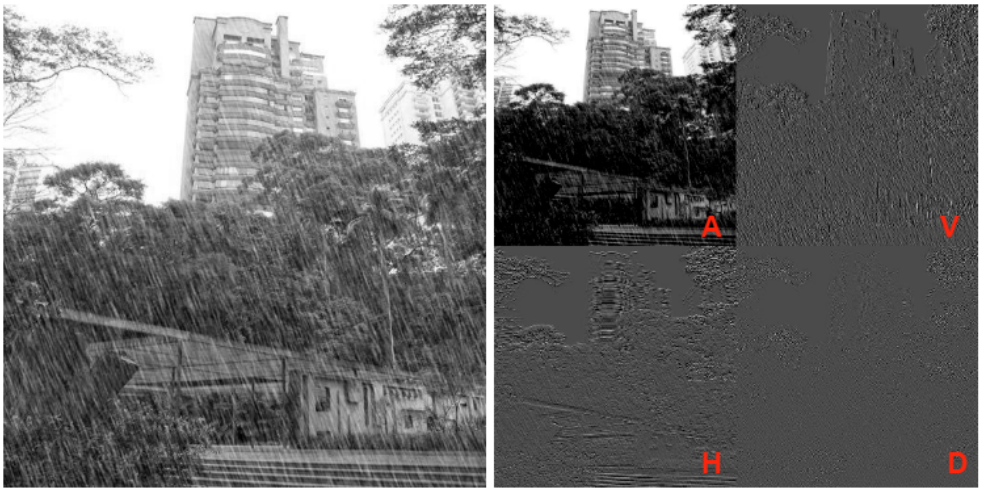

FIGURE 5.1: Example of Haar wavelet transformation. (a) Input image (b) Wavelet transformed image, where A, V, H, and D are four wavelet components.

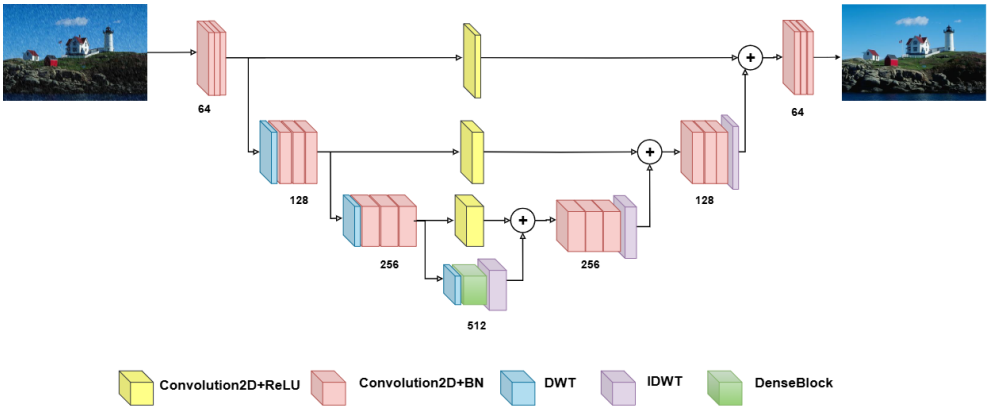

FIGURE 5.2: An illustration of the proposed Wavelet-based Auto-Encoder (WAE)

5.4 Proposed Method

This section discusses the Wavelet Transformed-based Auto-Encoder (WAE), a proposed wavelet-based dehaze-derain network. The architecture of the proposed WAE is shown in Figure 5.2. The WAE consists of a U-Net [187] like encoder-decoder network for image-to-image generation. The encoder part of the network consists of three layers of 3×3 convolutions layer followed by a Batch-Normalization layer. The discrete wavelet transformation (DWT) is used as an alternative to the pooling or down-sampling operation in the encoder part of the model.

For an input feature $f(x, y)$ with dimension $M \times N \times C$, the generalize DWT is described as,

$$f_{\psi}^{j}(x, y) = \frac{1}{\sqrt{M \times N}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)\psi^{j}(x, y) \tag{5.8}$$

where $j\epsilon\{A, V, H, D\}$, and M, N, and C are the height, width, and channel of the feature, respectively. f_{ψ} is the wavelet transformed output with dimension $M/2 \times N/2 \times C$, and $\psi(x, y)$ is called a scaling function. All four scaling functions are for the four components A, V, H, D. In this chapter, the Haar wavelet is used as the scaling function, because of its simplicity and efficient calculations using only addition and subtraction. Haar wavelet is well-suited for detecting sharp transitions or edges in an image. When noise is added to an image, the Haar wavelet can capture the noise at scales, where it appears as high-frequency variations. The four scaling functions are given as,

$$\psi^{A}(x, y) = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \psi^{V}(x, y) = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}, \psi^{H}(x, y) = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}, \psi^{D}(x, y) = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$$

The traditional pooling layer uses operations like estimating maximum, minimum, or average of a set of values, which sometimes suppress the input feature information. On the other hand, the proposed DWT-based down-sampling operation extracts the four frequency components from the input feature maps. The down-sampling operation preserves the edges and corners, etc. After each DWT operation, the number of output feature maps is doubled. Three DWTs are used in the encoder part of the network. No activation is used in this portion because the DWT adds non-linearity to the network. The DWT block serves a dual purpose, not only facilitating down-sampling but also contributing to the acquisition of multi-scale features, and the preservation of spatial resolution within the hidden features. After the third DWT, 5 DenseBlock layers are used. The architecture of the DenseBlock is shown in Figure 5.3. The DenseBlock operation is given as:

$$f_{DB}(x, y) = [f(x, y) \circledast W_1] \circledast W_2 + f(x, y) \tag{5.9}$$

Where, $f(x, y)$ is the input feature map of the DenseBlock, $\circledast$ is the convolution operation, W_1 and W_2 are the filters of the convolution layer. The number of filters in W_1 is half of the input channels and W_2 is the same as the input channels.

After each DenseBlock, Parametric Rectified Linear Unit (PReLU) activation is applied. The decoder network consists of three, 3×3 convolution layers, followed by an inverse discrete wavelet transform (IDWT) layer, which acts as an up-sampling layer. The traditional up-sampling operation cannot enhance the quality of features but IDWT enhances the quality of the features by transferring the features from the spatial domain to the frequency domain. The inverse wavelet transform is described as:

$$f'(x,y) = \frac{1}{\sqrt{M \times N}} \sum_{j \epsilon \{A,V,H,D\}} f_{\psi}^{j}(x,y)\psi^{j}(x,y)^{T} \tag{5.10}$$

where, $(.)^T$ is a transpose operation.After each IDWT operation, the convoluted feature from the encoder block is added. Utilizing 3×3 convolution layers followed by rectified linear unit (ReLU) activation, and combining the convoluted feature with the IDWT feature. This fusion is named Cross-Scale Fusion (CSF), denoted as,

$$f_{csf}(x,y) = ReLU(f'(x,y)) + f(x,y) \circledast W_{csf} \tag{5.11}$$

where, $f'(x,y)$ is the IDWT features, $ReLU(.)$ is the ReLU activation function and W_{csf} is the weight of the CSF layer. After each CSF layer, the number of feature maps is decreased to half of the previous layer. The network consists of a total of 31 convolution layers and 3 DWT and 3 IDWT layers. The network learns the haze and rain-related features in the tanning process. The dehaze or d-erain output is obtained by applying the input images to the network, which is denoted as:

$$O(x) = F_{\theta}(I(x)) \tag{5.12}$$

where, $O(x)$ is the output haze- or rain-free image, $I(x)$ is the input hazy or rainy image, $F_{\theta}(I(x))$ is the model output, and θ is a model parameters.

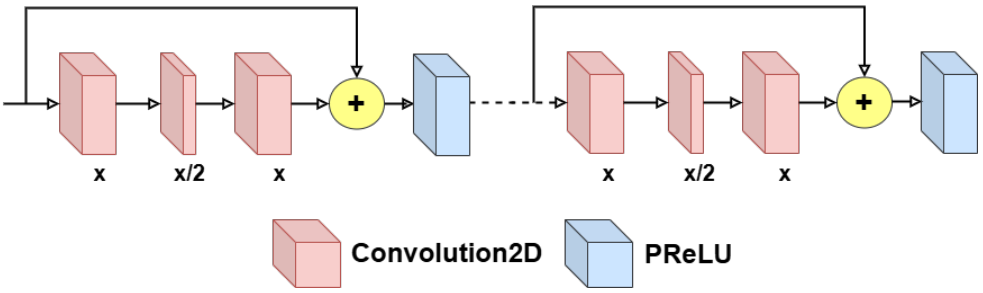

FIGURE 5.3: A visual representation of the DenseBlock layer introduced in the proposed model.

5.4.1 Loss Function

Mathematically, the proposed model learns the function $F_\theta(X|Y)$ in the training process, where, θ is the set of learnable parameters of the model, X is the input image, and Y is the corresponding ground truth. The most common image generation loss function is the Mean Square Error (MSE), which is defined as:

$$\mathcal{L}_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(O(x_i) - y_i)^2 \tag{5.13}$$

where x_i is the i-th input image and y_i is the corresponding ground truth. MSE is a useful loss function for image reconstruction. However, it is sensitive to outliers and over-penalizes the model during optimization, which causes a blur effect in the reconstructed image[22]. To reduce the blurring effect, the proposed model uses a combination of the other two losses namely, $\mathcal{L}_{MAE}$ and $\mathcal{L}_{SSIM}$, that are defined as,

$$\mathcal{L}_{MAE} = \frac{1}{N}\sum_{i=1}^{N}|O(x_i) - y_i| \tag{5.14}$$

$$\mathcal{L}_{SSIM} = -1 * SSIM(O(x_i), y_i) \tag{5.15}$$

where, $SSIM$ is the structural similarity index[142] of the reconstructed image and the ground truth. Mean absolute error (MAE) loss introduces a sparsity in the model and is less sensitive to the outliers. On the other hand, SSIM loss reduces the blurring effect. Thus the proposed model uses the total loss as follows:

$$\mathcal{L}_T = \alpha_1\mathcal{L}_{MSE} + \alpha_2\mathcal{L}_{MAE} + \alpha_3\mathcal{L}_{SSIM} \tag{5.16}$$

where, α_1, α_2, α_3 are the weight factors assigned to different loss functions. During

TABLE 5.1: Hyper-parameter details for training the proposed model WAE

Dataset	**Reside6k, Rain14000**	**RainKTTI2012,RainKITTI2015**	**JRSRD**
Parameter	**Value**	**Value**	**Value**
Image size	$256 \times 256 \times 3$	$256 \times 256 \times 3$	$256 \times 256 \times 3$
Optimizer	Adam	Adam	Adam
Initial learning rate	0.001	0.001	0.001
LR scheduler	ReduceLROnPlateau	ReduceLROnPlateau	ReduceLROnPlateau
Monitor metric	Validation Loss	Validation Loss	Validation Loss
Waiting time	5 epoch	3 epoch	5 epoch
Batch size	8	8	8
Number of epochs	200	50	100
Loss	$\mathcal{L}_T(\theta)$	$\mathcal{L}_T(\theta)$	$\mathcal{L}_T(\theta)$

training, the model parameters are optimized using the $\mathcal{L}_T(\theta)$ objective function with the back-propagation algorithm.

5.5 Experimentation

In this section, the implementation details, including the datasets utilized for training and testing the proposed model, are outlined. Additionally, the results obtained by the proposed model are presented.

5.5.1 Implementation Details

The proposed model, developed using PyTorch and trained on an NVIDIA Tesla P100 GPU, utilizes a batch size of 8 and an initial learning rate of 0.001. The Reside6K and Rain14000 datasets use 200 epochs to train the model, whereas the RainKITTI2012 and RainKITTI2015 datasets use 100 epochs and the JS-DRD dataset uses 50 epochs. To enhance the convergence and prevent local minima, a learning rate scheduler called ReduceLROnPlateau is employed, which dynamically adjusts the learning rate based on the validation loss. This technique reduces the learning rate if the validation loss does not improve for 5 epochs, thereby facilitating optimization with smaller steps. For the RainKITTI2012 and RainKITTI2015 datasets, the waiting time is set to 3 epochs. The Adam optimizer is utilized to minimize the objective function in Equation 5.16, where the values of α_1, α_2, α_3 are set to 0.25, 0.25, and 0.5, respectively. Additionally, the input images and output images are chosen to have a size of $256 \times 256 \times 3$ during training. All hyper-parameters used in the experiments are shown in Table 5.1.

5.5.2 Datasets

This chapter proposes an approach to simultaneously remove two types of weather-related image degradations, namely rain and haze. To this end, eleven distinct datasets have been used for experimentation.

Firstly, synthetic rain dataset is used, which comprises Rain14000[37] and Rain800[38]. These datasets have been used for training the proposed model to remove the

rain effect from images. Two datasets are chosen because they are some of the most widely used in the field of rain removal, and they represent different levels of complexity and realism. Specifically, Rain800 contains 800 synthetic rainy images, which include 100 test images called Rain100L, while Rain14000 includes 14,000 training images with ground truth information. Since the Rain14000 dataset does not come with a separate test set, the Rain1200[34] dataset is used for evaluating the performance of the trained model. RainKITTI2012[35] and RainKITTI2015[35] datasets are used to evaluate the model and provide information to show how well it performs on a variety of stereo-image captures in real-world scenarios. Furthermore, the JRSRD[36] dataset, which includes artificial rain streaks and raindrop simulations, is used for the evaluation of combined rain streaks and raindrop removal. Additionally, assessments are carried out utilizing the RIS[39] and RID[39] datasets to see how well the method addresses real-world haze circumstances.

In the case of haze removal, the Reside-6K[23] dataset is used for training the model. This dataset is one of the largest and most diverse datasets for haze removal, with over 6,000 images of indoor and outdoor scenes containing varying degrees of haze. It is believed that training the model on this dataset would allow it to generalize better to a wide range of real-world scenarios. Finally, the performance of the trained models is evaluated using two separate test datasets: Reside-indoor (ITS)[23] and Reside-outdoor (OTS)[23] for haze removal, which contains 500 images each. To assess the real-haze scenarios, the proposed approach is also evaluated using the real-haze datasets proposed by Fattal R.[188]. The model has been trained on a mixed set of images from Reside-6K[23], Rain14000[37], and Rain800[38] datasets.

5.6 Results

The proposed model is compared with state-of-the-art dehazing and derain techniques that include LMSFAN[189], VQDD[190], MSBDN[191], AODNet[192], DCP [193], SDA-GAN[194], HTFA[195], SNSPGAN[196], LIGHT-Net[197], YOLY[198], DDAP[199], USID-Net[200], D4[201], EPDN[202], FSAD-Net[203], FD-GAN[204] for dehazing results on ITS, OTS and Reside-6K datasets. Fattal's dataset[188] has been utilized to evaluate the real-haze dataset, and compared proposed model with the following algorithms: VC-SEM[205], JCE-EF[206], MSAFF-Net[207],

Dehaze-AGGAN[208], SGLC[209], SCANet[210].The Natural Image Quality Evaluator (NIQE), a non-reference image quality evaluation metric, has been utilized, as there are no ground truths available for this dataset. Similarly, derain results on Rain100L and Rain1200 datasets have been compared with DerainNet[211], SEMI[212], DIDMND[34], UMRL[213], PreNet[214], GMM[215], UGSM[216], DDN [37], GCA-Net[217], JORDER[218], DSC[219], CNN[220], RESCAN [221], RE-HEN [222], SSIR[223], RSGN[224] models. Stereo single-image deraining datasets, namely RainKITTI2012 and RainKITTI2015, have been used to evaluate the performance of the proposed model in various real-world outdoor scenes. It compares with existing algorithms, including DSC[219], GMM[215], DDN[37], RESCAN[221], PReNet[214], Uformer_B[225], DRCDNet[226], CDINet[227], Restormer[228], and ESTINet[229]. Raindrops and rain streaks are not the same thing. Rain streaks, which can vary in length, thickness, and direction, are the patterns that resemble lines that show up in images when rain is present. However, the individual water droplets that create rain streaks in images are known as raindrops. Using the JRSRD dataset, the model's performance has been assessed in both scenarios by contrasting it with that of other image-deraining algorithms, including RESCAN[221], PReNet[214], Qian et al.[230], the combination of Qian et al.[230] and PReNet[214], DAiAM[36], Uformer_B[225], and ESTINet[229]. RID and RIS datasets are compared with RESCAN[221], UMRL[231], VC-SEM[205], PRENet[214], MSPFN[232], MPRNet[233], DGUNet[234], and SCIUNet[235] in the evaluation experiments based on real-rain. Since there is no ground truth available, the NIQE non-reference metric is used to evaluate the real haze and rain dataset. All evaluations of synthetic hazy and rainy datasets are measured using SSIM and PSNR metrics.

5.6.1 Qualitative Evaluation

The qualitative outcomes of the proposed model are depicted in Figure 5.4, Figure 5.6, Figure 5.5, and Figure 5.7. To evaluate the dehazing capability of the proposed model, a comparative analysis has been conducted with five existing methods, namely ADO-Net[192], GCA-Net[217], D4[201], FSAD-Net[203], and FD-GAN[204]. These models have been assessed using both the Indoor Testing Set (ITS) and the Outdoor Testing Set (OTS), as illustrated in Figure 5.4. The analysis reveals that while ADO-Net produces darker results in outdoor images, it inadequately removes haze in indoor images. In contrast, GCA-Net, FD-GAN,

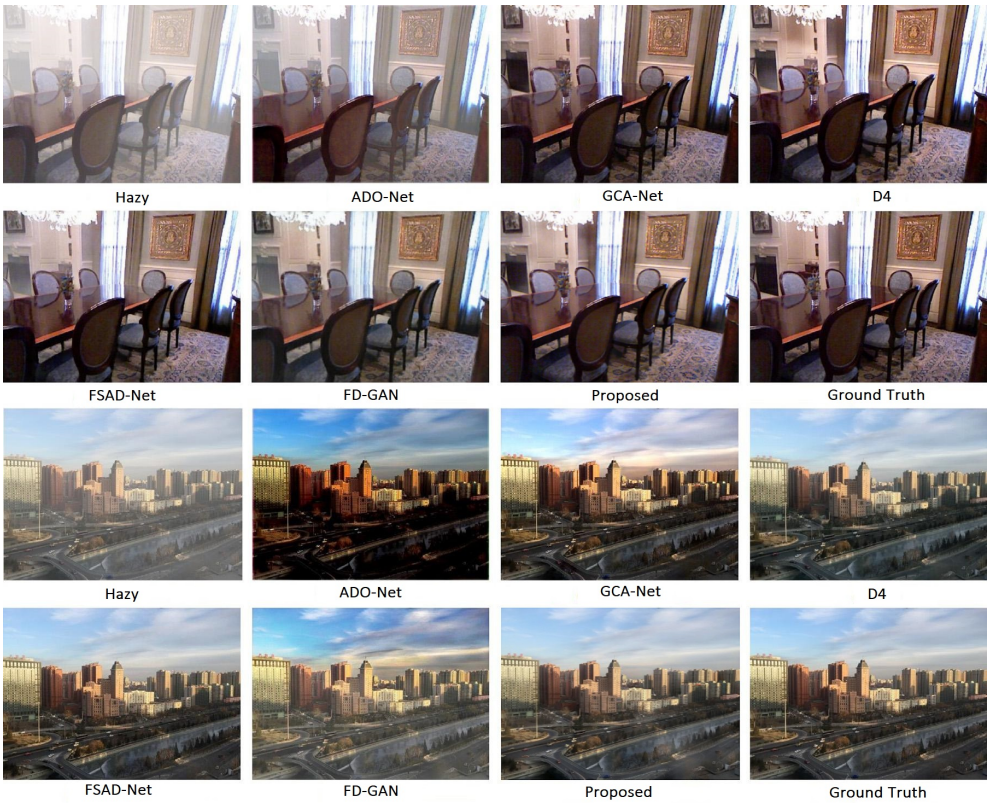

FIGURE 5.4: Qualitative comparison of the proposed model and some existing models on Reside6K-ITS (indoor) and Reside6K-ITS (outdoor) datasets. The top two rows are from the Reside6K-ITS (indoor) testing set, and the bottom two rows are from the Reside6K-ITS (outdoor) testing set.

TABLE 5.2: Quantitative comparison of the proposed model on ITS, OTS, and RESID6K datasets. The **bold** texts represent the best results.

Method	ITS		OTS		Reside6K				
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	Time(s)	Param(M)	Flops(G)
WAE (Proposed)	24.73	**0.940**	**26.95**	**0.955**	25.84	**0.949**	0.36	12.5	**0.08**
LMSFAN[189]	-	-	24.76	0.933	-	-	0.56	0.012	53.70
VQDD[190]	22.25	0.847	22.53	0.875	22.39	0.861	-	0.23	63.85
MSBDN[191]	**25.79**	0.871	26.78	0.926	**26.28**	0.899	0.134	31.35	41.54
AODNet[192]	20.51	0.816	24.14	0.920	22.33	0.868	**0.004**	0.008	0.39
DCP[193]	16.61	0.855	19.14	0.861	17.88	0.858	1.17	-	0.6
SDA-GAN[194]	18.08	0.776	19.26	0.813	18.67	0.795	-	19.96	72.19
HTFA[195]	17.98	0.695	17.45	0.704	17.72	0.699	-	1.57	72.19
SNSPGAN[196]	17.75	0.788	24.28	0.925	21.02	0.857	-	-	-
EPDN[202]	25.09	0.932	20.31	0.902	22.7	0.917	0.23	17.38	5.31
LIGHT-Net[197]	23.11	0.917	22.27	0.906	22.69	0.912	-	-	-
YOLY[198]	20.39	0.889	21.02	0.905	20.71	0.897	40.56	39.44	68.51
DDAP[199]	17.94	0.829	17.06	0.840	17.50	0.835	-	-	-
USID-Net[200]	17.45	0.769	23.86	0.888	20.65	0.828	0.014	3.77	160
D4[201]	23.47	0.898	25.62	0.939	24.55	0.918	0.432	10.70	24.7
FSAD-Net[203]	22.61	0.910	23.15	0.921	22.88	22.88	0.254	6.86	-
FD-GAN[204]	23.47	0.874	25.27	0.939	24.37	0.906	0.514	12.95	-

and FSAD-Net effectively mitigate haze in outdoor images but tend to introduce over-saturation. For indoor images, FSAD-Net and GCA-Net demonstrate superior performance compared to FD-GAN. Notably, the proposed WAE model and

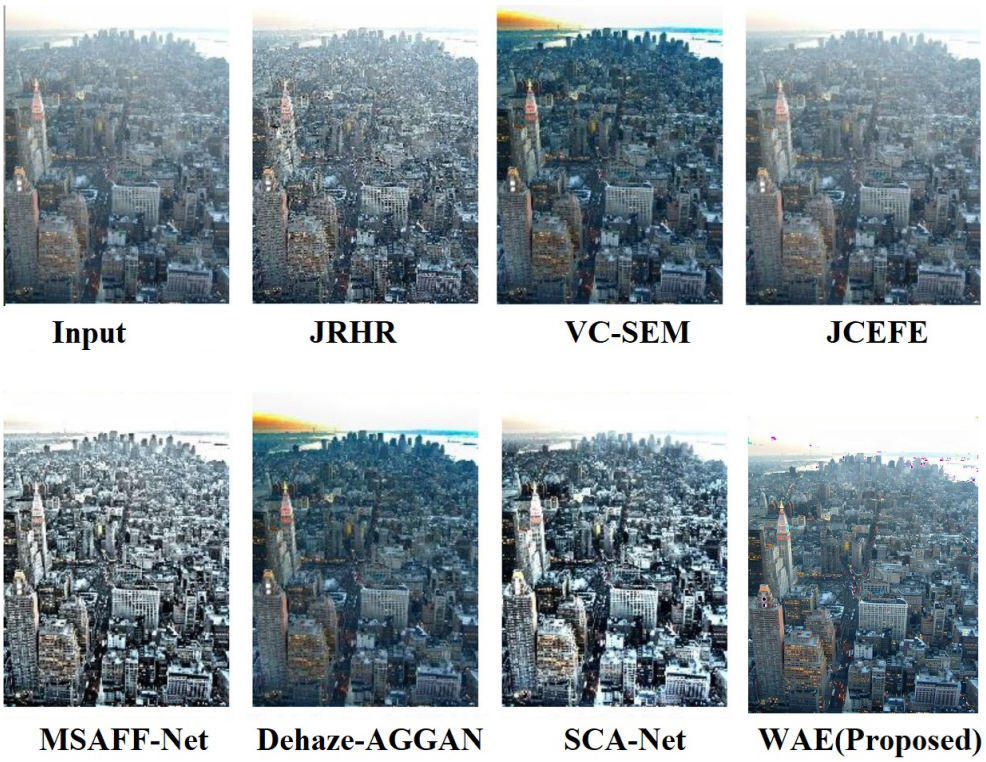

FIGURE 5.5: Qualitative comparison of the proposed model and some existing models on the Fattal's real-haze dataset.

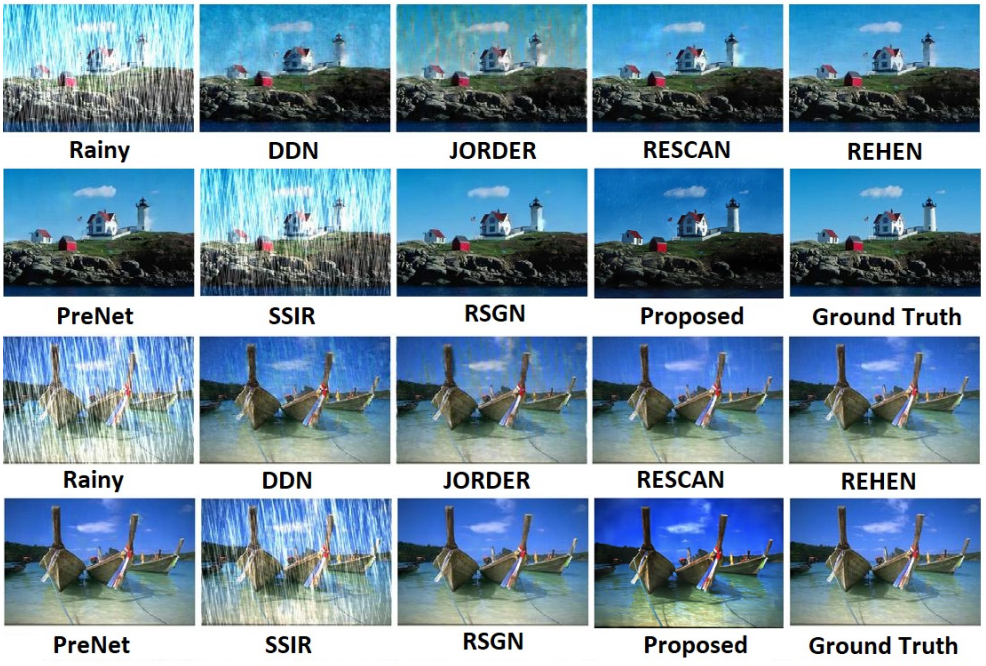

FIGURE 5.6: Qualitative comparison of the proposed model and some existing models on the Rain100L dataset.

the D4 model yield more realistic dehazing results under both indoor and outdoor conditions.

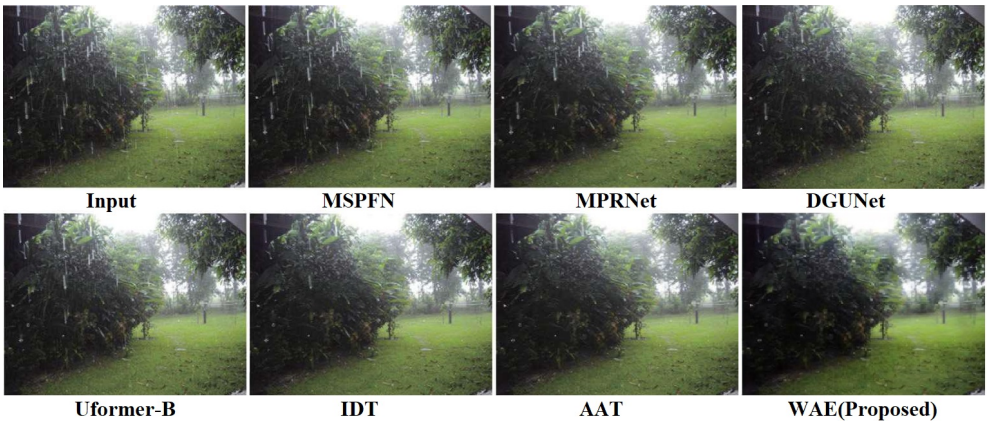

FIGURE 5.7: Qualitative comparison of the proposed model and some existing models on the real-rain dataset.

Moving on to the comparison of rain removal techniques, the evaluation encompasses DDN[37], JORDER[218], RESCAN[221], REHEN[222], PreNet[214], SSIR [223], and RSGN[224]. The outcomes derived from these methodologies, as applied to the Rain100L dataset, are depicted in Figure 5.6. An observation of the results reveals that the SSIR model struggles to effectively eliminate rain streaks, leading to a substantial presence of artifacts. JORDER, on the other hand, yields outputs marred by the presence of artifacts, thus reducing visual clarity. Contrarily, both RESCAN and DDN produce outputs with a discernible blurriness. REHEN outperforms RESCAN, delivering enhanced outcomes, yet it does not fully mitigate the blurriness. Notably, PreNet and RSGN yield satisfactory results, albeit with slight saturation imperfections in the first image from Figure 5.6. In comparison, the proposed model demonstrates superior performance, showcasing its efficacy in rain removal.

Figure 5.5 shows a comparative analysis of real haze-removal performance using JRHR[236], VC-SEM[205], JCE-EF[206], MSAFF-Net[207], Dehaze-AGGAN[208], SGLC[209], and SCANet[210] . Notably, in this comparison, the suggested technique performs better. Additionally, as shown in Figure 5.7, MSPFN[232], MPRNet [233], DGUNet[234], Uformer_B[225], AAT[237] and IDT[238] has been used to evaluate the model's performance in actual rain conditions. AAT shows improved rain removal performance, but with some lingering haziness, which is not found using the proposed model.

TABLE 5.3: Quantitative comparison of the proposed model on the Rain100L dataset. The **bold** values represent the best results.

Method	PSNR	SSIM	Time(s)	Param
WAE (Proposed)	**24.81**	**0.858**	0.36	12.6M
DerainNet[211]	22.77	0.810	0.55	0.8M
SEMI[212]	22.35	0.788	0.47	58.5K
DIDMND[34]	22.56	0.818	0.97	371.2M
UMRL[213]	24.41	0.825	**0.063**	0.98M
PreNet[214]	24.81	0.851	0.77	171.2K
GMM[215]	22.03	0.711	43.74	-
UGSM[216]	22.97	0.747	-	-
DDN[37]	22.02	0.742	0.34	**58.1K**
GCA-Net[217]	22.28	0.749	27.56	0.7M
JORDER[218]	22.18	0.757	1.74	4171.2K
SSIR[223]	22.47	0.7164	20.11	58.6K

TABLE 5.4: Quantitative comparison of the proposed model on the Rain1200 dataset. The **bold** text represents the best result.

Method	PSNR	SSIM	Time(s)	Param
WAE (Proposed)	**26.06**	**0.92**	**0.36**	12.6M
DerainNet[211]	23.38	0.835	0.55	0.8M
SEMI[212]	26.05	0.822	0.47	**58.5K**
DSC[219]	21.44	0.7896	99.17	-
GMM[215]	22.75	0.8352	43.74	-
CNN[220]	22.07	0.8422	24.00	-
JORDER[218]	24.32	0.8622	1.74	4171.2K

5.6.2 Quantitative Evaluation

The quantitative assessment of the proposed model is presented across multiple tables, specifically in Table 5.2, Table 5.3, Table 5.4, Table 5.5 Table 5.6, and Table 5.7. These tables serve as the backdrop for evaluating the model's performance using essential metrics, including the PSNR and SSIM. The real haze and rain removal performance is evaluated using the NIQE metric, which is shown in Table 5.8. The model's performance is also compared based on the number of FLOPs, parameter counts, and inference time per image.

In Table 5.2, the proposed model, trained on the Reside6K dataset, undergoes scrutiny on the ITS and OTS datasets. The outcomes underscore the model's prowess, with the MSBDN variant excelling in indoor hazy image processing, while the proposed model consistently outperforms competitors in terms of both PSNR and SSIM metrics on outdoor hazy images and the Reside6K dataset.

TABLE 5.5: Quantitative comparison of the proposed model on the Rain100L, Rain1200, ITS and OTS datasets trained on the combination of Reside6K and Rain1400 datasets.

Metric	Dataset			
	Rain100L	**Rain1200**	**ITS**	**OTS**
PSNR	23.658	26.372	20.587	26.69
SSIM	0.8365	0.8467	0.878	0.955

TABLE 5.6: Quantitative comparison of the proposed model on RainKITTI2012 and RainKITTI2015 datasets. The **bold** text represents the best result.

Method	RainKITTI2012		RainKITTI2015		Time(s)	Parameters(M)
	PSNR	**SSIM**	**PSNR**	**SSIM**		
DSC[219]	18.88	0.677	19.35	0.686	99.2	-
GMM[215]	18.54	0.673	20.39	0.692	371.2	-
DDN[37]	29.43	0.904	29.23	0.906	0.6	0.06
RESCAN[221]	34.09	0.955	35.28	0.950	0.6	**0.05**
PReNet[214]	34.12	0.933	35.27	0.949	0.10	0.17
Uformer_B[225]	30.95	0.946	32.69	0.947	3.3	5.29
DRCDNet[226]	31.91	0.934	32.10	0.931	0.173	2.25
Restormer[228]	33.79	0.947	34.00	0.944	0.439	26.13
CDINet[227]	34.67	0.953	33.92	0.947	**0.096**	10.63
ESTINet[229]	34.13	0.947	33.83	0.943	0.3	0.43
WAE (Proposed)	**36.84**	**0.972**	**36.88**	**0.956**	0.36	12.6

Moving on to Table 5.3 and Table 5.4, these tables offer insights into the model's performance after training on Rain800 and Rain14000, respectively. The evaluation extends to the Rain100L and Rain1200 datasets, revealing the model's unwavering superiority, as it consistently secures the top-ranking performance across both datasets. The RainKITTI2012 and RainKITTI2015 datasets have been utilized to evaluate the method's generalization capabilities over a variety of situations collected in stereo images. The suggested model performs better on the RainKITTI2012 and RainKITTI2015 datasets, as seen in Table 5.6. Furthermore, Table 5.7 provides an assessment of raindrops and streaks using the JRSRD dataset, demonstrating that the proposed model performs best in terms of SSIM and PSNR. Although RESCAN performed well in terms of run-time and number of parameters, the suggested model demonstrates reduced FLOPs, highlighting its cheaper computational cost.

Table 5.8 presents the model's performance in authentic rain and haze scenarios. Real rain assessment utilizes the RID and RIS datasets, while real haze evaluation employs Fattal's dataset. Across all scenarios, the proposed model consistently demonstrates superior performance, as evidenced by the NIQE metric.

TABLE 5.7: Quantitative comparison of the proposed model on the JRSRD dataset. The **bold** text represents the best result.

Method	PSNR	SSIM	Time	Flops(G)	Param(M)
RESCAN[221]	21.05	0.768	**0.07**	246	**0.15**
PReNet[214]	23.29	0.789	0.10	337	0.17
Qian et al.[230]	22.49	0.772	0.13	683	6.00
Qian et al.[230] + **PReNet**[214]	23.89	0.769	0.23	102	6.17
PReNet[214] + **Qian et al.**[230]	23.68	0.793	0.23	102	6.17
DAiAM[36]	24.67	0.819	0.13	890	3.60
D-DAiAM[36]	25.26	0.825	0.28	178	7.20
Uformer_B[225]	27.56	0.873	0.28	178	7.20
ESTINet[229]	27.52	0.868	0.28	178	7.20
WAE (Proposed)	**28.08**	**0.899**	0.36	**0.08**	12.6

Furthermore, Table 5.5 aggregates results from the proposed model's evaluation on a comprehensive dataset combining Rain14000, Rain100, and Reside6K. These tables provide a holistic view of the model's performance, showcasing its effectiveness in enhancing image dehazing and deraining quality across diverse scenarios and datasets. These findings validate the model's robustness and its capacity to address various challenging conditions in image enhancement tasks.

5.6.3 Ablation Studies

5.6.3.1 Study on Loss

In this ablation study, a combination of L1, MSE, and SSIM loss functions is employed, with the final loss function described in Equation 5.16. The constant values α_1, α_2, and α_3 are set to 0.1, 0.1, and 0.2, respectively. For a deeper understanding of the proposed model's performance under different loss configurations, experiments utilizing various loss types have been conducted. The results are presented in Table 5.9, which showcases the PSNR and SSIM values for models trained with different loss combinations, including L1, MSE, SSIM, a combination of L1 and SSIM, L1 and MSE, MSE and SSIM, L1, MSE, and SSIM, as well as the proposed loss, on both the Rain100L and OTS datasets.

On the Rain100L dataset, the L1 loss yields noticeable performance improvements. Conversely, on the OTS dataset, the combination of L1, MSE, and SSIM with weights $\alpha_1 = 0.2$, $\alpha_2 = 0.2$, and $\alpha_3 = 0.1$ achieves favorable results. However, the employment of the proposed loss function yields the best results across both datasets.

TABLE 5.8: Quantitative comparison of the proposed model on the real rain datasets RIS and RID, and real haze dataset proposed by Fattal R. using NIQE. The **bold** text represents the best result.

Method	Real Rain		Method	Real Haze
	RIS	RID		Fattal R. [188]
RESCAN[221]	6.485	6.641	**JRHR**[236]	3.8999
UMRL[231]	5.615	6.757	**VC-SEM**[205]	4.5242
PRENet[214]	6.722	7.007	**JCE-EF**[206]	4.9927
MSPFN[232]	6.135	6.518	**MSAFF-Net**[207]	5.8286
MPRNet[233]	4.710	4.487	**Dehaze-AGGAN**[208]	5.8825
DGUNet[234]	4.581	4.587	**SGLC**[209]	5.9726
SCIUNet[235]	4.293	4.354	**SCANet**[210]	6.4133
WAE(Proposed)	**4.255**	**4.301**	**WAE(Proposed)**	**3.7853**

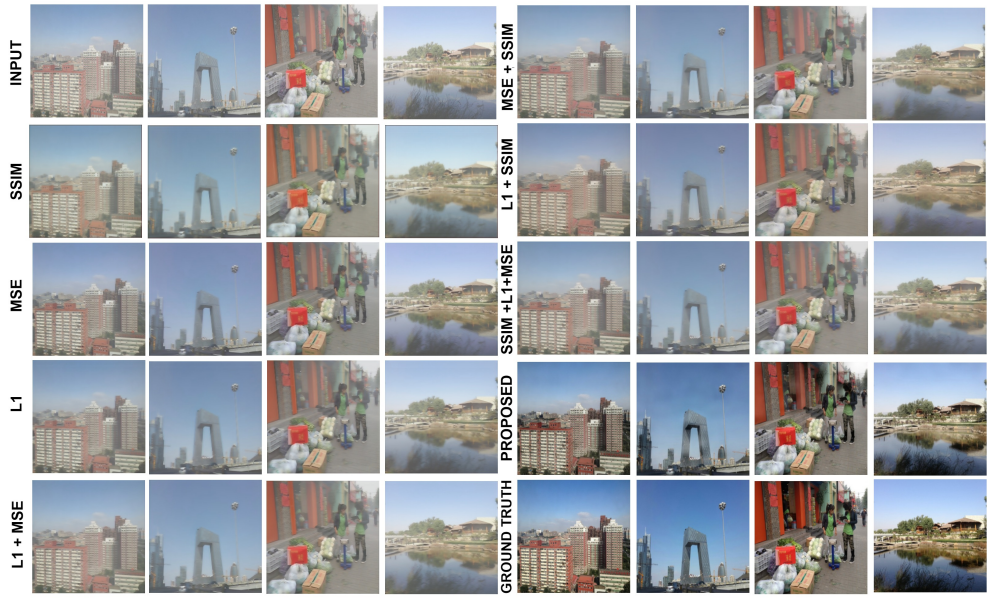

FIGURE 5.8: Assessment of image quality across diverse training configurations of the proposed model, featuring a range of loss functions including L1, MSE, SSIM, SSIM+L1, SSIM+MSE, L1+MSE, L1+MSE+SSIM (with equal weight), and the proposed loss function. The evaluation is performed on the OTS dataset.

Figure 5.8 and Figure 5.9 present qualitative results obtained from models trained using various loss functions and tested on hazy and rainy images, respectively. Examining Figure 5.9, it becomes evident that the proposed loss function achieves the best performance, effectively removing rain artifacts, while other loss functions fail to do so satisfactorily. In contrast, for hazy images shown in Figure 5.8, models trained with alternative loss functions produce similar outcomes, exhibiting comparable results. However, the model trained with the proposed loss function excels in generating superior dehazing outputs.

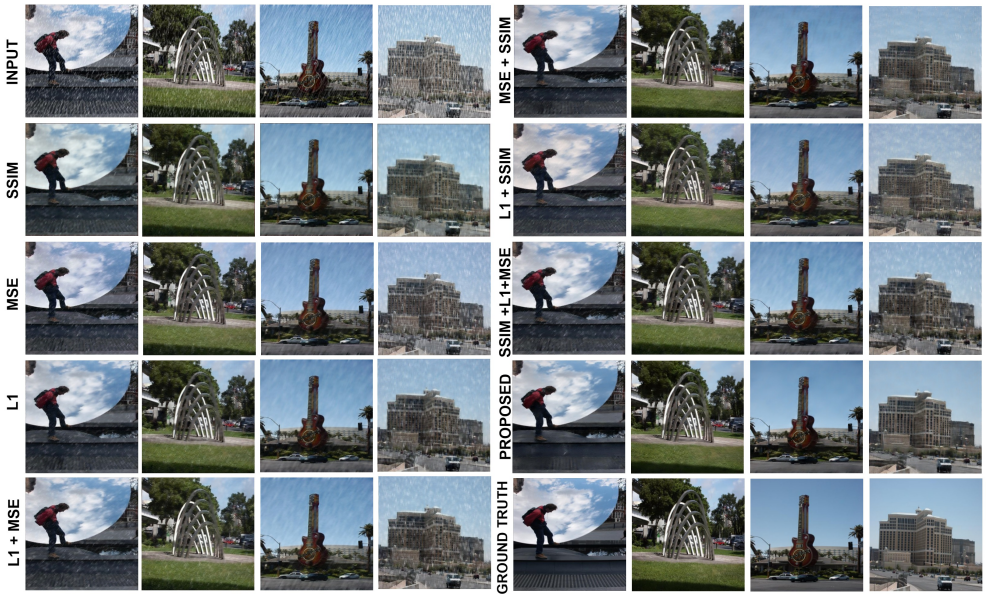

FIGURE 5.9: Assessment of image quality across diverse training configurations of the proposed model, featuring a range of loss functions including L1, MSE, SSIM, SSIM+L1, SSIM+MSE, L1+MSE, L1+MSE+SSIM (with equal weight), and the proposed loss. The evaluation is performed on the Rain100L dataset.

TABLE 5.9: Comparison of SSIM and PSNR metrics for the proposed model trained with various loss functions, including L1, MSE, SSIM, SSIM+L1, SSIM+MSE, L1+MSE, L1+MSE+SSIM (with equal weight), and the proposed loss, on the OTS and Rain100L datasets. The last row illustrates the results achieved by the proposed model incorporating max-pooling in conjunction with the proposed loss. The **bold** text represents the best result.

Model	Loss	Rain100L		OTS	
		PSNR	SSIM	PSNR	SSIM
Proposed model with Wavelet	L1	25.41	0.846	18.32	0.760
	MSE	25.01	0.843	18.51	0.792
	SSIM	24.09	0.826	17.40	0.730
	L1+MSE	25.18	0.847	18.18	0.779
	L1+SSIM	25.03	0.844	18.34	0.759
	MSE+SSIM	25.30	0.853	18.32	0.767
	L1+MSE+SSIM	25.10	0.843	18.55	0.773
	Proposed	**26.06**	**0.920**	**26.69**	**0.955**
Proposed model with MaxPool	Proposed	24.41	0.835	18.94	0.782

5.6.3.2 Study on Wavelet vs Pooling

This chapter, introduces a novel approach that leverages discrete wavelet transform as an alternative to traditional pooling techniques.The investigation centers on a comparative analysis between the proposed model employing standard max-pooling and the model incorporating wavelet operations. The final row in Table 5.9 provides a comprehensive view of the PSNR and SSIM scores for the proposed

TABLE 5.10: Comparison of the proposed model in terms of PSNR and SSIM using different numbers of filters on OTS and Rain100L datasets. The last column shows the processing time. The **bold** text represents the best results.

Dataset	OTS		Rain100L		Time (Sec)
No. of filters	PSNR	SSIM	PSNR	SSIM	
16	18.82	0.796	25.49	0.844	0.297
32	21.80	0.830	25.65	0.873	0.368
64	**26.69**	**0.955**	**26.06**	**0.920**	**0.343**
128	22.68	0.816	24.93	0.853	0.414
256	23.31	0.827	25.31	0.849	0.620

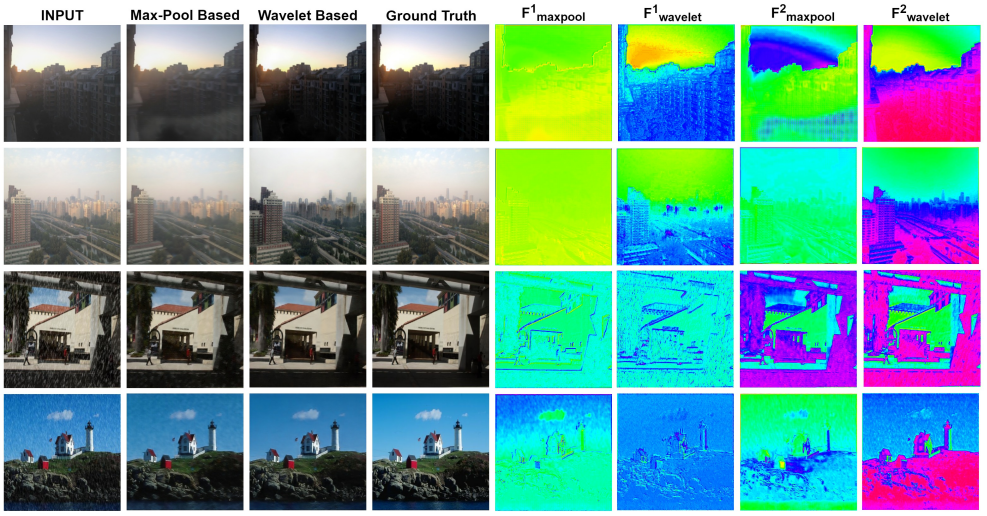

FIGURE 5.10: Assessment of image qualities in various training configurations of the proposed model, including max-pooling and wavelet approaches. The last four columns display features extracted by Wavelet-based and max-pooling-based models. The evaluation is carried out on the Rain100L and OTS datasets.

TABLE 5.11: Comparison of the proposed model in terms of PSNR and SSIM using different numbers of DenseBlocks on OTS and Rain100L datasets. The last column shows the processing time. The **bold** text represents the best results.

Dataset	OTS		Rain100L		Time (Sec)
No. of DenseBlocks	PSNR	SSIM	PSNR	SSIM	
2	21.66	0.868	26.01	0.875	**0.297**
5	26.69	0.955	26.06	0.920	0.343
8	**26.90**	**0.960**	**26.16**	**0.925**	0.369

model utilizing max-pooling layers trained with the proposed loss function. It is evident from these results that the wavelet-based model consistently outperforms its max-pooling counterpart on both datasets. Furthermore, Figure 5.10 offers a qualitative comparison, revealing that the max-pooling-based model generates blurry outputs and struggles to effectively remove rain and haze artifacts. In stark

contrast, the proposed wavelet-based method delivers markedly superior results.

Additionally, two features were extracted from the final layer of the proposed model, as depicted on the right side of Figure 5.10. A careful examination of Figure 5.10 highlights that the wavelet-based features contain richer information pertaining to rain and haze compared to max-pooling. These wavelet-based features exhibit enhanced texture and edge information, whereas the max-pooling-based features, characterized by reduced edge information, contribute to the generation of blurry output.

5.6.3.3 Study on the Number of Filters

The proposed model is composed of fully convolutional layers. Within the encoder portion of the network, each layer contains twice the number of filters as the preceding layer, while in the decoder section, each layer comprises half the number of filters of the previous layer. This investigation aims to assess the model's performance under varying filter counts.

The results, as detailed in Table 5.10, present a comparative analysis of PSNR, SSIM, and processing time across models employing different filter quantities. Specifically, multiplications of 16, 32, 64, 128, and 256 filters are considered. Notably, the proposed model attains optimal PSNR and SSIM scores when employing 64 filters. In terms of processing time, as the number of filters increases, the computational time required by the proposed model also rises. In the case of the proposed model employing 64 filters, the processing time is recorded at 0.343 secs, shedding light on the trade-off between the filter count and the computational efficiency.

5.6.3.4 Study on the Dense Layer

Within the architecture of the proposed model, a pivotal component is the dense layer, which serves as the bottleneck within the auto-encoder framework. In this ablation study, the impact of varying the number of dense blocks within the proposed model is investigated.

As detailed in Table 5.11, findings reveal an interesting relationship: as the number of dense blocks increases, the model's efficacy in removing haze and rain artifacts

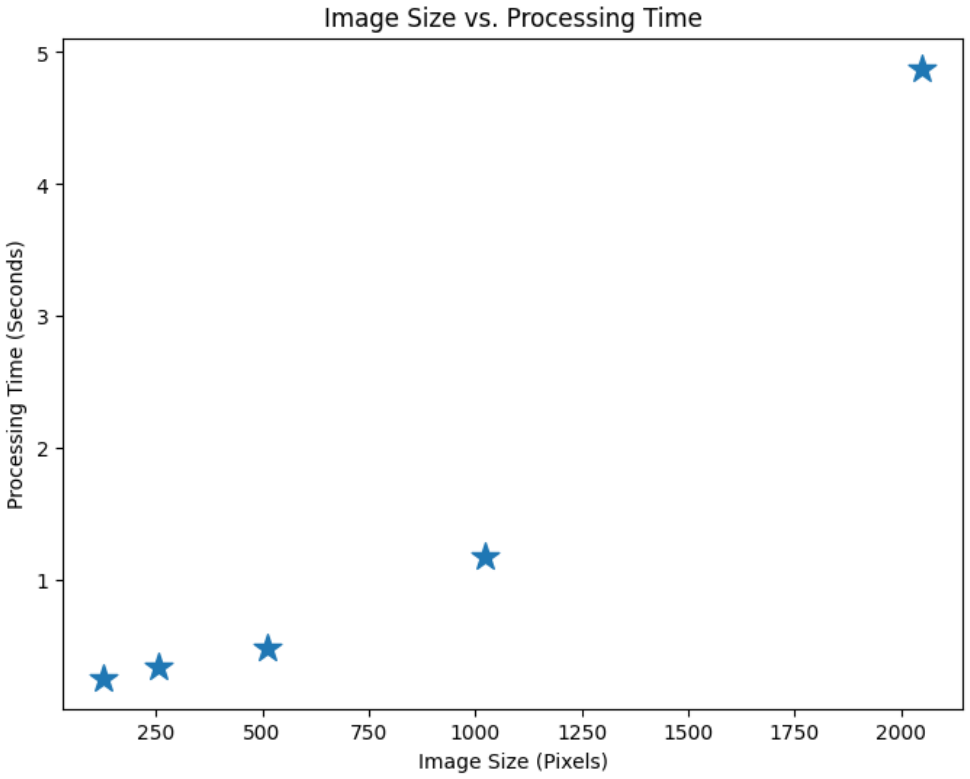

FIGURE 5.11: Processing time of the proposed model when the image size varies

also improves. However, it is crucial to note that this improvement comes at the cost of increased processing time as the number of dense blocks grows. This trade-off between enhanced performance and heightened computational requirements highlights the significance of optimizing the choice of dense blocks in the model's architecture.

5.6.3.5 Study on the Image Size

Given the fully connected nature of the proposed model, its adaptability to images of various sizes is a crucial aspect to consider. Understanding the processing time associated with different image dimensions is of paramount importance. Figure 6.8 presents a graphical representation of the relationship between the image size and processing time. A noteworthy observation from Figure 6.8 is the exponential increase in processing time as the image size grows. This insight underscores the significance of efficiently managing computational resources when dealing with diverse image scales.

5.7 Discussion

This chapter introduces a novel auto-encoder model, WAE, which represents a significant advancement in the realm of image enhancement. The primary focus has been on addressing the intricate challenges posed by rain streaks and haze in images, both of which are common nuisances in computer vision applications. The distinctive feature of the WAE architecture lies in its judicious use of wavelet transformations at various stages of the encoder, effectively serving as a robust pooling mechanism. This approach enables the model to capture intricate spatial and frequency information essential for accurate image restoration. Whereas, the decoder stage employs inverse-wavelet transformations, facilitating seamless up-sampling and image reconstruction.

The comprehensive evaluation of the proposed model on benchmark datasets, including Rain100L, Rain1200, ITS, OTS, Reside6K, Fattal's real-haze dataset, Rain-KITTI2012, Rain-KITTI2015, JRSRD, RID and RIS datasets, has yielded noteworthy results. The proposed model has demonstrated its efficacy in significantly improving image quality by removing rain streaks and haze, enhancing visual appeal, and contributing to better computer vision tasks.

Chapter 6

Object Tracking in Adverse Weather: A Dataset and a Model

6.1 Introduction

Visual object tracking poses significant challenges in computer vision, as it involves estimating the target's location by leveraging temporal information from its initial state. This task becomes notably arduous in adverse weather conditions, such as haze and rain, where visibility is severely reduced. Despite the emergence of numerous real-time object-tracking algorithms in recent years, there has been a glaring lack of research addressing tracking in adverse weather climates.

Given the applicability of object-tracking algorithms in security and surveillance, robust performance in adverse weather conditions is imperative for various real-world systems, including self-driving vehicles, automated retail, and visual surveillance. In adverse weather scenarios like heavy rain or haze, conventional tracking systems may fail due to information loss or alterations in color and texture induced by the weather, leading to erroneous tracking.

The significant advancements in object tracking owe much to the development of CNN-based deep learning models and the establishment of large-scale object tracking datasets and benchmarks such as OTB, GOT-10K, LaSOT, and VOT. These datasets have facilitated the evaluation of various tracking aspects such as color information impact, fast target motion, occlusion, and background clutter. However, they lack the provision for assessing tracker performance in challenging scenarios like adverse weather conditions.

In real-world situations characterized by rain and haze, trackers must exhibit robustness in handling adverse visibility conditions. Therefore, a critical need arises for a new benchmark dataset and baseline model for evaluating object tracking performance in adverse climate conditions. This chapter introduced a new dataset and a model to address the real-life issue.

A. Ali, and S. S. Chaudhuri, "**Video-Haze100 Dataset**". Diary No. 18933/2023-CO/L. ROC Number: L-8804/2023. Registered

A. Ali, D. Hossain, S. Sk, and S. S. Chaudhuri, "**Video-Rain99 Dataset**". Diary No. 8801/2023-CO/L. ROC Number: L-136249/2023. Registered

A. Ali, and S. S. Chaudhuri, "**A Feature Representation Technique For Angular Margin Loss**".Innovations in Computational Intelligence and Computer Vision. ICICV 2022. Lecture Notes in Networks and Systems, vol 680. Springer, Singapore. https://doi.org/10.1007/978-981-99-2602-2_7.

6.2 Contributions

This chapter introduces a new dataset, *ExtremeTrack*, and an unsupervised object tracking algorithm, *ArcTrack*, designed to evaluate model performance under adverse weather conditions. The key contributions are outlined as follows:

- A new synthetic dataset, *Extreme Weather Tracking Dataset* (*ExtremeTrack dataset*), is presented for visual object tracking under challenging weather conditions. The dataset includes 199 videos, comprising approximately 95,000 frames, with 100 videos depicting hazy conditions and 99 videos illustrating rainy conditions. The dataset is divided into 159 videos for training and 40 videos for testing.
- A novel zero-shot model is introduced called *ArcTrack*, which comprises a rain and haze removal model, a zero-shot object segmentation model, a new future similarity method, and the Kalman Filter.
- A novel feature representation method is proposed to improve similarity measurements between the template and detected objects, enabling robust tracking in visually degraded environments.
- The proposed *ArcTrack* algorithm integrates an unsupervised detection mechanism to improve usability and adaptability in real-world scenarios, making it a robust solution for adverse weather conditions.
- The *ArcTrack* model employs an ArcLoss-based similarity matching technique to enhance the robustness of object tracking under adverse weather conditions, addressing challenges such as occlusion and motion blur.

6.3 Proposed Method

This section introduces the *ExtremeTrack* dataset and a zero-shot object-tracking algorithm, called *ArcTrack*, specifically designed for adverse weather conditions.

TABLE 6.1: Comparison of *ExtremeTrack* with existing benchmark datasets for object tracking

Dataset	Videos	Min frames	Mean frames	Median frames	Max frames	Total frames	Frame rate
OTB-2013 [21]	51	71	578	392	3,872	29K	30 fps
OTB-2015 [21]	100	71	590	393	3,872	59K	30 fps
TC-128 [239]	128	71	429	365	3,872	55K	30 fps
VOT-2014 [240]	25	164	409	307	1,210	10K	30 fps
VOT-2017 [241]	60	41	356	293	1,500	21K	30 fps
NUS-PRO [242]	365	146	371	300	5,040	135K	30 fps
UAV123 [243]	123	109	915	882	3,085	113K	30 fps
UAV20L [243]	20	1,717	2,934	2,626	5,527	59K	30 fps
NfS [244]	100	169	3,830	2,448	20,665	383K	240 fps
GOT-10k [245]	10,000	-	-	-	-	1.5M	10 fps
LaSOT [246]	1,400	1,000	2,506	2,053	11,397	3.52M	30 fps
ExtremeTrack	199	71	473	390	3,872	94K	30 fps

6.3.1 Extreme Weather Tracking Dataset

6.3.1.1 Comparison with existing dataset

In recent years, significant advancements in single object tracking have been achieved due to continuous innovation in deep learning methodologies and the availability of benchmark datasets, such as OTB, GOT-10k, LaSOT, and various versions of VOT. However, most existing datasets primarily focus on long-range tracking benchmarks or large-scale datasets for training and testing under controlled conditions.

The OTB-2013 dataset, one of the earliest benchmarks for visual object tracking, comprises 51 videos exclusively for testing. It gained popularity due to the dominance of image-processing-based algorithms during its inception. However, the lack of training data and its limited applicability to modern deep-learning techniques made it less relevant over time. An updated version, OTB-2015, includes 100 testing videos but still lacks sufficient diversity for training deep learning models.

The TC-128 dataset was specifically developed to evaluate color-based tracking algorithms, consisting of 128 testing videos spanning 11 challenging scenarios. Similarly, the VOT dataset series, including VOT-2014 and VOT-2017, contains 25 and 60 videos, respectively. These datasets are characterized by their short video sequences, making them suitable only for testing purposes. The NUS-PRO dataset targets human and rigid object tracking, with 365 annotated frames designed to evaluate performance under occlusion. It remains a valuable benchmark for assessing algorithms in scenarios involving occluded objects. UAV123 and UAV20L

datasets were introduced for unmanned aerial vehicle tracking, containing 123 and 20 videos, respectively. These datasets focus on aerial applications but lack generalizability to other adverse conditions.

The NfS dataset includes 100 high-frame-rate sequences (240 fps) to track fast-moving objects, providing insights into performance under appearance variations. GOT-10k and LaSOT are large-scale datasets with high frame rates (30 fps), containing extensive sequences designed for diverse object tracking scenarios. Despite their scale, these datasets primarily feature high-quality videos and do not account for adverse conditions such as low visibility or degraded environments.

The need for a dataset addressing real-world challenges, including adverse weather and low-quality videos, remains largely unfulfilled in the existing benchmarks. The *ExtremeTrack* dataset addresses this gap by providing a degraded video benchmark, enabling the evaluation of object tracking algorithms under challenging conditions such as haze, rain, and other environmental adversities. A detailed comparison of existing datasets and the *ExtremeTrack* dataset is presented in Table 6.1, highlighting its unique features and suitability for benchmarking object tracking in extreme conditions. The proposed *ExtremeTrack* dataset consists of 100 hazy videos and 99 rainy videos, out of that 159 are for training and 40 are for testing.

6.3.1.2 Data Collection and Annotation

As described earlier, the proposed dataset contains 100 hazy videos and 99 rainy videos. These videos are primarily sourced from YouTube under the Creative Commons license, with additional samples captured using mobile cameras and selected videos extracted from the OTB-2015 dataset. The selected videos feature challenging tracking scenarios, ensuring that the dataset poses significant difficulties for object-tracking algorithms.

The dataset is annotated manually using the LabelIMG[1] software. The annotation process follows the standard bounding box annotation format and employs a deterministic annotation strategy. For each video, the annotation process focuses on a specific tracking target. If the target object is visible in a frame, a labeler manually draws or edits a bounding box to tightly encapsulate any visible part of

[1]https://github.com/HumanSignal/labelImg

the target. In cases where the target object is absent due to being out-of-view or fully occluded, the frame is assigned an absent label.

This strategy does not guarantee minimal background inclusion within the bounding box but provides consistent annotations that are stable for learning object dynamics. The annotation process involves four annotators. To ensure high annotation quality, the annotations undergo a three-step verification process conducted by other annotators.

All source videos used in the dataset are of high quality and free of haze or rain. Synthetic hazy and rainy videos are generated using clear video footage through two approaches: a pix-to-pix GAN model and Kinemaster[2] software. Details of the synthetic video generation process are provided in the following subsections.

6.3.1.3 Design Principle

- **Hazy Data Generation:** The Pix2Pix GAN framework, widely recognized for its effectiveness in image-to-image translation tasks, is utilized for generating hazy videos. The model is trained on the Reside6K dataset, which provides a diverse range of paired haze-free and hazy images, enabling the robust learning of transformations between these two domains. The input to the network consists of haze-free images, while the output comprises corresponding hazy images generated by the trained GAN.

  The GAN framework comprises a generator and a discriminator. The generator is tasked with transforming haze-free images into their hazy counterparts, while the discriminator distinguishes real hazy images from generated ones, guiding the generator toward producing photorealistic haze effects. The training process minimizes a combined objective function:

$$L_{\text{total}} = L_{\text{GAN}}(G, D) + \lambda L_{L1}(G), \tag{6.1}$$

  where $L_{\text{GAN}}(G, D)$ is the adversarial loss, and $L_{L1}(G)$ represents the pixel-wise $L1$-loss. The adversarial loss is formulated as:

$$L_{\text{GAN}}(G, D) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))], \tag{6.2}$$

[2]https://kinemaster.com/en

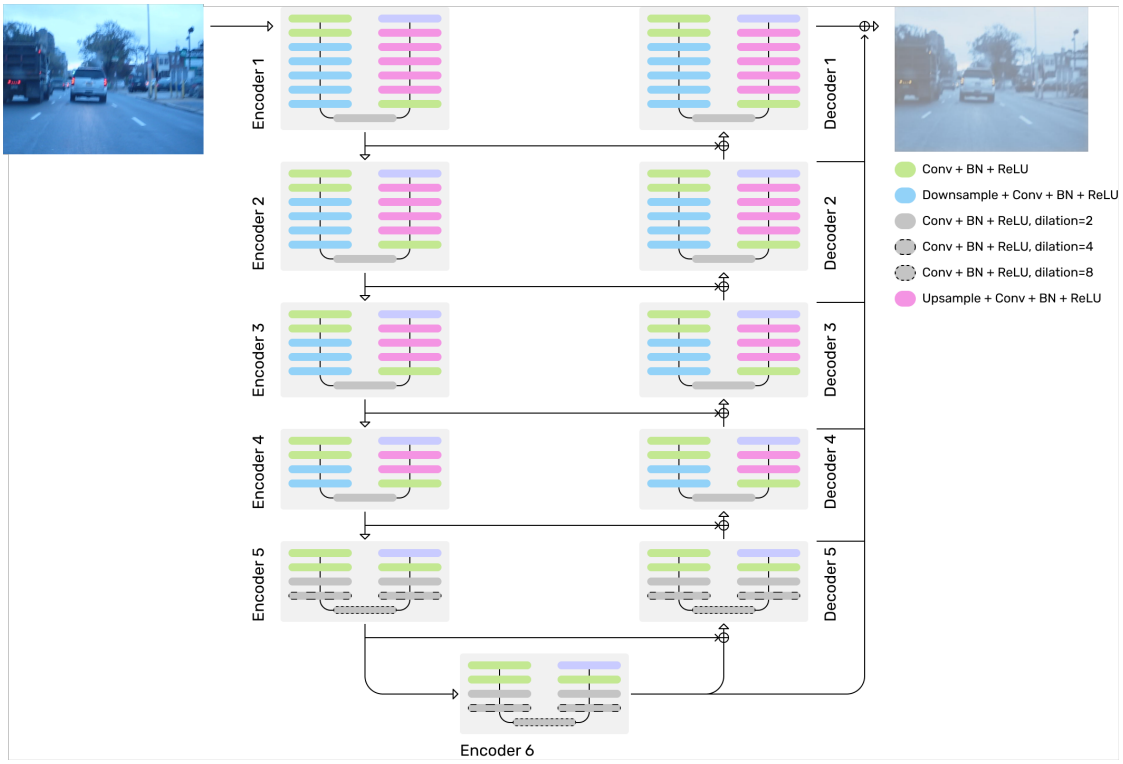

FIGURE 6.1: Architecture of the U^2 model used in the generator of the pix-to-pix GAN model for synthetic hazy frame generation.

where G and D represent the generator and discriminator, respectively, x is the input haze-free image, and z is a random noise vector. The $L1$-loss encourages similarity between the generated hazy image and the ground truth:

$$L_{L1}(G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|G(x) - y_{\text{hazy}}\|_1], \tag{6.3}$$

where y_{hazy} is the ground truth hazy image. The balance between these losses ensures that the generator produces perceptually realistic hazy images while maintaining fidelity to the ground truth.

For the generator, a $U^2 - Net$-based architecture is employed. This architecture consists of six encoder blocks and five decoder blocks arranged in a U-Net-like structure. Encoder blocks progressively downsample the input using convolutional layers with dilation factors of $2, 4,$ and 8, allowing the network to capture multi-scale contextual information. Each decoder block upscales the feature maps, restoring spatial resolution through convolutional layers with batch normalization and ReLU activation. Skip connections link corresponding encoder and decoder blocks, preserving critical features and improving output quality.

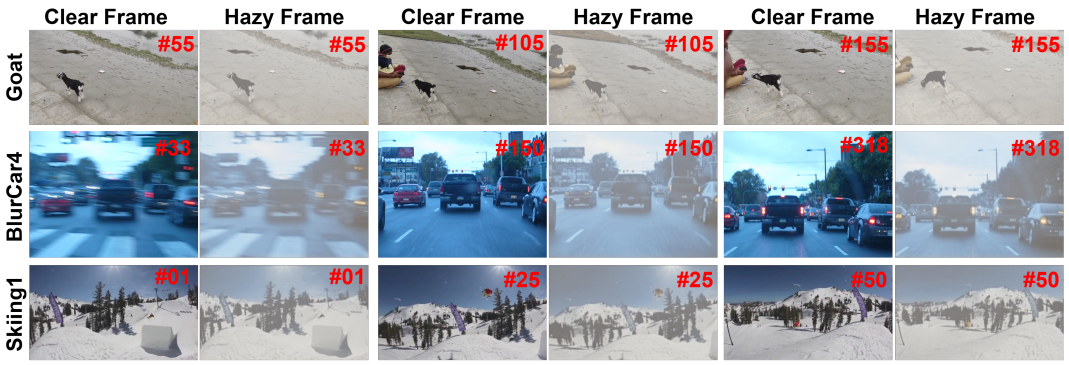

FIGURE 6.2: Example of the hazy video dataset.

The training process spans 100 epochs, enabling the generator to produce realistic hazy images while the discriminator sharpens its ability to differentiate real hazy images from generated ones. Following training, the generator processes individual frames of haze-free videos, transforming each into a corresponding hazy frame. These frames are subsequently reassembled to construct the final hazy video, ensuring a seamless and realistic output.

This approach effectively integrates multi-scale feature extraction, adversarial learning, and pixel-wise accuracy, demonstrating a robust capability for realistic haze generation in video sequences.

- **Rainy Data Generation :** For the generation of rainy videos, a Green Screen technique combined with KinMaster video editing software is used to overlay synthetic rainy effects onto natural, rain-free video sequences. The Green Screen video, which consists of a rain scene filmed in front of a green backdrop, serves as the source for creating the rainy texture. This video provides the visual elements required for superimposing the rainy effect onto the base video.

  The process involves two main steps. First, the green background in the rainy video is removed using chroma keying, which isolates the rain content, including streaks and other rainy effects. This step ensures that only the rainy elements remain. Next, the isolated rainy video is seamlessly overlaid onto the target rain-free video. This integration ensures that the rain effect blends naturally with the original background, creating the appearance of a realistic rainy environment.

  The editing capabilities of KinMaster allow precise control over the intensity, positioning, and blending of the rainy effect. This ensures that the composite rainy video retains the natural characteristics of rain while preserving the

original content of the base video. The final video is a composite where the rain-free video is enhanced with a visually convincing rainy overlay.

The mathematical formulation of this process is expressed as:

$$V_{\text{final}}(t) = V_{\text{base}}(t) + \alpha(t) \cdot V_{\text{rain}}(t), \tag{6.4}$$

where:

- $V_{\text{final}}(t)$ is the final rainy video at time step t,
- $V_{\text{base}}(t)$ is the base (rain-free) video at time step t,
- $V_{\text{rain}}(t)$ is the green-screen rainy video at time step t,
- $\alpha(t)$ is the blending factor at time step t, controlling the intensity of the rain effect.

In this formulation, the Green Screen rainy video $V_{\text{rain}}(t)$ is added to the rain-free video $V_{\text{base}}(t)$ at each time step t. The blending factor $\alpha(t)$ dynamically adjusts the rain effect's intensity, ensuring a natural-looking transition between the two video components. This approach achieves a realistic and visually consistent rainy video by effectively combining synthetic rain effects with natural video sequences.

6.3.2 Proposed Model

This section describes the proposed zero-shot model called *ArcTrack*, which comprises a rain and haze removal model which is described in the previous chapter, a zero-shot object segmentation model FastSAM, a new future similarity method, and *ArcTrack* model.

6.3.2.1 FastSAM

FastSAM introduces a real-time segmentation framework designed for efficiency and precision. It comprises two stages: All-Instance Segmentation and Prompt-Guided Selection. Unlike traditional transformer-based models, it incorporates convolutional designs and receptive-field strategies, enabling faster convergence with fewer parameters while maintaining high accuracy.

The All-Instance Segmentation stage leverages YOLOv8 as its foundation, incorporating advanced features like the Coarse-to-Fine module and anchor-free detection. The segmentation branch produces k prototypes and mask coefficients that, combined with high-resolution feature maps, yield instance masks for every object in the image. This architecture supports parallel segmentation and detection tasks, ensuring efficient processing across diverse scenarios.

In the second stage, prompts refine the segmentation to isolate specific objects of interest. Three types of prompts are employed:

- **Point Prompt:** Matches foreground/background points to masks, refining results with morphological operations.
- **Box Prompt:** Selects masks based on the highest IoU with a bounding box.
- **Text Prompt:** Uses CLIP embeddings to match text descriptions to masks.

By integrating these prompt-based techniques, FastSAM achieves precise object selection in real-time, enhancing YOLOv8's capabilities for complex segmentation tasks. The framework demonstrates significant potential for practical applications, where real-time and accurate segmentation are critical.

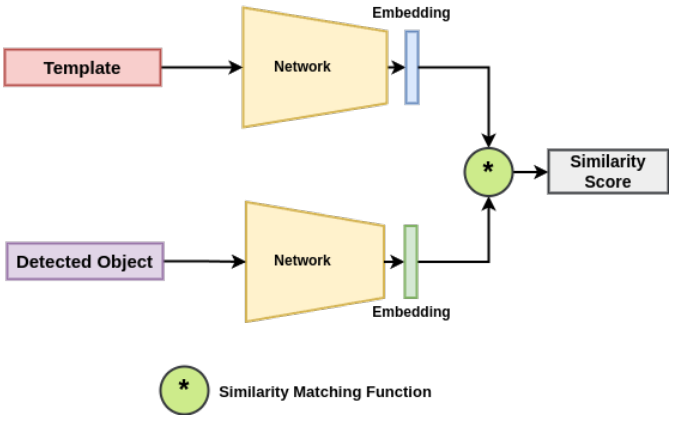

FIGURE 6.3: Model overview of the similarity matching of the template and detected object.

6.3.2.2 Feature Similarity Measures

Feature similarity measurement plays a critical role in various fields, including Computer Vision, Machine Learning, and Deep Learning. In this context of Arc-feature representation, the similarity between features is vital for tasks such as object identification, face recognition, and object tracking. This section outlines

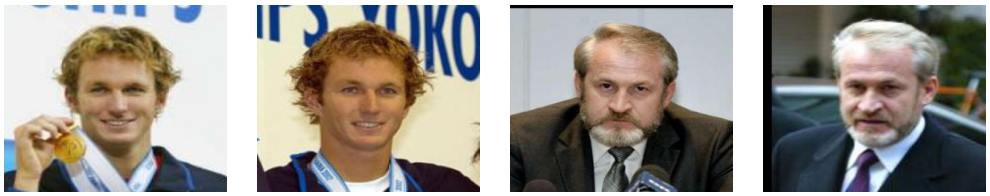

FIGURE 6.4: Example of LFW dataset (a) and (b) are two different images of Aaron Peirsol and (c) and (d) are images of Akhmed Zakayev

four similarity measures that can be applied to Arc features, including Difference Similarity, Cosine Similarity, Structural Similarity Index Measure (SSIM), and Cross-Correlation Similarity.

- **Difference Similarity :** The Difference Similarity measure is computed as the mathematical difference between two feature vectors. For two features f_1 and f_2, it is defined as:

$$D = 1 - \sum_{i=0}^{n-1} \left| f_1^i - f_2^i \right| \tag{6.5}$$

  where n is the number of feature points (e.g., $n = 512$ for ArcFace features). A higher value of D indicates greater similarity between the two features. The threshold Th is used to determine whether the features are similar:

$$D > Th \implies \text{Features are similar.} \tag{6.6}$$

- **Cosine Similarity:** Cosine Similarity calculates the angle between two feature vectors in vector space, providing a measure of their alignment. It is computed as:

$$\cos(f_1, f_2) = \frac{\sum_{i=0}^{n-1} f_1^i f_2^i}{\sqrt{\sum_{i=0}^{n-1} (f_1^i)^2} \sqrt{\sum_{i=0}^{n-1} (f_2^i)^2}} \tag{6.7}$$

  The similarity is determined based on a threshold:

$$\cos(f_1, f_2) > Th \implies \text{Features are similar.} \tag{6.8}$$

  For identical features, the cosine similarity is ideally 1.

- **Structural Similarity Index Measure (SSIM):** SSIM is traditionally used in image comparison but can also be adapted for feature similarity. For

two features f_1 and f_2, SSIM is computed using Equation 3.23. A higher SSIM value, ideally 1, indicates greater similarity between the features.

- **Cross-Correlation Similarity:** Cross-Correlation measures the sliding inner product of two feature vectors, providing insight into their relative displacement. It is defined as:

$$(f_1 \star f_2) = \max_i \left(\sum_{i=0}^{n-1} f_1^i f_2^i \right) \tag{6.9}$$

where the maximum value is taken over all possible alignments. If the maximum value of cross-correlation exceeds a threshold, the features are considered similar:

$$(f_1 \star f_2) > Th \implies \text{Features are similar.} \tag{6.10}$$

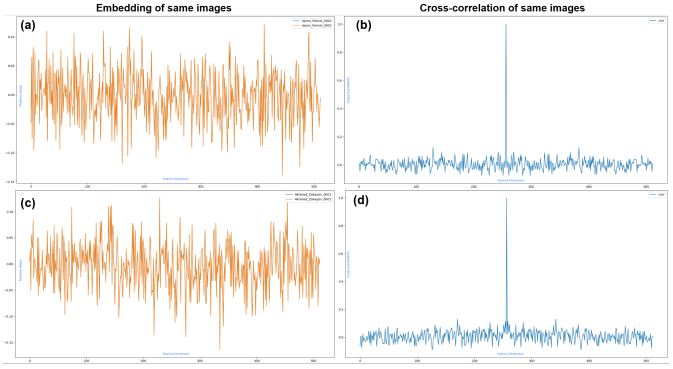

FIGURE 6.5: Embedding and Cross-correlation of the same image of the same person (a) Embedding of Aaron Peirsol's image 2, (b) Cross-correlation of Aaron Peirsol's image 2, (c) Embedding of Akhmed Zakayev's image 1, (d) Cross-correlation of Akhmed Zakayev's image 1

For example, the first case in Figure 6.5(b) and Figure 6.5 (d) as shows the cross-correlation between the two features of the same photo of the same person, Aaron Peirsol's image 2 and Akhmed Zakayev's image 1 respectively. For each of the cases, we got a peak at a lag of zero and $\max_i(f_1 \star f_2) = 1$ because the images are the same. Now for the second case, Figure 6.6(b) and Figure 6.6 (d) are shown the cross-correlation between the two features of the different images of the same person, Aaron Peirsol, and Akhmed Zakayev respectively. $\max_i(f_1 \star f_2) = 0.78$ for Aaron Peirsol and 0.82 for Akhmed Zakayev because though the image is different the person is the same for each of the cases we got a peak at a lag of zero. Lastly, Figure 6.7(b) and

Figure 6.7 (d) show the cross-correlation between the two features of the same person, Aaron Peirsol's image 2 and Akhmed Zakayev's image 2 and Aaron Peirsol's image 1 and Akhmed Zakayev's image 1 respectively. We obtain $\max_i(f_1 \star f_2) = 0.12$ for Aaron Peirsol and 0.11 for Akhmed Zakayev. There is no peak at a lag of zero because the person is different.

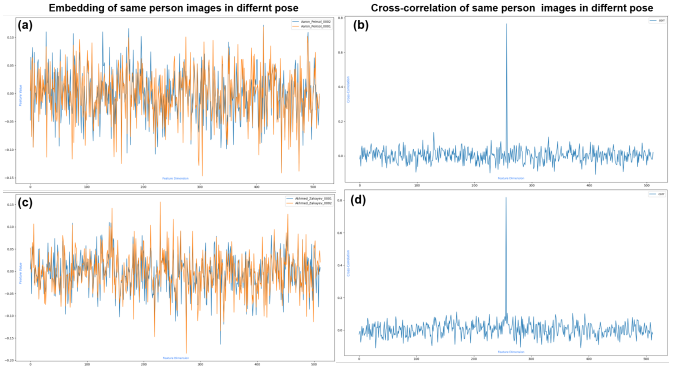

FIGURE 6.6: Embedding and Cross-correlation of different images of the same person, (a) Embedding of Aaron Peirsol's image 1 and Aaron Peirsol's image 2, (b) Cross-correlation of Aaron Peirsol's image 2 and Aaron Peirsol's image 2, (c) Embedding of Akhmed Zakayev's image 1 and (d) Akhmed Zakayev's image 2, Cross-correlation of Aaron Peirsol's image 1 and Aaron Peirsol's image 2

To evaluate the effectiveness of different similarity measures, we tested them using two benchmark datasets: LWF[247], AgeDB30[248], and CFP-FP[249], which are commonly used for face verification tasks. From the data presented in Table 6.2, it is evident that the SSIM method outperforms other similarity functions on the LWF and AgeDB30 datasets when using the ResNet50 backbone, particularly when the faces are not cropped. ResNet50 also shows the best performance on the CFP-FP dataset with both Difference Similarity and Cosine Similarity. For cropped faces, SSIM again provides the best results on LWF, but for AgeDB30 and CFP-FP, the performance of Difference Similarity, Cosine Similarity, and Cross-Correlation Similarity is quite similar on the ResNet50 model. On the MobileNetv2 backbone, Cross-Correlation Similarity and Cosine Similarity tend to provide the best accuracy across LWF, AgeDB30, and CFP-FP datasets.

Regarding the true positive rate, SSIM consistently achieves the highest performance on the LWF, AgeDB30, and CFP-FP datasets for both ResNet50 and MobileNetv2 models. However, on the LWF dataset, Cross-Correlation Similarity yields the best results, as indicated in Table 6.3.

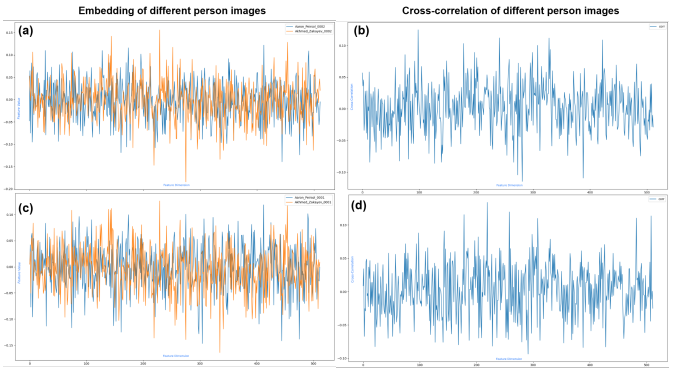

FIGURE 6.7: Embedding and Cross-correlation of different person, (a) Embedding of Aaron Peirsol's image 2 and Akhmed Zakayev's image 2, (b) Cross-correlation of Aaron Peirsol's image 2 and Akhmed Zakayev's image 2, (c) Embedding of Aaron Peirsol's image 1 and Akhmed Zakayev's image 1, and (d) Cross-correlation of Aaron Peirsol's image 1 and Akhmed Zakayev's image 1

TABLE 6.2: Comparison of Accuracy of different Similarities on ResNet50 and MobileNetv2 backbones. The best result is shown in red color.

Model	→	**Arc+ResNet50**		**Arc+MobileNetv2**	
Dataset	**Crop**	False	True	False	True
	Similarity function				
LWF[247]	Difference Similarity	99.42	99.30	99.10	98.88
	Cross-Correlation Similarity	99.42	99.23	**99.20**	**98.98**
	Cosine Similarity	99.42	99.23	**99.20**	**98.98**
	SSIM	**99.48**	**99.32**	99.12	98.83
AgeDB30[248]	Difference Similarity	95.32	**94.92**	91.62	**90.83**
	Cross-Correlation Similarity	95.32	**94.92**	**91.63**	90.72
	Cosine Similarity	95.32	**94.92**	**91.63**	90.72
	SSIM	**95.48**	94.75	91.55	90.28
CFP-FP[249]	Difference Similarity	**92.56**	**91.23**	91.50	**90.83**
	Cross-Correlation Similarity	92.37	91.11	**91.66**	**90.83**
	Cosine Similarity	**92.56**	**91.23**	**91.66**	**90.83**
	SSIM	92.23	90.97	91.13	90.09

6.3.2.3 ArcTrack

Object tracking in adverse weather conditions such as haze and rain presents significant challenges due to reduced visibility and distortion of visual features. The proposed model addresses these challenges by integrating haze and rain removal, robust object detection, similarity-based matching, and Kalman filtering. This section details the pipeline, highlighting key methodologies and mathematical formulations.

TABLE 6.3: Comparison of True Positive Rate of different Similarities on ResNet50 and MobileNetv2 backbones. The best result is shown in red color.

Model	→	**Arc+ResNet50**		**Arc+MobileNetv2**	
Dataset	**Crop**	False	True	False	True
	Similarity function				
LWF[247]	Difference Similarity	86.55	85.24	86.58	85.41
	Cross-Correlation Similarity	**93.22**	**92.56**	**93.23**	**92.64**
	Cosine Similarity	93.21	**92.56**	**93.23**	**92.64**
	SSIM	93.17	92.52	**93.23**	**92.64**
AgeDB30[248]	Difference Similarity	75.11	74.06	75.13	73.84
	Cross-Correlation Similarity	87.53	87.00	87.53	86.90
	Cosine Similarity	87.49	86.97	87.50	86.86
	SSIM	**87.63**	**87.11**	**87.71**	**87.10**
CFP-FP[249]	Difference Similarity	85.24	69.85	72.50	71.54
	Cross-Correlation Similarity	85.69	84.98	86.26	85.78
	Cosine Similarity	85.61	84.86	86.19	85.71
	SSIM	**85.79**	**85.08**	**86.41**	**85.94**

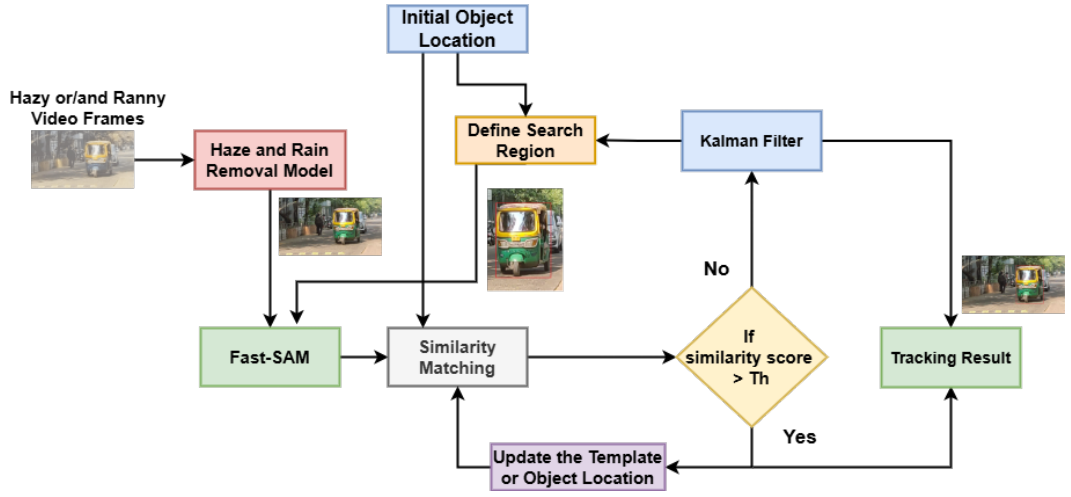

FIGURE 6.8: Flowchart of the proposed object tracking algorithm.

a) **Input and Initialization:** The tracking process begins with two primary inputs:

- A video frame F_t at time t, captured under adverse weather conditions, which contains the object of interest.
- An initial bounding box B_0, representing the object's location and dimensions:

$$B_0 = (x_0, y_0, w_0, h_0),$$

where (x_0, y_0) denote the top-left corner of the bounding box, and w_0 and h_0 are its width and height, respectively.

This initialization anchors the model by providing a reference point for subsequent object detection and tracking.

b) **Haze and Rain Removal:** To enhance object visibility, the input frame F_t undergoes preprocessing via a joint haze and rain removal model R. This neural network leverages domain-specific adaptations to restore image clarity, transforming F_t into F_t^{clear}:

$$F_t^{\text{clear}} = R(F_t).$$

The model R combines techniques such as multi-scale feature extraction and attention mechanisms to address the spatial and temporal inconsistencies caused by varying weather conditions. By improving the quality of F_t^{clear}, subsequent operations can focus on accurate object detection and tracking.

c) **Search Region Extraction:** To locate the object in frame F_t^{clear}, a search region R_t is extracted around the object's last known position. The search region is defined to be larger than the previous bounding box to accommodate potential object displacement:

$$R_t = \text{Crop}(F_t^{\text{clear}}, (x_{t-1}, y_{t-1}, k \cdot w_{t-1}, k \cdot h_{t-1})),$$

where (x_{t-1}, y_{t-1}) are the coordinates of the bounding box center in frame $t-1$, and k is a scaling factor (e.g., $k = 2$) to ensure sufficient coverage. This approach minimizes the risk of the object falling outside the search region while maintaining computational efficiency.

d) **Object Detection Using Fast-SAM:** To detect objects within the extracted search region R_t, the Fast Segment Anything Model (Fast-SAM) [250] is employed. Fast-SAM is a lightweight, high-speed object detection and segmentation framework capable of processing regions with minimal latency. The model outputs a set of bounding boxes D_t for all detected objects in R_t:

$$D_t = \{D_1, D_2, \ldots, D_n\},$$

where each bounding box $D_i = (x_i, y_i, w_i, h_i)$ represents the top-left corner coordinates (x_i, y_i), width w_i, and height h_i of a detected object. The set D_t

typically includes the target object as well as other objects present within the search region.

e) **Similarity Matching for Object Identification:** To isolate the target object from D_t, a similarity-based matching process is employed. Each detected object D_i is compared with the object template T_t, which encapsulates the target object's appearance. The similarity score $S(T_t, D_i)$ is computed using the Structural Similarity Index Measure (SSIM):

$$S(T_t, D_i) = \text{SSIM}(T_t, D_i),$$

where SSIM evaluates the perceptual similarity in structural details between T_t and D_i. To ensure robust feature extraction, the MobileNetv2 backbone is utilized, leveraging its efficient architecture to extract deep feature representations from both T_t and D_i.

If the highest similarity score $S(T_t, D_i^*)$ exceeds a predefined threshold τ, the corresponding bounding box D_i^* is identified as the target object:

$$T_{t+1} = D_i^* \quad \text{if } S(T_t, D_i^*) \geq \tau.$$

The template T_{t+1} is then updated to reflect the newly identified bounding box, enabling the model to adapt dynamically to variations in the object's appearance over time.

f) **Occlusion Handling with Kalman Filtering:** In cases where no detected object satisfies the similarity threshold:

$$S(T_t, D_i^*) < \tau \quad \forall\, D_i \in D_t,$$

the model assumes the object is either occluded or not detected. Under such circumstances, a Kalman filter is employed to predict the object's location based on its motion dynamics. The predicted bounding box P_t is derived as:

$$P_t = K(P_{t-1}),$$

where K denotes the Kalman filter operation, and P_{t-1} represents the bounding box in the previous frame.

The predicted bounding box P_t is used to redefine the search region R_t, ensuring tracking continuity:

$$R_t = \text{Crop}(F_t^{\text{clear}}, (x_t, y_t, 2w_t, 2h_t)),$$

where (x_t, y_t, w_t, h_t) correspond to the coordinates and dimensions predicted by the Kalman filter.

g) **Template Update:** Upon successful detection or prediction, the object template T_t is updated to incorporate the latest bounding box. This update ensures that the template evolves to reflect changes in appearance, illumination, and orientation, thereby enhancing the reliability of similarity matching in subsequent frames.

The tracking process is formulated as a recursive function:

$$T_{t+1} = \begin{cases} D_i^* & \text{if } S(T_t, D_i^*) \geq \tau, \\ P_t & \text{otherwise,} \end{cases}$$

where D_i^* represents the best-matched object, and P_t is the predicted bounding box.

The proposed tracking model integrates Fast-SAM for efficient object detection, similarity-based matching using SSIM for robust object identification, and Kalman filtering for occlusion handling. This hybrid approach ensures reliable object tracking across frames, even in adverse weather conditions, occlusion scenarios, and dynamic environments. By continuously updating the object template, the model achieves high accuracy and adaptability in challenging real-world settings.

6.4 Experiment Results

The proposed model is evaluated against various existing models, including KCF[74], BOOSTING[66], MEDIAN-FLOW[84], MIL[85], MOOSE[71], TLD[75], Kalman Filter[19], Particle Filter[54], MS[18], and CSRT[251]. The evaluation utilizes a proposed dataset consisting of 40 videos, with 20 videos representing hazy conditions and 20 videos representing rainy conditions. Both qualitative and quantitative comparisons are employed to assess the model's performance. Quantitative

evaluation incorporates metrics such as success rate, precision, and object tracking error. Qualitative analysis involves the use of precision plots, success plots, and object tracking error plots, which are discussed in detail in the following subsections.

Table 6.4: Tracking Performance in Haze Condition using Success Rate, Precision, and Object tracking error.

Tracker	Success Rate (%)	Precision (%)	Object racking error (px)
MOOSE	11.14	25.07	214.65
KCF	21.38	28.59	140.86
CSRT	45.99	43.39	166.52
MIL	22.20	38.17	149.90
MeanShift	12.22	13.01	240.07
KalmanFilter	0.00	0.13	539.22
Boosting	25.09	33.11	160.42
MedianFlow	15.30	21.78	203.26
TLD	24.04	32.67	191.05
ParticleFilter	0.02	3.05	248.84
Proposed	**51.20**	**54.71**	**108.50**

Table 6.5: Tracking Performance in Rain Condition using Success Rate, Precision, and Object tracking error.

Tracker	Success Rate (%)	Precision (%)	Object tracking error (px)
MOOSE	13.82	26.95	180.64
KCF	25.90	36.28	119.44
CSRT	46.47	48.79	147.44
MIL	23.32	41.52	138.71
MeanShift	8.68	8.32	216.65
KalmanFilter	0.00	0.13	540.30
Boosting	28.94	32.71	154.37
MedianFlow	17.86	22.83	190.81
TLD	23.67	30.51	176.82
ParticleFilter	1.56	2.88	207.60
Proposed	**57.31**	**49.02**	**89.37**

6.4.1 Quantitative comparison

For quantitative comparison, success rate, precision, and object tracking error are utilized. In the hazy video dataset, the results are presented in Table 6.4, while the results for the rainy dataset are shown in Table 6.5. On the hazy dataset, the proposed model achieves a success rate of 51.20%, a precision of 54.71%, and an object tracking error of 108.50 pixels. In comparison, the CSRT model secures the second position with a success rate of 46.47% and a precision of 48.79%. Regarding object tracking error, the KCF model achieves the second-best result with an error of 119.44 pixels. Conversely, the Kalman filter demonstrates significantly higher object tracking error and lower precision, as illustrated in Table 6.4.

For the rainy video dataset, the proposed model attains a success rate of 57.31%, a precision of 49.02%, and an object tracking error of 89.37 pixels. In contrast, the Kalman filter exhibits the highest error and the lowest precision, consistent with its performance on the hazy dataset. The CSRT model achieves the second-best results with a success rate of 45.99%, a precision of 43.39%, and an object tracking error of 166.52 pixels, as detailed in Table 6.5.

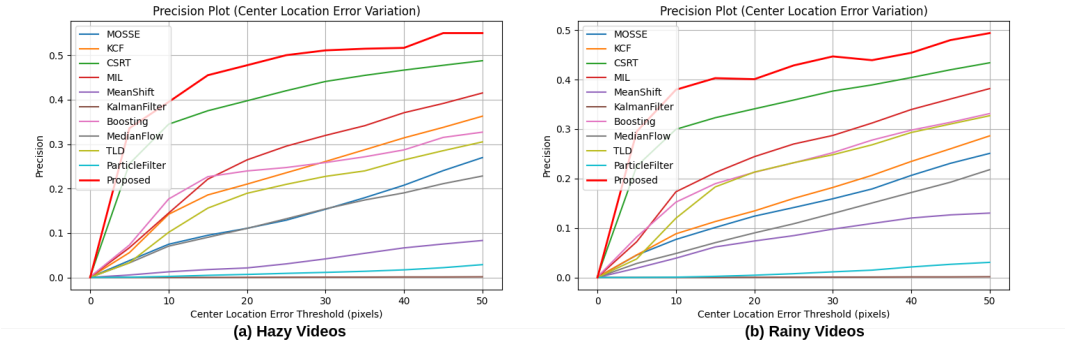

FIGURE 6.9: Precision comparison of the proposed model with existing models on (a) hazy video and (b) rainy video.

6.4.2 Qualitative comparison

Success plots, precision plots, and object tracking error plots are employed for qualitative comparison. Figure 6.9 (a) and Figure 6.9 (b) illustrate the precision plots of the proposed model and existing models on hazy and rainy videos, respectively. Similarly, Figure 6.10 (a) and Figure 6.10 (b) depict the success plots of the proposed model and existing models on hazy and rainy videos, respectively. In

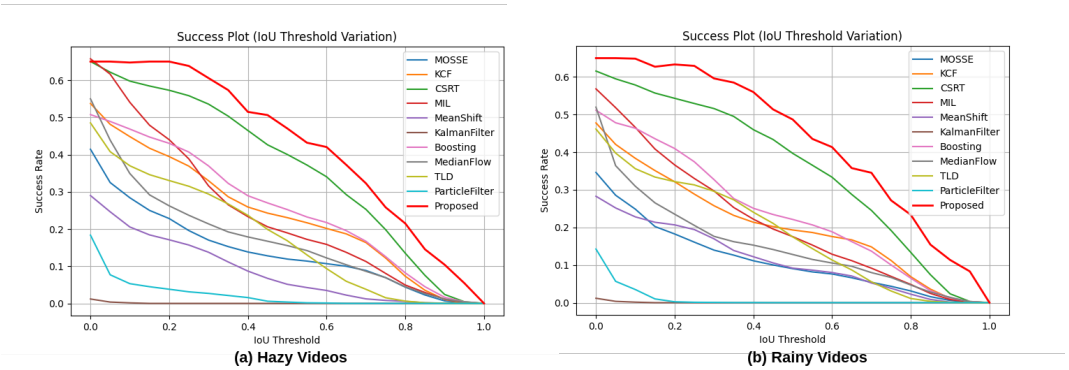

FIGURE 6.10: Success comparison of the proposed model with existing models on (a) hazy video and (b) rainy video.

all cases, the proposed model outperforms the existing algorithms, with the CSRT model achieving the second-best results. Conversely, the Kalman filter and the particle filter yield the lowest performance across both datasets.

Figure 6.11 and Figure 6.12 present the object tracking error plots for 20 hazy videos. The figures demonstrate that, although certain videos such as *bunjee_jumping, dat_2, dat_12, dat_32, Paralympic,* and *walking* exhibit high object tracking errors, the proposed model consistently outperforms the existing models. Similarly, Figure 6.13 and Figure 6.14 display the object tracking error plots for rainy videos. It is evident that videos like *bunjee_jumping, Car24, dat_12, dat_32, Human6, Human8, Nature_Wildlife, NOOSA_SURF, Paralympic, Penguin2,* and *walking* show higher object tracking errors. However, the proposed model still achieves lower object tracking errors compared to most of the existing object tracking algorithms.

6.5 Discussion

This chapter introduces a new synthetic dataset comprising a total of 199 videos, designed for training and testing object tracking models in hazy and rainy conditions. A WAE is used for haze and rain removal, serving as a preprocessing step. Additionally, a novel object tracking algorithm is introduced, specifically tailored for effective tracking in adverse weather conditions. The model leverages FastSAM, an unsupervised object detection and segmentation framework, for object localization. For similarity matching, a Siamese network pre-trained on the ImageNet dataset is utilized. The template and target images are processed through this network, and vector embeddings are matched using SSIM similarity. It is

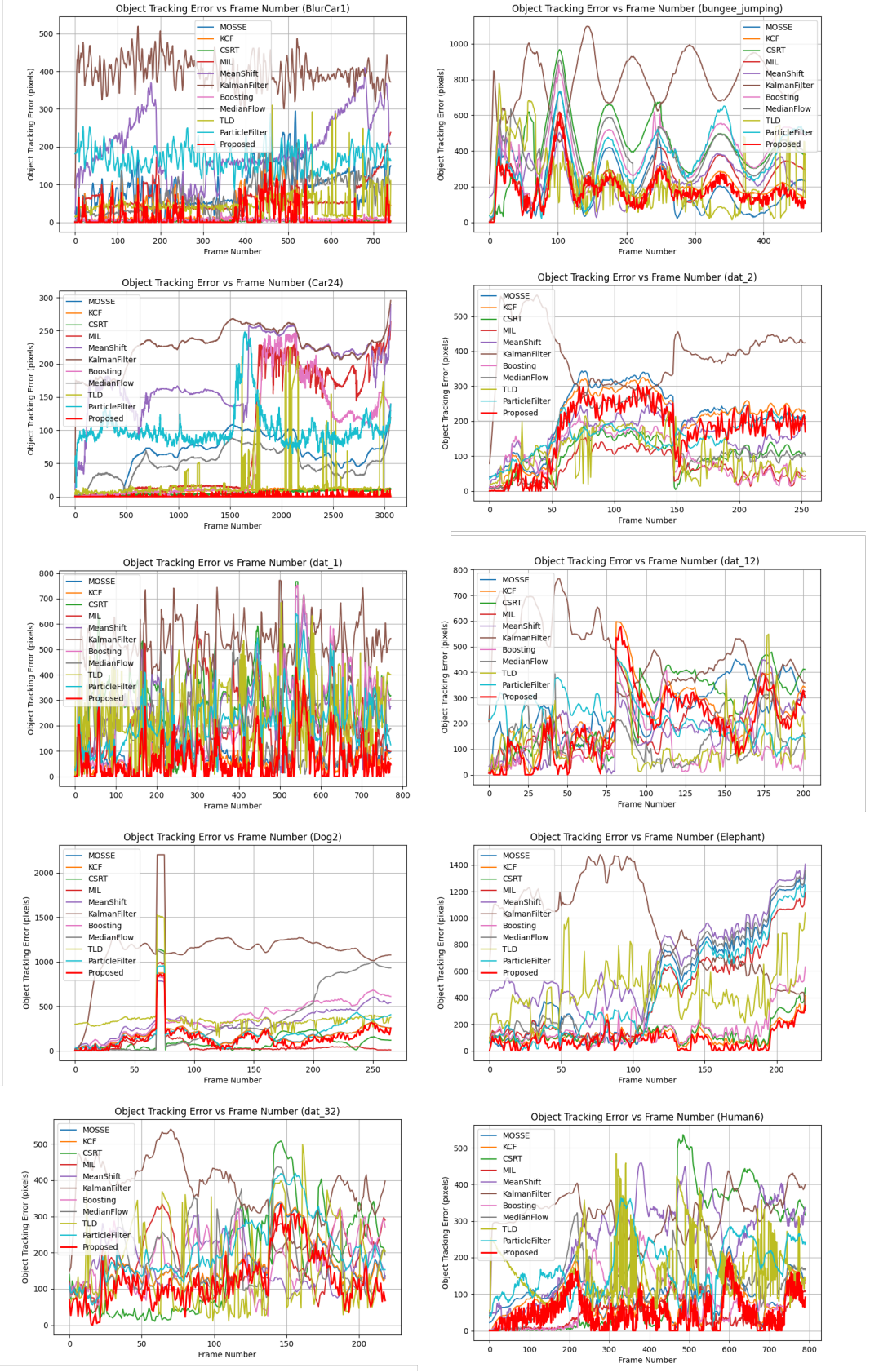

FIGURE 6.11: Object tracking error comparison of the proposed model with existing models on hazy videos.

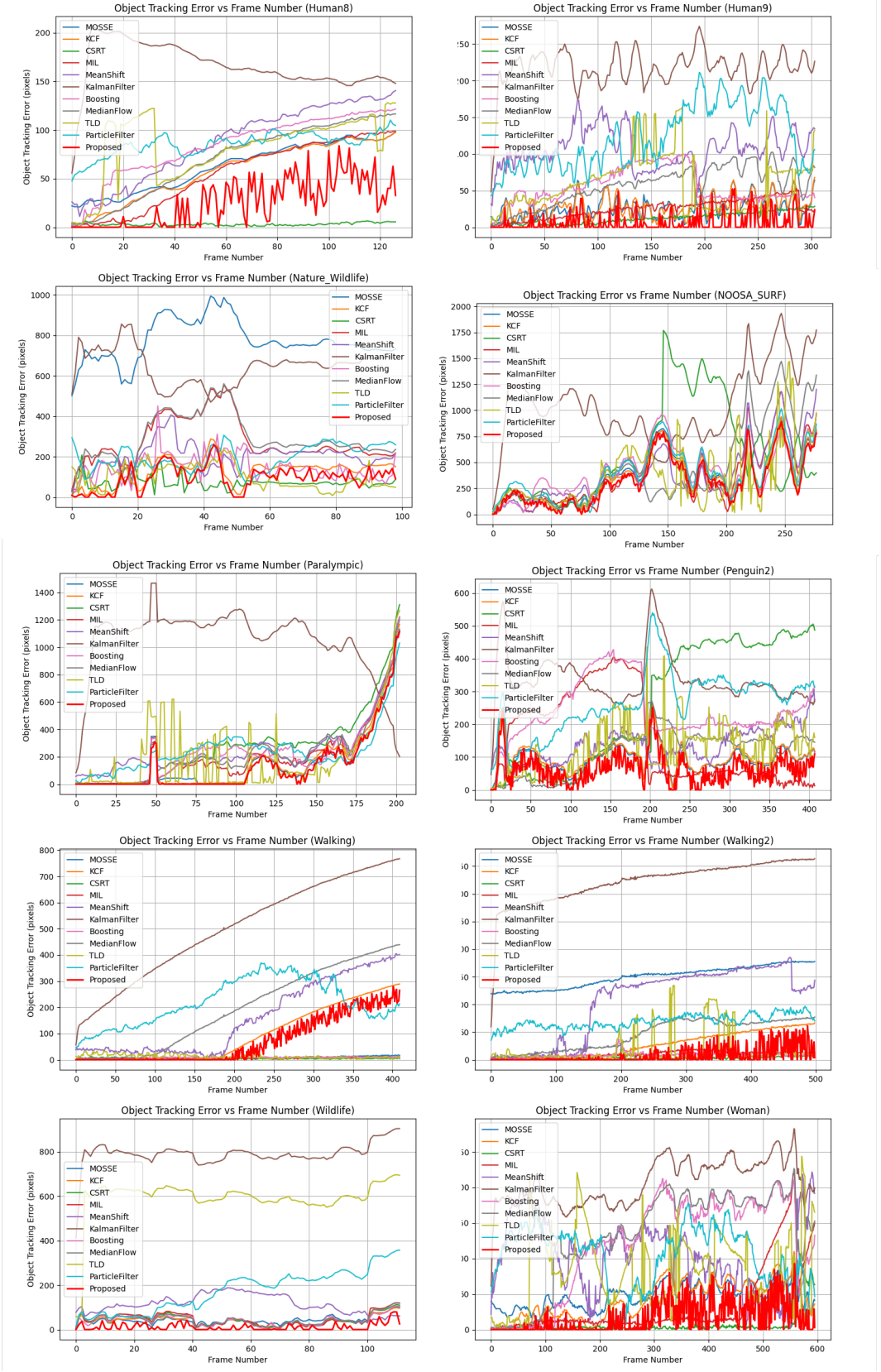

FIGURE 6.12: Object tracking error comparison of the proposed model with existing models on hazy videos.

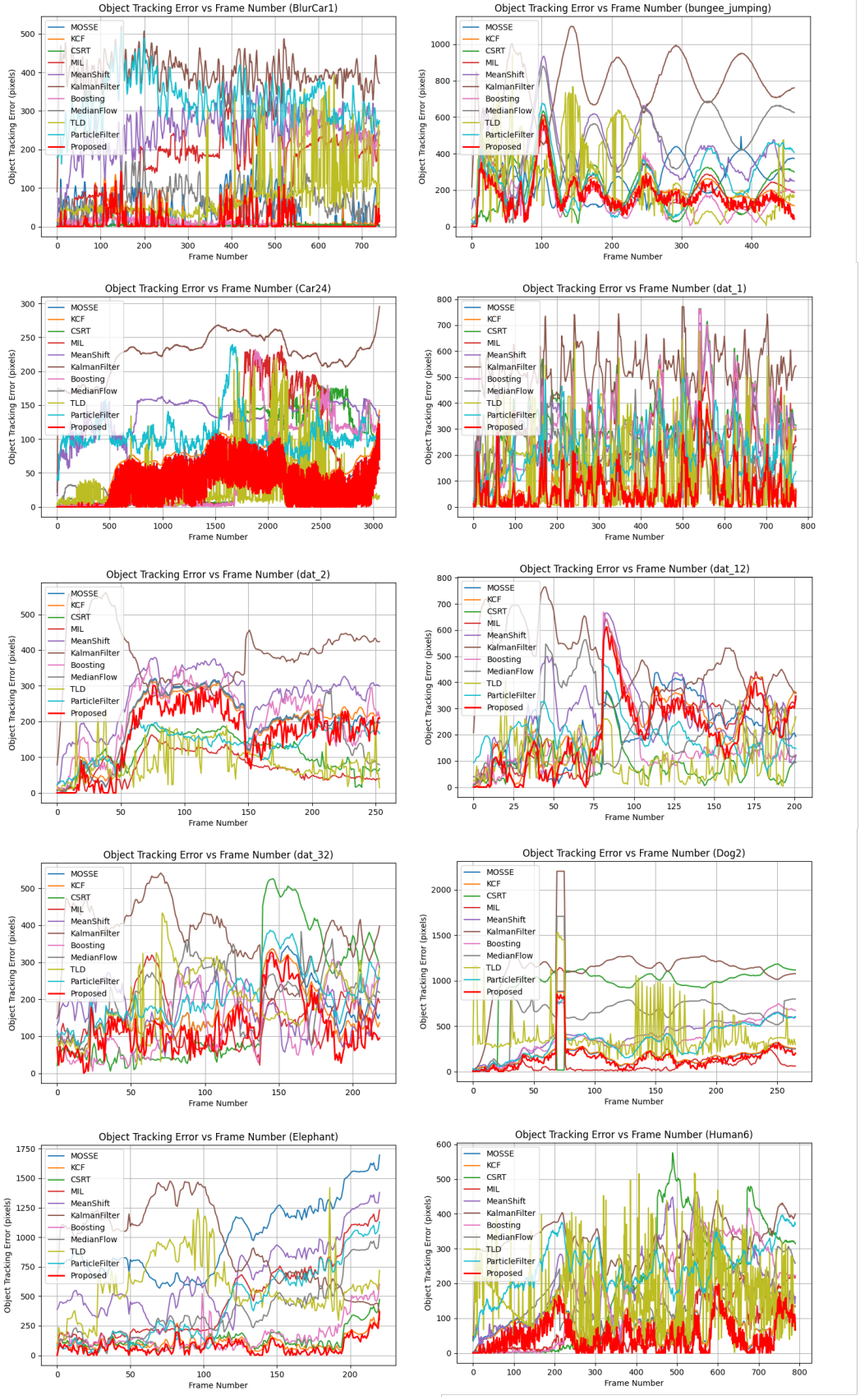

FIGURE 6.13: Object tracking error comparison of the proposed model with existing models on rain videos.

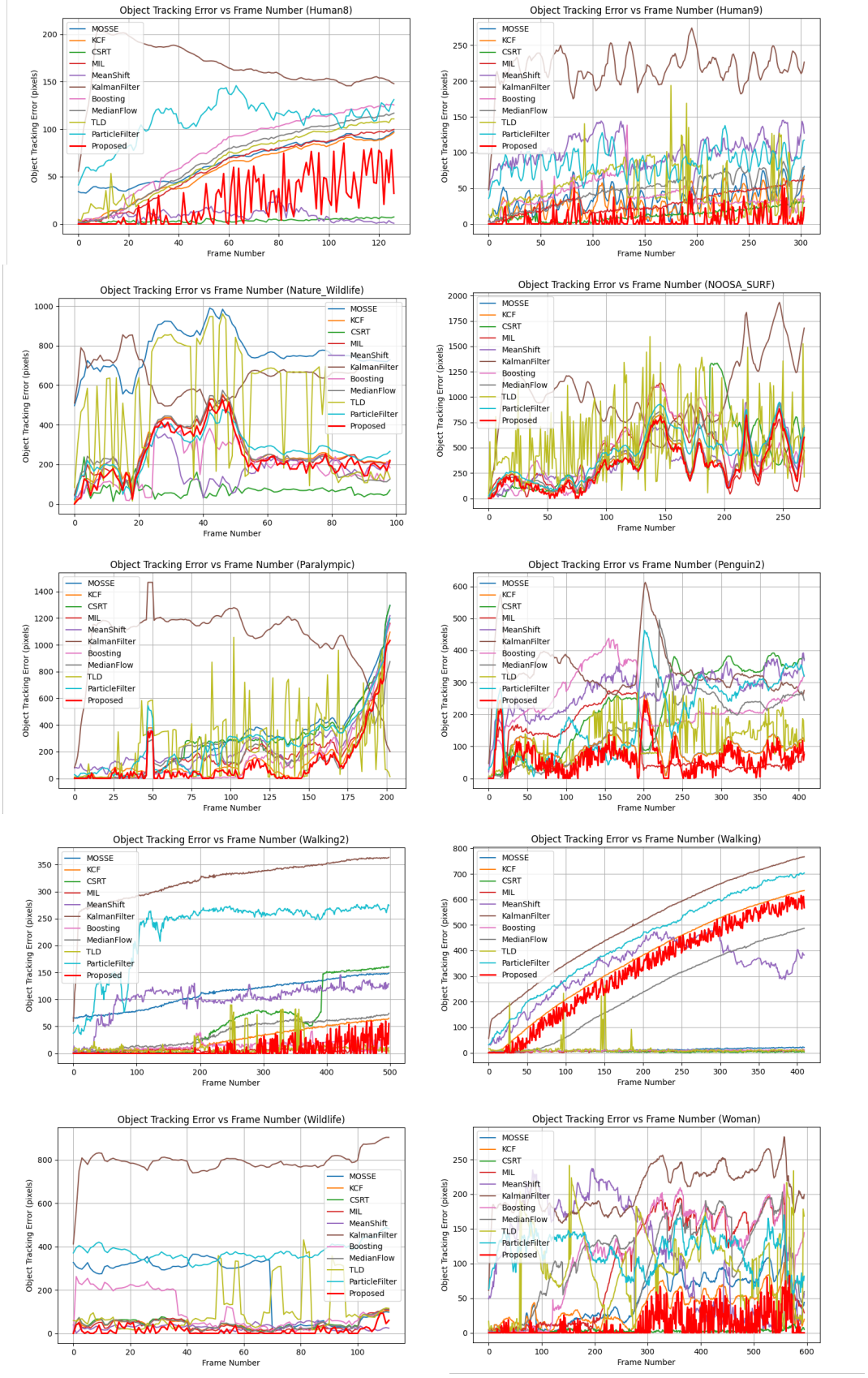

FIGURE 6.14: Object tracking error comparison of the proposed model with existing models on rain videos.

demonstrated that SSIM-based similarity matching outperforms the traditional cosine similarity approach. The proposed model also surpasses many existing object tracking algorithms in performance.

The model is entirely unsupervised and demonstrates robust performance on the synthetic dataset, achieving a success rate of 51.20% and a precision of 54.71% on hazy videos, as well as a success rate of 57.31% and a precision of 49.02% on rainy videos. There remains significant room for improvement in developing object-tracking models that can achieve higher accuracy in hazy and rainy weather conditions.

Chapter 7

Conclusion & Future Directions

With several real-time applications, object tracking is one of the difficult topics in computer vision. A tracker needs to be resistant to weather deterioration because such a system can be used for surveillance purposes. The primary emphasis of this thesis is object tracking under a variety of weather situations, such as hazy and rainy conditions. The main contributions, limitations, and potential future directions of this research are covered in this chapter, which also summarizes the thesis.

7.1 Summary

In the first part of the thesis, an object-tracking algorithm based on conventional computer vision is developed. While there are several object-tracking systems, they are all predicated on deep learning methodologies. Real-time performance is lacking in the existing standard computer vision-based and machine learning-based object-tracking systems. Those methods' inefficiency in tracking in scale-space is another problem. A multi-scale template matching tracker is created to address those issues. It primarily consists of two blocks: the template matching block and the region proposal block. The Mean Shift method (MS) with an Unscented Kalman Filter (UKF) makes up the region proposal block. Using a dynamic threshold to choose between UKF and MS under various circumstances, such as occlusion and rapid motion, is suggested. By determining the object's likely location, the region proposal block minimizes both the object's search area and the tracker's processing time. The precise location and dimensions of the target item are predicted using the template matching block. Evaluated on the OTB50 dataset, the proposed method demonstrates superior performance with a success rate of 68%, a precision of 73%, and an object tracking error of 24.30, outperforming state-of-the-art machine learning and deep learning-based trackers. Additionally, the proposed tracker achieves a tracking speed of 53 FPS on an Intel Core i5 2.3 GHz CPU with 8GB of RAM.

Although the developed tracker can manage quick motion, occlusion, and scale fluctuation, it cannot operate effectively under poor weather conditions. To resolve this problem, a new pre-processing block is required for the weather-damaged image repair. This thesis focuses on the restoration of deteriorated images caused by hazy weather. Current dehazing techniques are mostly based on either day-time or night-time haze models, which restrict their ability to handle haziness in

different lighting scenarios. The Light Invariant Dehazing Network (LIDN), an end-to-end picture dehazing network made up of four sub-modules—feature extractor, deep global atmospheric light estimator, medium transmission extractor, and encoder-decoder—is presented in this thesis as a solution to this issue. Sharper dehazed images are produced by the suggested model, which has been trained using quadruplet loss to efficiently eliminate artifacts. Comprehensive tests carried out in various lighting scenarios show that the suggested LIDN model outperforms the most advanced daytime and night-time dehazing techniques, with a PSNR value of 68.32 and SSIM value of 0.997 on the Reside6K dataset. With a runtime of just 0.24 seconds per image, the suggested model outperforms current dehazing techniques in terms of efficiency.

The developed haze removal method demonstrates a good performance on the Reside6K dataset, however, it is computationally demanding, which is undesirable. Thus, haze reduction with minimum processing overhead is required. Using a few video frames analyzed in a second, a unique haze metric called SATVAL—the ratio of an RGB image's highest saturation to maximum value—has been presented as a real-time approach for video dehazing. If a frame's SATVAL ratio is less than the set threshold value, it is either considered as haze-free or it passes without dehazing. This method enables the Raspberry Pi to run a 10 frames per second dehaze video sequence. However, this technique has limited effectiveness in eliminating haze. So a more generalized haze removal model is developed using a GAN-based model. The proposed model for dehazing process employs both generator and discriminator components and has been trained adversarially. To ensure better spatial and contextual information, various architectures like UNet, MANet, PSPNet, and FPN have been explored. Besides, to enhance feature extraction, the model includes a Vision Transformer, modified-MobileNet, EfficientNet, ResNet, and VGG as an encoder block. The model is trained and evaluated on the Reside6K dataset dataset, utilizing objective metrics such as PSNR and SSIM. It achieves impressive results with an SSIM of 0.893 and a PSNR of 24.89. Furthermore, the proposed approach is implemented on a Raspberry Pi device for real-time dehazing, demonstrating superior efficiency with a runtime lower than existing models and achieving a processing speed of 11 FPS.

The previous algorithms are designed only for removing haze from images. However, rain is also a weather phenomenon that can cause image degradation. Therefore, a novel Wavelet-based deep Auto-encoder, called WAE, has been proposed

in this thesis. This network is capable of simultaneously removing both haze and rain effects from images. The proposed model uses wavelet transformation and inverse wavelet transformation instead of down-sampling and up-sampling operations to add sparsity to the network. By training the model on both spatial and frequency domains, it can learn non-stationary features that are useful for removing both haze and rain effects from images. The proposed model has been extensively evaluated on multiple rain- and haze-affected image datasets, demonstrating strong performance across various benchmarks. On the Resid6K dataset, it achieves a PSNR of 24.37 and an SSIM of 0.906. For the Rain100L dataset, the model attains a PSNR of 22.47 and an SSIM of 0.7164, while on the Rain1200 dataset, it achieves a PSNR of 24.32 and an SSIM of 0.8622. On the ITS and OTS datasets, the model records PSNR values of 20.587 and 26.69, respectively, with corresponding SSIM scores of 0.878 and 0.955. For the Rain-KITTI2012 and Rain-KITTI2015 datasets, the model achieves PSNR values of 36.84 and 36.88, along with SSIM scores of 0.972 and 0.956, respectively. On the JRSRD dataset, it achieves a PSNR of 28.08 and an SSIM of 0.899. Additionally, the model was tested on the RID and RIS datasets for real rain scenarios, achieving NIQE scores of 4.255 and 4.301, respectively. The model was evaluated on Fattal's real-haze dataset for real-haze conditions, achieving a NIQE score of 3.7853. These results highlight the model's effectiveness in restoring rainy and hazy images, making it a valuable preprocessing tool for object tracking in degraded weather conditions.

Finally, an object tracking algorithm is proposed, incorporating a preprocessing block based on a Wavelet-based Autoencoder (WAE) for removing haze and rain from video frames. The algorithm utilizes FastSAM, an unsupervised object detection model, to identify objects within a specified search region. A novel similarity matching technique is introduced, it achieves superior performance in terms of SSIM achieves superior performance compared to traditional cosine similarity. A new synthetic dataset is also proposed for model evaluation, consisting of 20 videos for rainy conditions and 20 videos for hazy conditions. The proposed model achieves a success rate of 51.20% and a precision of 54.71% on hazy videos, as well as a success rate of 57.31% and a precision of 49.02% on rainy videos.

7.2 Limitations

This thesis presents an effective solution for object tracking in hazy and rainy weather conditions. To address the lack of existing datasets for training and testing object tracking models under adverse weather, a new dataset was specifically developed. However, this work has some limitations. First, the proposed model focuses solely on haze and rain degradation, excluding other challenging weather conditions such as sandstorms and snowfall. Additionally, the approach relies on supervised learning, which necessitates high-quality ground truth data for training. In real-world scenarios, obtaining perfectly clear images corresponding to every hazy or rainy scene is often impractical. Future research should explore semi-supervised or unsupervised learning techniques to mitigate this dependency and enhance the model's adaptability to diverse weather conditions.

Despite its strong performance compared to existing dehazing and deraining models in both qualitative and quantitative evaluations, the proposed method has certain drawbacks. One notable limitation is the presence of artifacts in the results when applied to real-world hazy and rainy images, which may affect object tracking accuracy. The method employs all four wavelet components to enhance image quality, leading to increased computational complexity. Future work will investigate alternative feature selection strategies within the wavelet domain to develop a more efficient model while maintaining performance. Moreover, beyond image enhancement, integrating the model into object detection and tracking pipelines as a pre-processing step could further improve tracking robustness under adverse weather conditions.

Another limitation lies in the dataset itself. The developed dataset primarily consists of synthetically generated hazy and rainy images, which may limit the model's generalization to real-world scenarios. Additionally, the dataset is designed for single-object tracking and does not support multi-object tracking in adverse weather, restricting its applicability to more complex tracking tasks. Furthermore, the final model employs a zero-shot method, which results in reduced accuracy. With a processing speed of only 11 FPS, the model requires optimization for real-time deployment. Moreover, the high computational cost of using SAM for object detection highlights the need for efficiency improvements to enable broader practical applications.

7.3 Future Scope

In the light of the developments presented in this thesis, this section outlines future directions and open research issues.

- New datasets can be developed for multi-object tracking in adverse weather conditions.
- Other degraded weather conditions such as sandstorms and heavy snowfall can be taken into consideration for future planning.
- The *ArcTrack* consists of two sub-blocks: an image restoration model and an object tracking model. In the future, a single model can be developed for simultaneous image restoration and tracking.
- Even though the developed image restoration GAN-based model is now running at 11 frames per second, real-time performance may still be improved, which will improve the system's overall processing time.
- To improve object tracking capabilities, a supervised model may be created and evaluated using the suggested dataset.

In a nutshell, this thesis provides a comprehensive study of the challenges associated with object tracking in adverse weather conditions. It successfully develops a preprocessing method capable of simultaneously removing haze and rain, effectively addressing the scale-space tracking problem. Additionally, the thesis introduces a novel dataset designed to assist future researchers in evaluating their models under challenging weather scenarios. A robust baseline is also proposed, serving as a valuable reference point for further advancements in the field. These contributions collectively enhance the understanding and performance of object-tracking systems in complex environmental conditions.

References

[1] Chi-Huang Shih, Cheng-Jian Lin, Ta-Sen Wei, Peng-Ta Liu, and Ching-Yu Shih. Behavior analysis based on local object tracking and its bed-exit application. In *2021 IEEE 4th International Conference on Knowledge Innovation and Invention (ICKII)*, pages 101–104, 2021. doi: 10.1109/ICKII51822.2021.9574741.

[2] Gyuyeong Kim, Hyuntae Kim, Jangsik Park, and Yunsik Yu. Vehicle tracking based on kalman filter in tunnel. In Tai-hoon Kim, Hojjat Adeli, Rosslin John Robles, and Maricel Balitanas, editors, *Information Security and Assurance*, pages 250–256, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-23141-4.

[3] Hongxia Chu, Zhongyu Xie, Xiangju Nie, Zhanying Li, and Xin Li. Particle filter target tracking method optimized by improved mean shift. In *2013 IEEE International Conference on Information and Automation (ICIA)*, pages 991–994, 2013. doi: 10.1109/ICInfA.2013.6720439.

[4] Sian Barris and Chris Button. A review of vision-based motion analysis in sport. *Sports medicine*, 38:1025–1043, 2008.

[5] Jaime Gallego, Montse Pardas, and Jose-Luis Landabaso. Segmentation and tracking of static and moving objects in video surveillance scenarios. In *2008 15th IEEE International Conference on Image Processing*, pages 2716–2719, 2008. doi: 10.1109/ICIP.2008.4712355.

[6] Qiang Wang, Jin Gao, Junliang Xing, Mengdan Zhang, and Weiming Hu. Dcfnet: Discriminant correlation filters network for visual tracking. *arXiv e-prints*, pages arXiv–1704, 2017.

[7] Jack Valmadre, Luca Bertinetto, João Henriques, Andrea Vedaldi, and Philip H. S. Torr. End-to-end representation learning for correlation filter

based tracking. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5000–5008, 2017. doi: 10.1109/CVPR.2017.531.

[8] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II 14*, pages 850–865. Springer, 2016.

[9] Ivan Saetchnikov, Victor Skakun, and Elina Tcherniavskaia. Efficient objects tracking from an unmanned aerial vehicle. In *2021 IEEE 8th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, pages 221–225, 2021. doi: 10.1109/MetroAeroSpace51421.2021.9511748.

[10] Angus Leigh, Joelle Pineau, Nicolas Olmedo, and Hong Zhang. Person tracking and following with 2d laser scanners. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 726–733, 2015. doi: 10.1109/ICRA.2015.7139259.

[11] Qiao Liu, Xin Li, Di Yuan, Chao Yang, Xiaojun Chang, and Zhenyu He. Lsotb-tir: A large-scale high-diversity thermal infrared single object tracking benchmark. *IEEE Transactions on Neural Networks and Learning Systems*, 35(7):9844–9857, 2024. doi: 10.1109/TNNLS.2023.3236895.

[12] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5320–5329, 2017. doi: 10.1109/CVPR.2017.565.

[13] Zahra Soleimanitaleb, Mohammad Ali Keyvanrad, and Ali Jafari. Object tracking methods:a review. In *2019 9th International Conference on Computer and Knowledge Engineering (ICCKE)*, pages 282–288, 2019. doi: 10.1109/ICCKE48569.2019.8964761.

[14] V. Arthi, R. Murugeswari, and Nagaraj P. Object detection of autonomous vehicles under adverse weather conditions. In *2022 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI)*, volume 01, pages 1–8, 2022. doi: 10.1109/ICDSAAI55433.2022.10028795.

[15] Isaac Oluwadunsin Ogunrinde, Simon Y., Foo, Doreen C., Kobelo, and Rodney G, Roberts. *Multi-Sensor Fusion for Object Detection and Tracking Under Foggy Weather Conditions*. PhD thesis, 2023. AAI29395453.

[16] Zihan Chu. D-yolo a robust framework for object detection in adverse weather conditions. *arXiv preprint arXiv:2403.09233*, 2024.

[17] Debasis Kumar and Naveed Muhammad. Object detection in adverse weather for autonomous driving through data merging and yolov8. *Sensors*, 23(20), 2023. ISSN 1424-8220. doi: 10.3390/s23208471. URL `https://www.mdpi.com/1424-8220/23/20/8471`.

[18] D. Comaniciu and V. Ramesh. Mean shift and optimal prediction for efficient object tracking. In *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)*, volume 3, pages 70–73 vol.3, 2000. doi: 10.1109/ICIP.2000.899297.

[19] Xi Chen, Xiao Wang, and Jianhua Xuan. Tracking multiple moving objects using unscented kalman filtering techniques. *arXiv preprint arXiv:1802.01235*, 2018.

[20] Jung Uk Cho, Seung Hun Jin, Xuan Dai Pham, Jae Wook Jeon, Jong Eun Byun, and Hoon Kang. A real-time object tracking system using a particle filter. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2822–2827, 2006. doi: 10.1109/IROS.2006.282066.

[21] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[22] Asfak Ali, Avra Ghosh, and Sheli Sinha Chaudhuri. Lidn: A novel light invariant image dehazing network. *Engineering Applications of Artificial Intelligence*, 126:106830, 2023. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2023.106830. URL `https://www.sciencedirect.com/science/article/pii/S095219762301014X`.

[23] 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. doi: 10.1109/TIP.2018.2867951.

[24] Avra Ghosh, Asfak Ali, Sangita Roy, and Sheli Sinha Chaudhuri. Novel parametric based time efficient portable real-time dehazing system. *Journal of Real-Time Image Processing*, 20(2):23, 2023.

[25] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[27] Rui Li, Shunyi Zheng, Ce Zhang, Chenxi Duan, Jianlin Su, Libo Wang, and Peter M Atkinson. Multiattention network for semantic segmentation of fine-resolution remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2021.

[28] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[29] Alireza Esmaeilzehi, M. Omair Ahmad, and M. N. S. Swamy. Fpnet: A deep light-weight interpretable neural network using forward prediction filtering for efficient single image super resolution. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 69(3):1937–1941, 2022. doi: 10.1109/TCSII.2021.3121667.

[30] Cosmin Ancuti, Codruta O Ancuti, Radu Timofte, and Christophe De Vleeschouwer. I-haze: a dehazing benchmark with real hazy and haze-free indoor images. In *Advanced Concepts for Intelligent Vision Systems: 19th International Conference, ACIVS 2018, Poitiers, France, September 24–27, 2018, Proceedings 19*, pages 620–631. Springer, 2018.

[31] Javier Hidalgo-Carrió, Daniel Gehrig, and Davide Scaramuzza. Learning monocular dense depth from events. In *2020 International Conference on 3D Vision (3DV)*, pages 534–542. IEEE, 2020.

[32] Codruta O Ancuti, Cosmin Ancuti, and Radu Timofte. Nh-haze: An image dehazing benchmark with non-homogeneous hazy and haze-free images. In

Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, pages 444–445, 2020.

[33] Asfak Ali, Ram Sarkar, and Sheli Sinha Chaudhuri. Wavelet-based auto-encoder for simultaneous haze and rain removal from images. *Pattern Recognition*, 150:110370, 2024.

[34] He Zhang and Vishal M. Patel. Density-aware single image de-raining using a multi-stream dense network. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 695–704, 2018. doi: 10.1109/CVPR.2018.00079.

[35] Kaihao Zhang, Wenhan Luo, Yanjiang Yu, Wenqi Ren, Fang Zhao, Changsheng Li, Lin Ma, Wei Liu, and Hongdong Li. Beyond monocular deraining: Parallel stereo deraining network via semantic prior. *International Journal of Computer Vision*, 130(7):1754–1769, Jul 2022. ISSN 1573-1405. doi: 10.1007/s11263-022-01620-w. URL `https://doi.org/10.1007/s11263-022-01620-w`.

[36] Kaihao Zhang, Dongxu Li, Wenhan Luo, Wenqi Ren, Lin Ma, and Hongdong Li. Dual attention-in-attention model for joint rain streak and raindrop removal. *IEEE Transactions on Image Processing*, 30:7608–7619, 2021. URL `https://api.semanticscholar.org/CorpusID:232222608`.

[37] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[38] He Zhang, Vishwanath Sindagi, and Vishal M. Patel. Image de-raining using a conditional generative adversarial network. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(11):3943–3956, 2020. doi: 10.1109/TCSVT.2019.2920407.

[39] Siyuan Li, Iago Breno Araujo, Wenqi Ren, Zhangyang Wang, Eric K. Tokuda, Roberto Hirata Junior, Roberto Cesar-Junior, Jiawan Zhang, Xiaojie Guo, and Xiaochun Cao. Single image deraining: A comprehensive benchmark analysis. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3833–3842, 2019. doi: 10.1109/CVPR.2019.00396.

[40] T. Matsuzaki, H. Kameda, S. Tsujimichi, and K. Kosuge. Maneuvering target tracking using constant velocity and constant angular velocity model. In *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics. 'cybernetics evolving to systems, humans, organizations, and their complex interactions' (cat. no.0*, volume 5, pages 3230–3234 vol.5, 2000. doi: 10.1109/ICSMC.2000.886501.

[41] Pramod R. Gunjal, Bhagyashri R. Gunjal, Haribhau A. Shinde, Swapnil M. Vanam, and Sachin S. Aher. Moving object tracking using kalman filter. In *2018 International Conference On Advances in Communication and Computing Technology (ICACCT)*, pages 544–547, 2018. doi: 10.1109/ICACCT.2018.8529402.

[42] Qiang Li, Ranyang Li, Kaifan Ji, and Wei Dai. Kalman filter and its application. In *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, pages 74–77, 2015. doi: 10.1109/ICINIS.2015.35.

[43] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[44] Alexander Sibiryakov. Fast and high-performance template matching method. pages 1417–1424, 06 2011. doi: 10.1109/CVPR.2011.5995391.

[45] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *CoRR*, abs/1511.08458, 2015. URL `http://arxiv.org/abs/1511.08458`.

[46] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. doi: 10.1162/neco.1997.9.8.1735.

[47] S. Avidan. Support vector tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1064–1072, 2004. doi: 10.1109/TPAMI.2004.53.

[48] Hongyan Wang, Helei Qiu, and Wenshu Li. Nonconvex dictionary learning based visual tracking method. *Signal Processing*, 172:107535, 2020. ISSN 0165-1684. doi: https://doi.org/10.1016/j.sigpro.2020.107535. URL `https://www.sciencedirect.com/science/article/pii/S0165168420300785`.

[49] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002. doi: 10.1109/34.1000236.

[50] R.T. Collins. Mean-shift blob tracking through scale space. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–234, 2003. doi: 10.1109/CVPR.2003.1211475.

[51] Shailesh Kamble, Nileshsingh Thakur, Apurva Samdurkar, and Akshay Patharkar. Object detection and tracking using modified diamond search block matching motion estimation algorithm. *International Journal of Interactive Multimedia and Artificial Intelligence*, 5 (1):73–85, 06/2018 2018. ISSN 1989-1660. doi: 10.9781/ijimai.2017.10.004. URL `http://www.ijimai.org/journal/sites/default/files/files/2017/10/ijimai_5_1_10_pdf_12412.pdf`.

[52] Gyuyeong Kim, Hyuntae Kim, Jangsik Park, and Yunsik Yu. Vehicle tracking based on kalman filter in tunnel. pages 250–256, 08 2011. ISBN 978-3-642-23140-7. doi: 10.1007/978-3-642-23141-4_24. URL `http://dx.doi.org/10.1007/978-3-642-23141-4_24`.

[53] Iman Iraei and Karim Faez. Object tracking with occlusion handling using mean shift, kalman filter and edge histogram. In *2015 2nd International Conference on Pattern Recognition and Image Analysis (IPRIA)*, pages 1–6, 2015. doi: 10.1109/PRIA.2015.7161637.

[54] Hongxia Chu, Zhongyu Xie, Xiangju Nie, Zhanying Li, and Xin Li. Particle filter target tracking method optimized by improved mean shift. In *2013 IEEE International Conference on Information and Automation (ICIA)*, pages 991–994, 2013. doi: 10.1109/ICInfA.2013.6720439.

[55] O.-D. Nouar, G. Ali, and C. Raphael. Improved object tracking with camshift algorithm. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 2, pages II–II, 2006. doi: 10.1109/ICASSP.2006.1660428.

[56] Siyi Li and Dit-Yan Yeung. Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models. *Proceedings of the AAAI*

Conference on Artificial Intelligence, 31(1), Feb. 2017. URL https://ojs.aaai.org/index.php/AAAI/article/view/11205.

[57] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13–es, 2006.

[58] Bo Li. An improved bernoulli particle filter for single target tracking. *Multidimensional Syst. Signal Process.*, 29(3):799–819, jul 2018. ISSN 0923-6082. doi: 10.1007/s11045-017-0471-2. URL https://doi.org/10.1007/s11045-017-0471-2.

[59] Bohyung Han, Dorin Comaniciu, Ying Zhu, and Larry S. Davis. Sequential kernel density approximation and its application to real-time visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7): 1186–1197, 2008. doi: 10.1109/TPAMI.2007.70771.

[60] Nikos Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *International Journal of Computer Vision*, 46:223–247, 02 2002. doi: 10.1023/A:1014080923068.

[61] Xue Mei, Haibin Ling, Yi Wu, Erik Blasch, and Li Bai. Minimum error bounded efficient l1 tracker with occlusion detection. In *CVPR 2011*, pages 1257–1264, 2011. doi: 10.1109/CVPR.2011.5995421.

[62] Naiyan Wang, Jingdong Wang, and Dit-Yan Yeung. Online robust non-negative dictionary learning for visual tracking. In *2013 IEEE International Conference on Computer Vision*, pages 657–664, 2013. doi: 10.1109/ICCV.2013.87.

[63] B. Stenger, V. Ramesh, N. Paragios, F. Coetzee, and J.M. Buhmann. Topology free hidden markov models: application to background modeling. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, pages 294–301 vol.1, 2001. doi: 10.1109/ICCV.2001.937532.

[64] Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. *Introduction to linear regression analysis*. John Wiley & Sons, 2021.

[65] Antonio Brunetti, Domenico Buongiorno, Gianpaolo Francesco Trotta, and Vitoantonio Bevilacqua. Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. *Neurocomputing*, 300:17–33, 2018.

[66] Helmut Grabner, Christian Leistner, and Horst Bischof. Semi-supervised on-line boosting for robust tracking. In *European conference on computer vision*, pages 234–247. Springer, 2008.

[67] Shai Avidan. Ensemble tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):261–271, 2007. doi: 10.1109/TPAMI.2007.35.

[68] Qinxun Bai, Zheng Wu, Stan Sclaroff, Margrit Betke, and Camille Monnier. Randomized ensemble tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2040–2047, 2013.

[69] Naiyan Wang and Dit-Yan Yeung. Ensemble-based tracking: Aggregating crowdsourced structured time series data. In *International Conference on Machine Learning*, pages 1107–1115. PMLR, 2014.

[70] Zdenek Kalal, Jiri Matas, and Krystian Mikolajczyk. Pn learning: Bootstrapping binary classifiers by structural constraints. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 49–56. IEEE, 2010.

[71] Yingjie Yao, Xiaohe Wu, Lei Zhang, Shiguang Shan, and Wangmeng Zuo. Joint representation and truncated inference learning for correlation filter based tracking. *CoRR*, abs/1807.11071, 2018. URL `http://arxiv.org/abs/1807.11071`.

[72] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 2544–2550. IEEE, 2010.

[73] Joao F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *European conference on computer vision*, pages 702–715. Springer, 2012.

[74] Erhan Gundogdu and A. Aydin Alatan. Good features to correlate for visual tracking. *CoRR*, abs/1704.06326, 2017. URL `http://arxiv.org/abs/1704.06326`.

[75] Xiaoyu Wang, Tony X Han, and Shuicheng Yan. An hog-lbp human detector with partial occlusion handling. In *2009 IEEE 12th international conference on computer vision*, pages 32–39. IEEE, 2009.

[76] Seunghoon Hong, Tackgeun You, Suha Kwak, and Bohyung Han. Online tracking by learning discriminative saliency map with convolutional neural network. *CoRR*, abs/1502.06796, 2015. URL `http://arxiv.org/abs/1502.06796`.

[77] Guibo Zhu, Jinqiao Wang, Yi Wu, and Hanqing Lu. Collaborative correlation tracking. In *BMVC*, pages 184–1, 2015.

[78] Honghong Yang, Sheng Gao, Xiaojun Wu, and Yumei Zhang. Online multi-object tracking using kcf-based single-object tracker with occlusion analysis. *Multimedia Syst.*, 26(6):655–669, dec 2020. ISSN 0942-4962. doi: 10.1007/s00530-020-00675-4. URL `https://doi.org/10.1007/s00530-020-00675-4`.

[79] Mathew Francis and Prithwijit Guha. Siamese fully convolutional tracker with motion correction. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2218–2225, 2021. doi: 10.1109/ICPR48806.2021.9412986.

[80] Yejin Yan, Wenxiao Huo, Jiayu Ou, Zhifeng Liu, and Tianping Li. Improved siamfc target tracking algorithm based on anti-interference module. *Journal of Sensors*, 2022, 2022.

[81] Chang Gao, Junkun Yan, Shenghua Zhou, Pramod K Varshney, and Hongwei Liu. Long short-term memory-based deep recurrent neural networks for target tracking. *Information Sciences*, 502:279–296, 2019.

[82] Dalei Qiao, Guangzhong Liu, Taizhi Lv, Wei Li, and Juan Zhang. Marine vision-based situational awareness using discriminative deep learning: A survey. *Journal of Marine Science and Engineering*, 9(4):397, 2021.

[83] Zhenjun Han, Pan Wang, and Qixiang Ye. Adaptive discriminative deep correlation filter for visual object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(1):155–166, 2018.

[84] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Forward-backward error: Automatic detection of tracking failures. In *2010 20th International*

Conference on Pattern Recognition, pages 2756–2759, 2010. doi: 10.1109/ICPR.2010.675.

[85] Kaihua Zhang, Lei Zhang, and Ming-Hsuan Yang. Fast compressive tracking. *IEEE transactions on pattern analysis and machine intelligence*, 36(10): 2002–2015, 2014.

[86] Chenglong Bao, Yi Wu, Haibin Ling, and Hui Ji. Real time robust l1 tracker using accelerated proximal gradient approach. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1830–1837, 2012. doi: 10.1109/CVPR.2012.6247881.

[87] Eli G. Pale-Ramon, Luis J. Morales-Mendoza, Mario González-Lee, Oscar G. Ibarra-Manzano, Jorge A. Ortega-Contreras, and Yuriy S. Shmaliy. Improving visual object tracking using general ufir and kalman filters under disturbances in bounding boxes. *IEEE Access*, 11:57905–57915, 2023. doi: 10.1109/ACCESS.2023.3280420.

[88] Matthias Mueller, Neil Smith, and Bernard Ghanem. Context-aware correlation filter tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[89] Jungsup Shin, Heegwang Kim, Dohun Kim, and Joonki Paik. Fast and robust object tracking using tracking failure detection in kernelized correlation filter. *Applied Sciences*, 10(2), 2020. ISSN 2076-3417. doi: 10.3390/app10020713. URL `https://www.mdpi.com/2076-3417/10/2/713`.

[90] Kaihua Zhang, Lei Zhang, Qingshan Liu, Dapeng Zhang, and Ming Hsuan Yang. Fast visual tracking via dense spatio-temporal context learning. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8693 LNCS(PART 5): 127–141, January 2014. ISSN 0302-9743. doi: 10.1007/978-3-319-10602-1_9. 13th European Conference on Computer Vision, ECCV 2014 ; Conference date: 06-09-2014 Through 12-09-2014.

[91] Yihong Zhang, Yijin Yang, Wuneng Zhou, Lifeng Shi, and Demin Li. Motion-aware correlation filters for online visual tracking. *Sensors*, 18:3937, 11 2018. doi: 10.3390/s18113937.

[92] Khizer Mehmood, Ahmad Ali, Abdul Jalil, Baber Khan, Khalid Mehmood Cheema, Maria Murad, and Ahmad H. Milyani. Efficient online object tracking scheme for challenging scenarios. *Sensors*, 21(24), 2021. ISSN 1424-8220. doi: 10.3390/s21248481. URL `https://www.mdpi.com/1424-8220/21/24/8481`.

[93] Baber Khan, Ahmad Ali, Abdul Jalil, Khizer Mehmood, Maria Murad, and Hamdan Awan. Afam-pec: Adaptive failure avoidance tracking mechanism using prediction-estimation collaboration. *IEEE Access*, 8:149077–149092, 2020. doi: 10.1109/ACCESS.2020.3015580.

[94] Khizer Mehmood, Abdul Jalil, Ahmad Ali, Baber Khan, Maria Murad, Wasim Ullah Khan, and Yigang He. Context-aware and occlusion handling mechanism for online visual object tracking. *Electronics*, 10(1), 2021. ISSN 2079-9292. doi: 10.3390/electronics10010043. URL `https://www.mdpi.com/2079-9292/10/1/43`.

[95] Zhenyang Li, Ran Tao, Efstratios Gavves, Cees G. M. Snoek, and Arnold W.M. Smeulders. Tracking by natural language specification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[96] Yihao Luo, Min Xu, Caihong Yuan, Xiang Cao, Liangqi Zhang, Yan Xu, Tianjiang Wang, and Qi Feng. Siamsnn: Siamese spiking neural networks for energy-efficient object tracking. In *International Conference on Artificial Neural Networks*, 2020.

[97] Yinda Xu, Zeyu Wang, Zuoxin Li, Ye Yuan, and Gang Yu. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34: 12549–12556, 04 2020. doi: 10.1609/aaai.v34i07.6944.

[98] Qi Feng, Vitaly Ablavsky, Qinxun Bai, and Stan Sclaroff. Siamese natural language tracker: Tracking by natural language descriptions with siamese trackers. pages 5847–5856, 06 2021. doi: 10.1109/CVPR46437.2021.00579.

[99] Xiao Wang, Xiujun Shu, Zhipeng Zhang, Bo Jiang, Yaowei Wang, Yonghong Tian, and Feng Wu. Towards more flexible and accurate object tracking with natural language: Algorithms and benchmark. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 13763–13773, June 2021.

[100] Qi Feng, Vitaly Ablavsky, Qinxun Bai, Guorong Li, and Stan Sclaroff. Real-time visual object tracking with natural language description. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[101] Shuiying Xiang, Tao Zhang, Shuqing Jiang, Yanan Han, Yahui Zhang, Chenyang Du, Xingxing Guo, Licun Yu, Yuechun Shi, and Yue Hao. Spiking siamfc++: Deep spiking neural network for object tracking, 2022.

[102] Mingzhe Guo, Zhipeng Zhang, Heng Fan, and Liping Jing. Divert more attention to vision-language tracking. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 4446–4460. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/1c8c87c36dc1e49e63555f95fa56b153-Paper-Conference.pdf`.

[103] Jiahao Bao, Menglong Yan, Yiran Yang, and Kaiqiang Chen. Siamffn: Siamese feature fusion network for visual tracking. *Electronics*, 12(7), 2023. ISSN 2079-9292. doi: 10.3390/electronics12071568. URL `https://www.mdpi.com/2079-9292/12/7/1568`.

[104] Li Zhou, Zikun Zhou, Kaige Mao, and Zhenyu He. Joint visual grounding and tracking with natural language specification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23151–23160, June 2023.

[105] Liang Li, Wei Feng, and Jiawan Zhang. Contrast enhancement based single image dehazing via tv-l1 minimization. In *2014 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2014. doi: 10.1109/ICME.2014.6890277.

[106] John P. Oakley and Hong Bu. Correction of simple contrast loss in color images. *IEEE Transactions on Image Processing*, 16(2):511–522, 2007. doi: 10.1109/TIP.2006.887736.

[107] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2341–2353, 2011. doi: 10.1109/TPAMI.2010.168.

[108] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, 2013. doi: 10.1109/TPAMI.2012.213.

[109] Qingsong Zhu, Jiaming Mai, and Ling Shao. A fast single image haze removal algorithm using color attenuation prior. *IEEE Transactions on Image Processing*, 24(11):3522–3533, 2015. doi: 10.1109/TIP.2015.2446191.

[110] Sriparna Banerjee and Sheli Chaudhuri. Nighttime image-dehazing: A review and quantitative benchmarking. *Archives of Computational Methods in Engineering*, 09 2020. doi: 10.1007/s11831-020-09485-3.

[111] Soo-Chang Pei and Tzu-Yen Lee. Nighttime haze removal using color transfer pre-processing and dark channel prior. *Proceedings / ICIP ... International Conference on Image Processing*, pages 957–960, 09 2012. doi: 10.1109/ICIP.2012.6467020.

[112] Yun Liu, Zhongsheng Yan, Jinge Tan, and Yuche Li. Multi-purpose oriented single nighttime image haze removal based on unified variational retinex model. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2022. doi: 10.1109/TCSVT.2022.3214430.

[113] Wenhui Wang, Anna Wang, and Chen Liu. Variational single nighttime image haze removal with a gray haze-line prior. *IEEE Transactions on Image Processing*, 31:1349–1363, 2022. doi: 10.1109/TIP.2022.3141252.

[114] Cosmin Ancuti, Codruta O. Ancuti, Christophe De Vleeschouwer, and Alan C. Bovik. Day and night-time dehazing by local airlight estimation. *IEEE Transactions on Image Processing*, 29:6264–6275, 2020. doi: 10.1109/TIP.2020.2988203.

[115] H. Koschmieder. Theorie der horizontalensichtweite. *Beitr. Phys. Freien Atm., vol. 12*, pages 171–181, 09 1924.

[116] J.P. Oakley and B.L. Satherley. Improving image quality in poor visibility conditions using a physical model for contrast degradation. *IEEE Transactions on Image Processing*, 7(2):167–179, 1998. doi: 10.1109/83.660994.

[117] Y.Y. Schechner, S.G. Narasimhan, and S.K. Nayar. Instant dehazing of images using polarization. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, 2001. doi: 10.1109/CVPR.2001.990493.

[118] S.G. Narasimhan and S.K. Nayar. Contrast restoration of weather degraded images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):713–724, 2003. doi: 10.1109/TPAMI.2003.1201821.

[119] S. Shwartz, E. Namer, and Y.Y. Schechner. Blind haze separation. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1984–1991, 2006. doi: 10.1109/CVPR.2006.71.

[120] Robby T. Tan. Visibility in bad weather from a single image. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. doi: 10.1109/CVPR.2008.4587643.

[121] Raanan Fattal. Single image dehazing. *ACM transactions on graphics (TOG)*, 27(3):1–9, 2008. doi: https://doi.org/10.1145/1360612.1360671.

[122] 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. doi: 10.1109/ICCV.2009.5459251.

[123] Bolun Cai, Xiangmin Xu, Kui Jia, Chunmei Qing, and Dacheng Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing*, 25(11):5187–5198, 2016. doi: 10.1109/TIP.2016.2598681.

[124] Patricia L. Suárez, Angel D. Sappa, Boris X. Vintimilla, and Riad I. Hammoud. Deep learning based single image dehazing. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1250–12507, 2018. doi: 10.1109/CVPRW.2018.00162.

[125] Codruta Orniana Ancuti and Cosmin Ancuti. Single image dehazing by multi-scale fusion. *IEEE Transactions on Image Processing*, 22(8):3271–3282, 2013. doi: 10.1109/TIP.2013.2262284.

[126] Yu Li, Robby T. Tan, and Michael S. Brown. Nighttime haze removal with glow and multiple light colors. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 226–234, 2015. doi: 10.1109/ICCV.2015.34.

[127] Dana Berman, Tali Treibitz, and Shai Avidan. Non-local image dehazing. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1674–1682, 2016. doi: 10.1109/CVPR.2016.185.

[128] Zhan Li, Xiaopeng Zheng, Bir Bhanu, Shun Long, Qingfeng Zhang, and Zhenghao Huang. Fast region-adaptive defogging and enhancement for outdoor images containing sky. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 8267–8274, 2021. doi: 10.1109/ICPR48806.2021.9412595.

[129] Teng Yu, Kang Song, Pu Miao, Guowei Yang, Huan Yang, and Chenglizhao Chen. Nighttime single image dehazing via pixel-wise alpha blending. *IEEE Access*, 7:114619–114630, 2019. doi: 10.1109/ACCESS.2019.2936049.

[130] Mingye Ju, Can Ding, Wenqi Ren, Yi Yang, Dengyin Zhang, and Y. Jay Guo. Ide: Image dehazing and exposure using an enhanced atmospheric scattering model. *IEEE Transactions on Image Processing*, 30:2180–2192, 2021. doi: 10.1109/TIP.2021.3050643.

[131] Qingsong Zhu, Jiaming Mai, and Ling Shao. A fast single image haze removal algorithm using color attenuation prior. *IEEE Transactions on Image Processing*, 24(11):3522–3533, 2015. doi: 10.1109/TIP.2015.2446191.

[132] Javier Diaz-Cely, Carlos Arce-Lopera, Juan Cardona Mena, and Lina Quintero. The effect of color channel representations on the transferability of convolutional neural networks. In Kohei Arai and Supriya Kapoor, editors, *Advances in Computer Vision*, pages 27–38, Cham, 2020. Springer International Publishing. ISBN 978-3-030-17795-9.

[133] Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, page III–1319–III–1327. JMLR.org, 2013.

[134] Codruta O. Ancuti, Cosmin Ancuti, Radu Timofte, and Christophe De Vleeschouwer. I-haze: a dehazing benchmark with real hazy and haze-free indoor images. In *arXiv:1804.05091v1*, 2018.

[135] Codruta O. Ancuti, Cosmin Ancuti, Radu Timofte, and Christophe De Vleeschouwer. O-haze: A dehazing benchmark with real hazy and haze-free outdoor images. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 867–8678, 2018. doi: 10.1109/CVPRW.2018.00119.

[136] Codruta O. Ancuti, Cosmin Ancuti, and Radu Timofte. NH-HAZE: an image dehazing benchmark with non-homogeneous hazy and haze-free images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, IEEE CVPR 2020, 2020.

[137] Codruta O. Ancuti, Cosmin Ancuti, Mateu Sbert, and Radu Timofte. Dense haze: A benchmark for image dehazing with dense-haze and haze-free images. In *IEEE International Conference on Image Processing (ICIP)*, IEEE ICIP 2019, 2019.

[138] Sriparna Banerjee, Pranay Singha, Pritam Ghosh, and Sheli Chaudhuri. Daytime and nighttime dehazing benchmarking database. In *Diary Number: 8997/2021-CO/L (Registered Copyright), Registration No. L-103093/2021*, 05 2021.

[139] Claude Sammut and Geoffrey I. Webb, editors. *Mean Squared Error*, pages 653–653. Springer US, Boston, MA, 2010. ISBN 978-0-387-30164-8. doi: 10.1007/978-0-387-30164-8_528.

[140] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369, 2010. doi: 10.1109/ICPR.2010.579.

[141] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. A comprehensive evaluation of full reference image quality assessment algorithms. In *2012 19th IEEE International Conference on Image Processing*, pages 1477–1480, 2012. doi: 10.1109/ICIP.2012.6467150.

[142] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

[143] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar*

Conference on Signals, Systems & Computers, 2003, volume 2, pages 1398–1402. Ieee, 2003.

[144] Qingsong Zhu, Jiaming Mai, and Ling Shao. A fast single image haze removal algorithm using color attenuation prior. *IEEE Transactions on Image Processing*, 24(11):3522–3533, 2015. doi: 10.1109/TIP.2015.2446191.

[145] Boyi Li, Xiulian Peng, Zhangyang Wang, Jizheng Xu, and Dan Feng. An all-in-one network for dehazing and beyond. *CoRR*, abs/1707.06543, 2017.

[146] Vladimir Frants, Sos Agaian, and Karen Panetta. Qcnn-h: Single-image dehazing using quaternion neural networks. *IEEE Transactions on Cybernetics*, pages 1–11, 2023. doi: 10.1109/TCYB.2023.3238640.

[147] Le-Anh Tran, Seokyong Moon, and Dong-Chul Park. A novel encoder-decoder network with guided transmission map for single image dehazing. *Procedia Computer Science*, 204:682–689, 2022. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2022.08.082. International Conference on Industry Sciences and Computer Science Innovation.

[148] Wenqi Ren, Lin Ma, Jiawei Zhang, Jinshan Pan, Xiaochun Cao, Wei Liu, and Ming-Hsuan Yang. Gated fusion network for single image dehazing. *CoRR*, abs/1804.00213, 2018.

[149] Wenqi Ren, Si Liu, Hua Zhang, Jinshan Pan, Xiaochun Cao, and Ming-Hsuan Yang. Single image dehazing via multi-scale convolutional neural networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 154–169, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46475-6.

[150] Dongdong Chen, Mingming He, Qingnan Fan, Jing Liao, Liheng Zhang, Dongdong Hou, Lu Yuan, and Gang Hua. Gated context aggregation network for image dehazing and deraining. *CoRR*, abs/1811.08747, 2018.

[151] Xiaohong Liu, Yongrui Ma, Zhihao Shi, and Jun Chen. Griddehazenet: Attention-based multi-scale network for image dehazing. *CoRR*, abs/1908.03245, 2019.

[152] Hang Dong, Jinshan Pan, Lei Xiang, Zhe Hu, Xinyi Zhang, Fei Wang, and Ming-Hsuan Yang. Multi-scale boosted dehazing network with dense feature fusion. In *2020 IEEE/CVF Conference on Computer Vision and Pattern*

Recognition (CVPR), pages 2154–2164, 2020. doi: 10.1109/CVPR42600.2020.00223.

[153] Jiangxin Dong and Jinshan Pan. Physics-based feature dehazing networks. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX*, page 188–204, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58576-1. doi: 10.1007/978-3-030-58577-8_12.

[154] Xu Qin, Zhilin Wang, Yuanchao Bai, Xiaodong Xie, and Huizhu Jia. Ffa-net: Feature fusion attention network for single image dehazing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):11908–11915, Apr. 2020. doi: 10.1609/aaai.v34i07.6865.

[155] Haiyan Wu, Yanyun Qu, Shaohui Lin, Jian Zhou, Ruizhi Qiao, Zhizhong Zhang, Yuan Xie, and Lizhuang Ma. Contrastive learning for compact single image dehazing. *CoRR*, abs/2104.09367, 2021.

[156] Yuda Song, Zhuqing He, Hui Qian, and Xin Du. Vision transformers for single image dehazing. *arXiv preprint arXiv:2204.03883*, 2022.

[157] Codruta O Ancuti, Cosmin Ancuti, Radu Timofte, Luc Van Gool, Lei Zhang, and Ming-Hsuan Yang. Ntire 2019 image dehazing challenge report. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, IEEE CVPR 2019, 2019.

[158] Avra Ghosh, Sangita Roy, and Sheli Sinha Chaudhuri. Hardware implementation of image dehazing mechanism using verilog hdl and parallel dcp. In *2020 IEEE Applied Signal Processing Conference (ASPCON)*, pages 283–287, 2020. doi: 10.1109/ASPCON49795.2020.9276702.

[159] Avra Ghosh and Sheli Sinha Chaudhuri. Iot based portable image dehazing machine. In *2021 8th International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 31–35, 2021. doi: 10.1109/SPIN52536.2021.9565998.

[160] Yücel Çimtay. Smart and real-time image dehazing on mobile devices. *Journal of Real-Time Image Processing*, 18:1–10, 12 2021. doi: 10.1007/s11554-021-01085-z.

[161] Qingsong Zhu, Jiaming Mai, and Ling Shao. A fast single image haze removal algorithm using color attenuation prior. *IEEE Transactions on Image Processing*, 24(11):3522–3533, 2015. doi: 10.1109/TIP.2015.2446191.

[162] Wei-Ling Chen, Chung-Dann Kan, Chia-Hung Lin, Ying-Shin Chen, and Yi-Chen Mai. Hypervolemia screening in predialysis healthcare for hemodialysis patients using fuzzy color reason analysis. *International Journal of Distributed Sensor Networks*, 13:155014771668509, 01 2017. doi: 10.1177/1550147716685090.

[163] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[164] Sungmin Lee, Seokmin Yun, Ju-Hun Nam, Chee Sun Won, and Seung-Won Jung. A review on dark channel prior based image dehazing algorithms. *EURASIP Journal on Image and Video Processing*, 2016:1–23, 2016.

[165] Bolun Cai, Xiangmin Xu, Kui Jia, Chunmei Qing, and Dacheng Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing*, 25(11):5187–5198, 2016.

[166] Boyi Li, Xiulian Peng, Zhangyang Wang, Jizheng Xu, and Dan Feng. Aod-net: All-in-one dehazing network. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4780–4788, 2017.

[167] He Zhang and Vishal M Patel. Densely connected pyramid dehazing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3194–3203, 2018.

[168] Wenqi Ren, Lin Ma, Jiawei Zhang, Jinshan Pan, Xiaochun Cao, Wei Liu, and Ming-Hsuan Yang. Gated fusion network for single image dehazing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3253–3261, 2018.

[169] Gaofeng Meng, Ying Wang, Jiangyong Duan, Shiming Xiang, and Chunhong Pan. Efficient image dehazing with boundary constraint and contextual regularization. In *2013 IEEE International Conference on Computer Vision*, pages 617–624, 2013. doi: 10.1109/ICCV.2013.82.

[170] Nicolas Hautière, Jean-Philippe Tarel, and Didier Aubert. Towards fog-free in-vehicle vision systems through contrast restoration. 06 2007. doi: 10.1109/CVPR.2007.383259.

[171] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2341–2353, 2011. doi: 10.1109/TPAMI.2010.168.

[172] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE Transactions on Image Processing*, 21(4):1742–1755, 2012. doi: 10.1109/TIP.2011.2179057.

[173] Peter C. Barnum, Srinivasa Narasimhan, and Takeo Kanade. Analysis of rain and snow in frequency space. *International Journal of Computer Vision*, 86 (2):256–274, Jan 2010. ISSN 1573-1405. doi: 10.1007/s11263-008-0200-2. URL `https://doi.org/10.1007/s11263-008-0200-2`.

[174] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. *Recurrent Squeeze-and-Excitation Context Aggregation Net for Single Image Deraining: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part VII*, pages 262–277. 09 2018. ISBN 978-3-030-01233-5. doi: 10.1007/978-3-030-01234-2_16.

[175] Wenqi Ren, Lin Ma, Jiawei Zhang, Jinshan Pan, Xiaochun Cao, Wei Liu, and Ming-Hsuan Yang. Gated fusion network for single image dehazing. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3253–3261, 2018. doi: 10.1109/CVPR.2018.00343.

[176] Xiaohong Liu, Yongrui Ma, Zhihao Shi, Linhui Dai, and Jun Chen. Towards a unified approach to single image deraining and dehazing, 03 2021.

[177] Wenhan Yang, Robby T. Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep joint rain detection and removal from a single image. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1685–1694, 2017. doi: 10.1109/CVPR.2017.183.

[178] Vijay M. Galshetwar, Ashutosh Kulkarni, and Sachin Chaudhary. Consolidated adversarial network for video de-raining and de-hazing. In *2022 18th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–8, 2022. doi: 10.1109/AVSS56176.2022.9959454.

[179] Xiao Liang, Runde Li, and Jinhui Tang. Selective attention network for image dehazing and deraining. In *Proceedings of the ACM Multimedia Asia*, MMAsia '19, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368414. doi: 10.1145/3338533.3366688. URL https://doi.org/10.1145/3338533.3366688.

[180] Hao Sun, Marcelo H. Ang, and Daniela Rus. A convolutional network for joint deraining and dehazing from a single image for autonomous driving in rain. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 962–969, 2019. doi: 10.1109/IROS40897.2019.8967644.

[181] Dong Hwan Kim, Woo Jin Ahn, Myo Taeg Lim, Tae Koo Kang, and Dong Won Kim. Frequency-based haze and rain removal network (fhrr-net) with deep convolutional encoder-decoder. *Applied Sciences*, 11(6), 2021. ISSN 2076-3417. doi: 10.3390/app11062873. URL https://www.mdpi.com/2076-3417/11/6/2873.

[182] Liang Shen, Zihan Yue, Quan Chen, Fan Feng, and Jie Ma. Deep joint rain and haze removal from a single image. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2821–2826, 2018. doi: 10.1109/ICPR.2018.8545729.

[183] Yu Li, Robby T. Tan, Xiaojie Guo, Jiangbo Lu, and Michael S. Brown. Rain streak removal using layer priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[184] Earl J McCartney. *Optics of the atmosphere : scattering by molecules and particles / Earl J. McCartney.* Wiley series in pure and applied optics. Wiley, New York ;, 1976. ISBN 0471015261.

[185] Siyuan Li, Iago Breno Araujo, Wenqi Ren, Zhangyang Wang, Eric K. Tokuda, Roberto Hirata Junior, Roberto Marcondes Cesar Junior, Jiawan Zhang, Xiaojie Guo, and Xiaochun Cao. Single image deraining: A comprehensive benchmark analysis. *CoRR*, abs/1903.08558, 2019. URL http://arxiv.org/abs/1903.08558.

[186] Ruoteng Li, Loong Fah Cheong, and Robby T. Tan. Heavy rain image restoration: Integrating physics model and conditional adversarial learning.

2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 1633–1642, 2019.

[187] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

[188] Raanan Fattal. Single image dehazing. In *ACM SIGGRAPH 2008 Papers*, SIGGRAPH '08, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781450301121. doi: 10.1145/1399504.1360671. URL https://doi.org/10.1145/1399504.1360671.

[189] Yong Liu and Xiaorong Hou. Local multi-scale feature aggregation network for real-time image dehazing. *Pattern Recognition*, 141:109599, 2023. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2023.109599. URL https://www.sciencedirect.com/science/article/pii/S003132032300300X.

[190] Aiping Yang, Yumeng Liu, Jinbin Wang, Xiaoxiao Li, Jiale Cao, Zhong Ji, and Yanwei Pang. Visual-quality-driven unsupervised image dehazing. *Neural Networks*, 167:1–9, 2023. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2023.08.010. URL https://www.sciencedirect.com/science/article/pii/S0893608023004288.

[191] Dong Hang, Pan Jinshan, Hu Zhe, Lei Xiang, Zhang Xinyi, Wang Fei, and Yang Ming-Hsuan. Multi-scale boosted dehazing network with dense feature fusion. In *CVPR*, 2020.

[192] Boyi Li, Xiulian Peng, Zhangyang Wang, Jizheng Xu, and Dan Feng. Aod-net: All-in-one dehazing network. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4780–4788, 2017. doi: 10.1109/ICCV.2017.511.

[193] Geet Sahu, Ayan Seal, Ondrej Krejcar, and Anis Yazidi. Single image dehazing using a new color channel. *Journal of Visual Communication and Image Representation*, 74:103008, 2021. ISSN 1047-3203. doi: https://doi.org/10.1016/j.jvcir.2020.103008. URL https://www.sciencedirect.com/science/article/pii/S1047320320302212.

[194] Yu Dong, Yunan Li, Qian Dong, He Zhang, and Shifeng Chen. Semi-supervised domain alignment learning for single image dehazing. *IEEE Transactions on Cybernetics*, 2022.

[195] Huafeng Li, Jirui Gao, Yafei Zhang, Minghong Xie, and Zhengtao Yu. Haze transfer and feature aggregation network for real-world single image dehazing. *Knowledge-Based Systems*, 251:109309, 2022. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2022.109309. URL `https://www.sciencedirect.com/science/article/pii/S0950705122006566`.

[196] Yongzhen Wang, Xuefeng Yan, Donghai Guan, Mingqiang Wei, Yiping Chen, Xiao-Ping Zhang, and Jonathan Li. Cycle-snspgan: Towards real-world image dehazing via cycle spectral normalized soft likelihood estimation patch gan. *IEEE Transactions on Intelligent Transportation Systems*, 23(11):20368–20382, 2022. doi: 10.1109/TITS.2022.3170328.

[197] Akshay Dudhane, Prashant W. Patil, and Subrahmanyam Murala. An end-to-end network for image de-hazing and beyond. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(1):159–170, 2022. doi: 10.1109/TETCI.2020.3035407.

[198] Boyun Li, Yuanbiao Gou, Shuhang Gu, Jerry Zitao Liu, Joey Tianyi Zhou, and Xi Peng. You Only Look Yourself: Unsupervised and Untrained Single Image Dehazing Neural Network. *International Journal of Computer Vision*, pages 1–14, March 2021.

[199] Zhengguo Li, Haiyan Shu, and Chaobing Zheng. Multi-scale single image dehazing using laplacian and gaussian pyramids. *IEEE Transactions on Image Processing*, 30:9270–9279, 2021. doi: 10.1109/TIP.2021.3123551.

[200] Jiafeng Li, Yaopeng Li, Li Zhuo, Lingyan Kuang, and Tianjian Yu. Usid-net: Unsupervised single image dehazing network via disentangled representations. *IEEE transactions on multimedia*, 2022.

[201] Yang Yang, Chaoyue Wang, Risheng Liu, Lin Zhang, Xiaojie Guo, and Dacheng Tao. Self-augmented unpaired image dehazing via density and depth decomposition. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2027–2036, 2022. doi: 10.1109/CVPR52688.2022.00208.

[202] Yanyun Qu, Yizi Chen, Jingying Huang, and Yuan Xie. Enhanced pix2pix dehazing network. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8152–8160, 2019. URL `https://api.semanticscholar.org/CorpusID:195489762`.

[203] Yu Zhou, Zhihua Chen, Ping Li, Haitao Song, C. L. Philip Chen, and Bin Sheng. Fsad-net: Feedback spatial attention dehazing network. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–15, 2022. doi: 10.1109/TNNLS.2022.3146004.

[204] Yu Dong, Yihao Liu, He Zhang, Shifeng Chen, and Yu Qiao. Fd-gan: Generative adversarial networks with fusion-discriminator for single image dehazing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):10729–10736, Apr. 2020. doi: 10.1609/aaai.v34i07.6701. URL `https://ojs.aaai.org/index.php/AAAI/article/view/6701`.

[205] Po-Wen Hsieh and Pei-Chiang Shao. Variational contrast-saturation enhancement model for effective single image dehazing. *Signal Processing*, 192:108396, 2022. ISSN 0165-1684. doi: https://doi.org/10.1016/j.sigpro.2021.108396. URL `https://www.sciencedirect.com/science/article/pii/S0165168421004333`.

[206] Xiaoning Liu, Hui Li, and Ce Zhu. Joint contrast enhancement and exposure fusion for real-world image dehazing. *IEEE Transactions on Multimedia*, 24:3934–3946, 2022. doi: 10.1109/TMM.2021.3110483.

[207] Cunyi Lin, Xianwei Rong, and Xiaoyan Yu. Msaff-net: Multiscale attention feature fusion networks for single image dehazing and beyond. *IEEE Transactions on Multimedia*, 25:3089–3100, 2023. doi: 10.1109/TMM.2022.3155937.

[208] Yitong Zheng, Jia Su, Shun Zhang, Mingliang Tao, and Ling Wang. Dehaze-aggan: Unpaired remote sensing image dehazing using enhanced attention-guide generative adversarial networks. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2022. doi: 10.1109/TGRS.2022.3204890.

[209] Bilel Benjdira, Anas M. Ali, and Anis Koubaa. Streamlined global and local features combinator (sglc) for high resolution image dehazing. In *2023*

IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 1855–1864, 2023. doi: 10.1109/CVPRW59228.2023.00184.

[210] Yu Guo, Yuan Gao, Wen Liu, Yuxu Lu, Jingxiang Qu, Shengfeng He, and Wenqi Ren. Scanet: Self-paced semi-curricular attention network for non-homogeneous image dehazing. 04 2023. doi: 10.1109/CVPRW59228.2023.00186.

[211] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley. Clearing the skies: A deep network architecture for Single-Image rain removal. *IEEE Trans Image Process*, 26(6):2944–2956, April 2017.

[212] Yanyan Wei, Zhao Zhang, Yang Wang, Haijun Zhang, Mingbo Zhao, Mingliang Xu, and Meng Wang. Semi-deraingan: A new semi-supervised single image deraining. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2021. doi: 10.1109/ICME51207.2021.9428285.

[213] Rajeev Yasarla and Vishal M. Patel. Uncertainty guided multi-scale residual learning-using a cycle spinning cnn for single image de-raining. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[214] Junbo Wang, Wei Wang, Liang Wang, and Tieniu Tan. Prenet: Parallel recurrent neural networks for image classification. In Jinfeng Yang, Qinghua Hu, Ming-Ming Cheng, Liang Wang, Qingshan Liu, Xiang Bai, and Deyu Meng, editors, *Computer Vision*, pages 461–473, Singapore, 2017. Springer Singapore. ISBN 978-981-10-7302-1.

[215] Yu Li, Robby T. Tan, Xiaojie Guo, Jiangbo Lu, and Michael S. Brown. Rain streak removal using layer priors. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2736–2744, 2016. doi: 10.1109/CVPR.2016.299.

[216] Liang-Jian Deng, Ting-Zhu Huang, Xi-Le Zhao, and Tai-Xiang Jiang. A directional global sparse model for single image rain removal. *Applied Mathematical Modelling*, 59:662–679, 2018. ISSN 0307-904X. doi: https://doi.org/10.1016/j.apm.2018.03.001. URL `https://www.sciencedirect.com/science/article/pii/S0307904X18301069`.

[217] Dongdong Chen, Mingming He, Qingnan Fan, Jing Liao, Liheng Zhang, Dongdong Hou, Lu Yuan, and Gang Hua. Gated context aggregation network for image dehazing and deraining. *WACV 2019*, 2018.

[218] Wenhan Yang, Robby T. Tan, Jiashi Feng, Zongming Guo, Shuicheng Yan, and Jiaying Liu. Joint rain detection and removal from a single image with contextualized deep networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(6):1377–1393, 2020. doi: 10.1109/TPAMI.2019.2895793.

[219] Yu Luo, Yong Xu, and Hui Ji. Removing rain from a single image via discriminative sparse coding. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3397–3405, 2015. URL `https://api.semanticscholar.org/CorpusID:14460720`.

[220] P. Musafira and K. S. Shanthini. Single image rain removal using convolutional neural network. In S. N. Merchant, Krishna Warhade, and Debashis Adhikari, editors, *Advances in Signal and Data Processing*, pages 135–145, Singapore, 2021. Springer Singapore. ISBN 978-981-15-8391-9.

[221] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *European Conference on Computer Vision*, 2018. URL `https://api.semanticscholar.org/CorpusID:49864080`.

[222] Youzhao Yang and Hong Lu. Single image deraining via recurrent hierarchy enhancement network. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, page 1814–1822, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450368896. doi: 10.1145/3343031.3351149. URL `https://doi.org/10.1145/3343031.3351149`.

[223] Wei Wei, Deyu Meng, Qian Zhao, Zongben Xu, and Ying Wu. Semi-supervised transfer learning for image rain removal. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[224] Cong Wang, Honghe Zhu, Wanshu Fan, Xiao-Ming Wu, and Junyang Chen. Single image rain removal using recurrent scale-guide networks. *Neurocomputing*, 467:242–255, 2022. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2021.10.029. URL `https://www.sciencedirect.com/science/article/pii/S0925231221015071`.

[225] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17662–17672, 2022. doi: 10.1109/CVPR52688.2022.01716.

[226] Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. A model-driven deep neural network for single image rain removal. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[227] Jing Nie, Jin Xie, Jiale Cao, and Yanwei Pang. Context and detail interaction network for stereo rain streak and raindrop removal. *Neural Networks*, 166:215–224, 2023. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2023.07.013. URL `https://www.sciencedirect.com/science/article/pii/S0893608023003702`.

[228] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, 2022.

[229] Kaiyu Zhang, Jinglong Chen, Shuilong He, Fudong Li, Yong Feng, and Zitong Zhou. Triplet metric driven multi-head gnn augmented with decoupling adversarial learning for intelligent fault diagnosis of machines under varying working condition. *Journal of Manufacturing Systems*, 62:1–16, 2022. ISSN 0278-6125. doi: https://doi.org/10.1016/j.jmsy.2021.10.014. URL `https://www.sciencedirect.com/science/article/pii/S0278612521002211`.

[230] Rui Qian, Robby T Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2482–2491, 2018.

[231] Rajeev Yasarla and Vishal M. Patel. Uncertainty guided multi-scale residual learning-using a cycle spinning cnn for single image de-raining. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8397–8406, 2019. doi: 10.1109/CVPR.2019.00860.

[232] Kui Jiang, Zhongyuan Wang, Peng Yi, Chen Chen, Baojin Huang, Yimin Luo, Jiayi Ma, and Junjun Jiang. Multi-scale progressive fusion network for single image deraining. In *2020 IEEE/CVF Conference on Computer*

Vision and Pattern Recognition (CVPR), pages 8343–8352, 2020. doi: 10.1109/CVPR42600.2020.00837.

[233] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14816–14826, 2021. doi: 10.1109/CVPR46437.2021.01458.

[234] Chong Mou, Qian Wang, and Jian Zhang. Deep generalized unfolding networks for image restoration. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17378–17389, 2022. doi: 10.1109/CVPR52688.2022.01688.

[235] Yitong Yang, Yongjun Zhang, Zhongwei Cui, Haoliang Zhao, and Ting Ouyang. Single image deraining using scale constraint iterative update network. *Expert Systems with Applications*, 236:121339, 2024. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2023.121339. URL `https://www.sciencedirect.com/science/article/pii/S0957417423018419`.

[236] Kanchan Yadav, Pratibha Soram, Sheela Bijlwan, Bhawna Goyal, Ayush Dogra, and Dawa Chyophel Lepcha. Dynamic economic load dispatch problem in power system using iterative genetic algorithm. In *2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 1629–1632, 2023. doi: 10.1109/ICIRCA57980.2023.10220653.

[237] Dawei Yang, Xin He, and Ruiheng Zhang. Alternating attention transformer for single image deraining. *Digital Signal Processing*, 141:104144, 2023. ISSN 1051-2004. doi: https://doi.org/10.1016/j.dsp.2023.104144. URL `https://www.sciencedirect.com/science/article/pii/S1051200423002397`.

[238] Jie Xiao, Xueyang Fu, Aiping Liu, Feng Wu, and Zheng-Jun Zha. Image deraining transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12978–12995, 2023. doi: 10.1109/TPAMI.2022.3183612.

[239] Pengpeng Liang, Erik Blasch, and Haibin Ling. Encoding color information for visual tracking: Algorithms and benchmark. *IEEE Transactions on Image Processing*, 24(12):5630–5644, 2015. doi: 10.1109/TIP.2015.2482905.

[240] Matej Kristan, Roman P Pflugfelder, Ales Leonardis, Jiri Matas, Luka Cehovin, Georg Nebehay, Tomas Vojir, Gustavo Fernandez, Alan Lukezi, Aleksandar Dimitriev, Alfredo Petrosino, Amir Saffari, Bo Li, Bohyung Han, CherKeng Heng, Christophe Garcia, Dominik Pangersic, Gustav Häger, Fahad Shahbaz Khan, Franci Oven, Horst Possegger, Horst Bischof, Hyeonseob Nam, Jianke Zhu, JiJia Li, Jin Young Choi, Jin-Woo Choi, Joao F Henriques, Joost van de Weijer, Jorge Batista, Karel Lebeda, Kristoffer Ofjall, Kwang Moo Yi, Lei Qin, Longyin Wen, Mario Edoardo Maresca, Martin Danelljan, Michael Felsberg, Ming-Ming Cheng, Philip Torr, Qingming Huang, Richard Bowden, Sam Hare, Samantha YueYing Lim, Seunghoon Hong, Shengcai Liao, Simon Hadfield, Stan Z Li, Stefan Duffner, Stuart Golodetz, Thomas Mauthner, Vibhav Vineet, Weiyao Lin, Yang Li, Yuankai Qi, Zhen Lei, and ZhiHeng Niu. The visual object tracking vot2014 challenge results, 2015. ISSN 0302-9743.

[241] Matej Kristan, Aleš Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Cehovin Zajc, Tomás Vojír, Gustav Häger, Alan Lukežic, Abdelrahman Eldesokey, Gustavo Fernández, Álvaro García-Martín, A. Muhic, Alfredo Petrosino, Alireza Memarmoghadam, Andrea Vedaldi, Antoine Manzanera, Antoine Tran, Aydin Alatan, Bogdan Mocanu, Boyu Chen, Chang Huang, Changsheng Xu, Chong Sun, Dalong Du, David Zhang, Dawei Du, Deepak Mishra, Erhan Gundogdu, Erik Velasco-Salido, Fahad Shahbaz Khan, Francesco Battistone, Gorthi R. K. Sai Subrahmanyam, Goutam Bhat, Guan Huang, Guilherme Bastos, Guna Seetharaman, Hongliang Zhang, Houqiang Li, Huchuan Lu, Isabela Drummond, Jack Valmadre, Jae-chan Jeong, Jae-il Cho, Jae-Yeong Lee, Jana Noskova, Jianke Zhu, Jin Gao, Jingyu Liu, Ji-Wan Kim, João F. Henriques, Jose M. Martínez, Junfei Zhuang, Junliang Xing, Junyu Gao, Kai Chen, Kannappan Palaniappan, Karel Lebeda, Ke Gao, Kris M. Kitani, Lei Zhang, Lijun Wang, Lingxiao Yang, Longyin Wen, Luca Bertinetto, Mahdieh Poostchi, Martin Danelljan, Matthias Mueller, Mengdan Zhang, Ming-Hsuan Yang, Nianhao Xie, Ning Wang, Ondrej Miksik, P. Moallem, Pallavi M. Venugopal, Pedro Senna, Philip H. S. Torr, Qiang Wang, Qifeng Yu, Qingming Huang, Rafael Martín-Nieto, Richard Bowden, Risheng Liu, Ruxandra Tapu, Simon Hadfield, Siwei Lyu, Stuart Golodetz, Sunglok Choi, Tianzhu Zhang, Titus Zaharia, Vincenzo Santopietro, Wei Zou, Weiming Hu, Wenbing Tao, Wenbo Li, Wengang Zhou, Xianguo Yu, Xiao Bian, Yang Li, Yifan Xing,

Yingruo Fan, Zheng Zhu, Zhipeng Zhang, and Zhiqun He. The visual object tracking vot2017 challenge results. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 1949–1972, 2017. doi: 10.1109/ICCVW.2017.230.

[242] Annan Li, Min Lin, Yi Wu, Ming-Hsuan Yang, and Shuicheng Yan. Nus-pro: A new visual tracking challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):335–349, 2016. doi: 10.1109/TPAMI.2015.2417577.

[243] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. volume 9905, pages 445–461, 10 2016. ISBN 978-3-319-46447-3. doi: 10.1007/978-3-319-46448-0_27.

[244] Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, and Simon Lucey. Need for speed: A benchmark for higher frame rate object tracking. *CoRR*, abs/1703.05884, 2017. URL `http://arxiv.org/abs/1703.05884`.

[245] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *CoRR*, abs/1810.11981, 2018. URL `http://arxiv.org/abs/1810.11981`.

[246] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. *CoRR*, abs/1809.07845, 2018. URL `http://arxiv.org/abs/1809.07845`.

[247] Gary Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Tech., 2008.

[248] S. Sengupta, J. C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs. Frontal to profile face verification in the wild. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9, 2016. doi: 10.1109/WACV.2016.7477558.

[249] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. Agedb: The first manually collected, in-the-wild age database. In *Proceedings of CVPRW*, pages 1997–2005, 2017. doi: 10.1109/CVPRW.2017.250.

[250] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything. *arXiv preprint arXiv:2306.12156*, 2023.

[251] Alan Lukezic, Tomas Vojir, Luka ˇCehovin Zajc, Jiri Matas, and Matej Kristan. Discriminative correlation filter with channel and spatial reliability. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6309–6318, 2017.